



MapXtreme

v7.3

开发人员指南

本文档中的信息如有更改恕不另行通知，且不表示对供应商或其代表的任何部分做出承诺。未经 Pitney Bowes Inc. (地址为：One Global View, Troy, New York 12180-8399) 书面许可，不得以任何电子或机械形式或手段（包括影印），复制或传播本文档的任何部分。

© 2015 Pitney Bowes Inc. 保留所有权利。Pitney Bowes Inc. Pitney Bowes、公司徽标、MapInfo、Group 1 Software 和 MapXtreme 是 Pitney Bowes Inc. 或其子公司的注册商标。所有其它的商标均为其各自所有者的财产。

所有 Pitney Bowes Inc. 分部的联系信息都位于：www.pitneybowes.com。

本产品中包含 Spatialite v 3.1.0, 其许可为 GNU Lesser General Public License, Version 2.1, February 1999。可从以下位置下载该许可：
<http://www.gnu.org/licenses/lgpl-2.1.html>。本软件的源代码：<http://www.gaia-gis.it/gaia-sins/win-bin-x86-test/spatialite-3.1.0b-test-win-x86.zip> 和
<http://www.gaia-gis.it/gaia-sins/win-bin-amd64-test/spatialite-3.1.0b-test-win-amd64.zip>。

本产品中包含 Feature Data Objects v 3.6.0, 其许可为 GNU Lesser General Public License, Version 2.1, February 1999。可从以下位置下载该许可：
<http://fdo.osgeo.org/lGPL.html>。本软件的源代码：<http://fdo.osgeo.org/content/fdo-360-downloads>。

本产品中包含 HelpLibraryManagerLauncher.exe v 1.0.0.1, 其许可为 Microsoft Public License。可从以下位置下载该许可：
<http://shfb.codeplex.com/license>。本软件的源代码：<http://shfb.codeplex.com>。

2015 年 06 月 01 日

目录

第 1 章: MapXtreme 简介	22
MapXtreme 概述	23
主要功能	23
迁移到 MapXtreme	25
学习使用 MapXtreme	27
支持资源	27
第 2 章: 入门指南	29
安装要求	30
最低系统要求	31
安装类型	31
开发 (SDK) 安装	31
部署 (运行时) 安装	31
并行安装和使用	32
MapXtreme 许可	32
许可证类型	32
获取许可证	33
许可证文件的位置	34
故障诊断	36
已知的许可问题	37
安装之前	38
管理员特权	38
首先安装 .NET Framework 和 Visual Studio	38
支持 IIS 7/8	38
MapXtreme 的默认安装目录	39
其它安装特性	40
将 MapXtreme 安装到系统中	40
升级到 MapXtreme	43
将 Web 站点迁移到 64 位 Web 应用程序中	43
更新现有的 Web 站点	43
更新现有的桌面应用程序	44
在 Visual Studio 中创建应用程序	46

地图应用程序	46
ASP.NET Web 应用程序	48
MapXtreme控件	50
构建 ASP.NET Web 应用程序但不使用模板	50
部署应用程序	52
用运行时安装程序部署	52
用自己的安装程序进行部署	53
部署 web 应用程序	56
部署访问数据的应用程序	56
在代理服务器背后部署 MapXtreme Web 应用程序	57
已部署 Web 应用程序需要访问临时目录的权限	57
应用程序数据文件	57
部署安装故障诊断	58
第 3 章: 地图绘制概念	59
地图绘制和MapXtreme	60
地图	60
表	60
Layers	61
Features	61
标注和图例	61
Themes	62
工具	63
工作空间	63
坐标系和投影	63
使用 MapXtreme 进行地理编码	64
使用 MapXtreme 进行路径规划	64
第 4 章: 理解 MapXtreme 体系结构	65
MapXtreme体系结构	66
对象模型概述	67
MapInfo.Data 命名空间	67
MapInfo.Data.Find 命名空间	68
MapInfo.Engine 命名空间	68
MapInfo.Geometry 命名空间	68
MapInfo.Mapping 命名空间	68
MapInfo.Mapping.Legends 命名空间	68
MapInfo.Mapping.Thematics 命名空间	68
MapInfo.Persistence 命名空间	68
MapInfo.Raster 命名空间	69
MapInfo.Styles 命名空间	69
MapInfo.WebControls 命名空间	69

MapInfo.Windows 命名空间	69
MapInfo.Tools 命名空间	69
MapInfo.Geocoding 命名空间	69
MapInfo.Routing 命名空间	70
应用程序体系结构	70
Web 应用程序体系结构	71
桌面应用程序体系结构	73
第 5 章: Web 应用程序、控件和工具	74
Web 应用程序请求/响应周期	75
MapXtreme Web 应用程序组件	75
MapXtreme Session	75
背景地图	75
MapControl	76
地图工具	78
状态管理和对象池功能	78
MapXtreme Web 控件和工具	78
Web 控件和工具的描述	80
Web 控件的体系结构	81
地图工具体系结构	82
地图工具的功能	83
使用 MapXtreme Web 控件	84
管理 Web 控件	85
事件处理	85
错误管理	85
状态管理 (State Management)	86
创建定制工具	86
使用和分发定制 Web 控件	88
创建 Web 程序集	88
将 InfoTool 添加到 Web 应用程序	89
ASP.NET AJAX 和 MapXtreme Web 应用程序	90
将 ASP.NET AJAX 控件添加至 MapXtreme Web 应用程序	90
MapXtreme 瓦块句柄	92
使用 MapXtreme 瓦块句柄	92
缓存	94
HTML/XHTML 验证问题	95
将 Post-back Web 控件迁移到 JavaScript Web 控件	96
加载数据	96
替换控件	96
状态和事件管理	97
Web 控件专题	97
在框架中使用 Web 控件	97

在表单元格中使用 MapControl	98
Web 控件本地化	98
第 6 章: 了解状态管理	99
概述	100
术语	100
什么是 状态管理	101
有哪些状态管理选项可用?	102
撰写应用程序前应提出的问题	102
InProc 开发模型	104
InProc 开发模型的利与弊	104
InProc 管理: 简介	105
配置应用程序使用 InProc 开发模型	105
使用支持 InProc 开发模型的 MapXtreme 模板	106
对象池的状态管理	106
什么是对象池?	106
对象池的利与弊	107
保存对象池应用程序的状态	108
手动状态管理: 简介	109
配置对象池应用程序使用手动状态管理	110
手动状态管理详探	111
主题图示例概述	111
应用程序设置	112
实现 StateManager	113
按照正常顺序对 MapXtreme 对象进行序列化	114
自动序列化 MapXtreme 对象	115
处理初始请求	115
处理后继请求	116
关于 MapXtreme Session 的深入讨论	116
配置 Microsoft COM+ 对象池	117
第 7 章: 桌面应用程序、控件、对话框和工具	119
规划桌面应用程序	120
桌面应用程序的最佳实践	120
MapXtreme 和 COM	120
示例应用程序和项目模板	121
MapInfo.Windows.Controls 命名空间	121
要在桌面应用程序中使用的键控件	122
MapControl	123
MapToolStripButtons	124
MapToolBar	125
图层控制	127

MapInfo.Windows.Dialogs 命名空间	128
CreateThemeWizard	129
定制控件和对话框	133
MapInfo.Tools 命名空间概述	134
MapXtreme 桌面工具 API.	135
View 工具	136
Select 工具	136
Add 工具	137
Custom 工具	137
Shape 工具	138
使用 InfoTip	139
定制工具	139
工具事件	140
使用 Select 工具编辑 FeatureGeometry	141
重新定形图元	141
增加节点	143
通过编程方式重新定形和增加节点	144
第 8 章: 使用数据	146
MapInfo.Data 命名空间概述	147
目录和表	148
表	148
Catalog	152
支持的表类型	153
处理目录和表	155
定位打开的表	156
关闭表	156
打包表	157
侦听表和目录事件	157
表示数据 (TableInfo).	158
检查 TAB 文件元数据	159
新建表	159
将表达式列增加到表	162
数据源	162
选取正确的数据源	163
访问数据的方法	163
Data Readers、MemTables 和 Result Sets	164
使用 ADO.NET 数据提供方	165
数据绑定 (Data Binding).	168
使表成为可制图	171
MapInfo ADO.NET 数据提供方	174
MConnection	174

MICommand	175
MIDataReader	176
MapInfo SQL	177
Features 和 Feature 集合	178
Feature	178
Feature 集合	179
搜索图元	179
Catalog 搜索方法	180
SearchInfo 和 SearchInfoFactory	181
分析数据	184
提升数据访问性能	187
第 9 章: 运用核心 MapXtreme 类	188
Session 接口	189
会话管理	189
使用 Session.Dispose 方法	190
ISessionEventHandlers	191
串行化和永久化	191
序列化 (Serialization)	192
永久性	193
打开和保存包含命名资源的工作空间	193
打开 MWS: ResolveResource()	194
保存 MWS: GetResourceName()	194
使用 MapXtreme 注册您的实现。	194
设置参数选择	194
Selection 类	194
使用选择集的属性	195
突出显示和导出选择集	195
SelectionChangedEvent	195
Selection 和 Selections 类上的 ISerializable 接口	196
Selection 代码示例	196
选择另一个图元中的图元	196
检查选择集的表	196
从表中返回所有列	197
更改选择集后的地图视图	197
事件参数	198
异常	198
第 10 章: 创建表达式	199
表达式概述	200
创建表达式	200
Where 子句 - 布尔表达式	201

表达式中的函数	201
DateTlme 和 Time 表达式	202
表达式示例	202
第 11 章: 从 DBMS 访问数据	206
访问远程空间数据	207
通过 .TAB 文件访问远程表	207
不使用 .TAB 文件访问远程表	207
使用 X/Y 列映射 DBMS 数据	208
访问 Oracle 数据库中的数据	208
几何体转换	208
对 Z 和 M 值的 Oracle 支持	209
SDO_Geometry Arc 和 Circle 转换	210
可视化不可转换的 Oracle 对象	210
质心支持	210
Oracle Spatial 引用支持 (SRID)	210
OCI 连接对话框	211
访问 MS SQL Server 中的数据	211
支持 SQL Server 2008	211
DBMS 连接字符串格式	214
ODBC 连接字符串格式	214
Web 应用程序中的 ODBC 图层和对象池	215
Oracle Spatial 连接字符串格式	215
样本连接字符串	215
在服务器表查询中定义可制图的表	216
几何体列	216
键列	217
访问属性数据	217
性能问题	218
运用缓存	218
什么是缓存?	218
缓存如何工作	219
TableInfoServer 对象和 CacheSettings 属性	219
MapInfo_MapCatalog	220
将空间数据加载到 DBMS	220
手动创建 MapInfo MapCatalog	221
向 MapInfo_MapCatalog 添加行	222
每记录样式	226
Symbol、Pen、Brush 子句语法	226
文本对象限制	227
故障诊断	227

第 12 章: 将地图绘制功能增加到 应用程序	229
MapInfo.Mapping 命名空间简介	230
基本地图绘制类	230
MapExport	230
Map	231
MapFactory	231
MapLoader	232
MapViewList、MapView	232
MapControl	232
Layers	232
FeatureLayer	233
Layers	233
MapLayer	233
UserDrawLayer	234
ObjectThemeLayer	234
GroupLayer	234
LabelLayer	234
GraticuleLayer	234
图层筛选器	234
IVisibilityConstraint	234
代码示例: 动画图层	234
Labels	236
LabelLayer	237
LabelSource	237
LabelModifier	237
ILabelSourceFilter	238
LabelProperties	238
生成标注	238
标注优先权	238
标注图层的可选择性	239
代码示例: 创建 LabelLayer	239
曲线标注	240
Adornments	240
图例	240
ScaleBar 修饰	241
标题修饰	242
图元样式修饰符	242
FeatureStyleModifier	243
FeatureStyleModifiers	243
FeatureOverrideStyleModifier	243

打印地图	244
第 13 章: 查找位置	245
查找功能概述	246
查找过程	246
匹配地址号码	247
使用优化边界表匹配	248
查找结果	248
Data.Find 命名空间概述	248
查找	249
.FindAddressRange	250
FindCloseMatch	251
FindResult	252
微调查找过程	254
编辑 Mapinfow.abb 文件	254
第 14 章: 使用主题和图例	260
主题概述	261
Mapping.Thematics 命名空间	261
修饰符主题	261
对象主题	261
GraduatedSymbolTheme	262
何时使用分级符号主题	263
PieTheme	263
何时使用饼图主题	263
打印包括饼形/条形主题的地图	264
BarTheme	264
何时使用条形图主题	264
控制饼状主题图和条形主题图的显示尺寸	265
RangedTheme	265
何时使用范围主题	266
范围值的类型	266
RangedLabelTheme	267
何时使用 RangedLabelTheme 类	268
范围主题和序列化	268
IndividualValueTheme	268
何时使用 IndividualValueTheme 类	269
使用自定义位图符号创建 IndividualValueTheme	269
IndividualValueLabelTheme	270
何时使用 IndividualValueLabelTheme 类	270
IndividualValue 主题和序列化	271
DotDensityTheme	271

何时使用 DotDensityTheme 类	271
双变量主题地图	272
图例概述	273
主题图例	273
制图图例	273
将图例格式化	274
第 15 章: 为地图设置样式	275
MapInfo.Styles 命名空间概述	276
StyleFactory	277
样式说明	277
AreaStyle	277
BitmapPointStyle	277
CompositeStyle	278
SimpleInterior	278
Font	278
FontPointStyle	278
GridStyle	278
RasterStyle	278
Hillshade	279
反射	279
SimpleLineStyle	279
BasePointStyle	279
BaseLineStyle	279
BaseInterior	279
StockStyles	279
TextStyle	280
SimpleVectorPointStyle	280
预定义的样式和 StyleRepository 类	280
StyleRepository 类	280
使用样式	281
样式和图层控制	281
创建定制位图样式	281
覆盖样式	282
FeatureOverrideStyleModifiers	282
第 16 章: 空间对象和坐标系	283
MapInfo.Geometry 命名空间简介	284
几何体	284
Geometry 对象	285
FeatureGeometry 对象	286
Geometry 对象	289

将 FeatureGeometry 包括在地图中	291
检查折线中的点	292
坐标系	293
创建 CoordSys 对象	293
更改 Geometry 对象的坐标系	294
确定 MapControl 中的地图坐标系	295
将坐标系添加到 MapXtreme	295
第 17 章: 运用光栅和网格	299
MapInfo.Raster 命名空间概述	300
光栅图像	300
光栅类	301
光栅图像和坐标系	302
栅格重投影	302
光栅图像的限制	302
代码示例: 将光栅图像添加到地图	302
光栅句柄	303
光栅句柄属性	304
配置自定义光栅处理程序	305
网格图像	306
网格类	306
代码示例: 将网格图像添加到地图	307
代码示例: 从网格地图检索数据	307
创建网格	308
网格内插器	309
反距离加权 (IDW) 内插器	309
不规则三角网络 (TIN) 内插器	309
IInterpolator 接口	310
网格样式	310
网格图像和变化	310
变化方法	310
计算网格图层的变化值和颜色	311
辅助底纹	312
“网格样式”对话框	312
GridInfoForm 示例应用程序	314
第 18 章: 使用图片服务器地图	315
图片服务器图像	316
瓦块缓存	316
图片服务器图层的地图行为	316
使用图片服务器图像	317
QuadKey	317

LevelRowColumn	317
向图片服务器验证	317
图片服务器设置	318
Bing 地图的许可密钥	318
通过 Web 或桌面配置文件	318
通过 MapInfo.Engine.TileServerSettings 类	318
TileServerSettings 类的示例代码	318
使用 TableInfoTileServer 类	319
图片服务器示例应用程序	320
第 19 章: 地理编码	321
MapInfo.Geocoding 命名空间概述	322
主要的地理编码类	322
GeocodeRequest	323
GeocodeResponse	323
GeocodeClientFactory	323
GeocodingConstraints	323
AddressCandidates	324
BaseGeocodeMatchCode 和 GeocodeMatchCode	324
CandidateAddress	324
了解地理编码模型	324
有关地理编码的一些权衡考虑	324
有关地址的说明	325
什么是自定义用户字典?	325
什么是 World Geocoding?	326
对位置进行地理编码	326
街道地址地理编码	326
街道十字路口地理编码	328
邮政编码的地理编码	328
Gazetteer 类型地理编码	328
批量地理编码	329
使用约束条件以实现精确地理编码	329
什么是匹配约束条件?	329
发送匹配约束条件的影响	331
了解近似匹配项的精确度	332
单个近似匹配 (S 类别)	333
来自多个候选选项的最佳匹配 (M 类别)	333
邮政编码中心位置匹配 (Z 类别)	333
地理中心匹配 (G 类别)	334
无匹配代码	334

第 20 章: 路径规划	335
MapInfo.Routing 命名空间概述	336
主要的路径规划类	336
路径计算	337
点到点路径规划	337
多点路径规划	338
矩阵路径规划	339
高级路径选项	340
路径规划参数选择	340
驾驶导航	341
路径几何体	342
避免点、图元和分段	342
基于时间的路径规划	342
Iso 路径规划（驾驶时间和驾驶距离）	343
创建 IsoChrone（驾驶时间）	344
创建 IsoDistance（驾驶距离）	346
使用路径规划数据更新请求	347
返回分段信息	348
瞬态更新	348
第 21 章: 线性参考	353
什么是线性参考	354
使用线性参考的 M 值	354
度量值确定法	355
线性参考运算	355
动态分段运算 (PerpendicularOffset)	356
曲线顺序	356
线性参考示例应用程序	357
第 22 章: Web 图元服务	358
Web 图元服务	359
理解 WFS 服务器的操作	359
配置 WFS 服务器	363
步骤 1: 创建 Web.config 文件	364
步骤 2: 为待托管的图元创建有效的 WFS 配置文件	365
步骤 3: 配置和测试 WFS 服务器	367
通过编程方式使用 MapXtreme WFS 客户端	369
在 WFS 查询中使用筛选器	370
根据 WFS 响应创建地图图层	372

第 23 章: Web 地图服务	376
MapXtreme Web 地图服务简介	377
理解 WMS 操作	377
使用 MapXtreme 作为 WMS 客户端	378
代码示例: 请求 WMS 图层	379
WMS 和坐标系	380
地图和图像边界	380
MapXtreme WMS 和身份验证	381
基本身份验证	381
设置 MapXtreme WMS 服务器	382
步骤 1: 创建 Web.config 文件	382
步骤 2: 为待托管的数据创建有效的 WMS 配置文件	383
步骤 3a: 使用 IIS7 配置和测试 WMS 服务器	386
步骤 3b: 使用 IIS6 配置和测试 WMS 服务器	387
配置 WMS 服务器的图层信息	390
第 24 章: Workspace 管理器	393
Workspace 管理器的特性	394
工作区格式和内容	395
Workspace 管理器菜单命令	395
“文件”菜单命令	395
查看菜单命令	397
“地图”菜单命令	398
“工具”菜单命令	400
扩展菜单命令	403
图层控制	404
图层控件工具	404
图层树	405
图层控件选项卡	406
地图设置	406
图层设置	410
主题图层设置	411
标注图层设置	411
“组图层”设置	414
“样式覆盖”设置	414
网格图层设置	414
使用 Workspace 管理器的功能	415
使用 GDI+ 半透明度和抗锯齿增强渲染能力	416
创建半透明效果	418
曲线标注	420
网格图层	423

第 25 章: 使用 GeoDictionary 管理器	425
使用 GeoDictionary 管理器	426
GeoDictionary 管理器中的变化	426
GeoDictionary 管理器的用户界面	426
运行 GeoDictionary 管理器	426
GeoDictionary 文件	429
示例 .dct 文件	429
附录 A: 如何创建和部署 MapXtreme 应用程序	431
定制 MapXtreme 示例	432
构建桌面应用程序	432
运行示例应用程序	432
修改应用程序	433
在发行模式下生成	439
打包桌面应用程序	439
部署桌面应用程序	441
生成 Web 应用程序	441
运行示例 Web 应用程序	442
修改应用程序	443
关于状态管理的考虑	445
配置发行模式	445
打包 Web 应用程序	446
部署 Web 应用程序	449
附录 B: 自定义 MapXtreme	451
可定制的类	452
MapInfo.Data.Provider 命名空间	452
ADO.NET	452
Engine.CustomProperties	452
Search	453
FeatureStyleModifier 或 FeatureOverrideStyleModifier	453
UserDrawLayer	454
Windows.Controls	455
工具	456
样式	457
GmlFeatureCollection	458
WorkSpacePersistence 和 WorkSpaceLoader	458
Workspace 管理器扩展	459
创建 Workspace 扩展	459
加载扩展	461
卸载扩展	462
示例扩展	462

应用程序数据文件的位置.....	463
查找缩写文件.....	465
附录 C: 理解 MapInfo 工作空间	466
什么是 MapInfo 工作空间?	467
工作空间的结构.....	467
Header 部分	468
Connection 部分	468
DataSource Definition 部分	469
Map Definition 部分	469
通过 .GST 以编程方式创建 .MWS 工作空间	471
附录 D: 可扩展数据提供方	473
简介.....	474
可扩展数据提供方概述	474
入门.....	476
所需组件	477
可选构建块: 基类、帮助程序和实用程序	479
示例: COTW (Center of the World) 数据提供方.	480
可选接口	481
IDataSource	482
IDataSourceDefinition	482
ITableModifyProcessor	482
构建和测试数据提供方	482
Spatialite 示例数据提供方.....	484
高级主题/重点考虑事项	485
创建几何体	485
坐标系	486
样式	486
异常处理	487
永久性提供方	487
序列化	489
身份验证	492
线程安全	495
附录 E: 从 MapXtreme 应用程序进行打印	496
概述	497
了解 MapXtreme 中的打印选项	497
打印大小	498
特殊透明光栅处理	498
特殊透明矢量处理	498
尽可能地以真彩色显示光栅	499
GDI+ 半透明度和抗锯齿	499

抖动方法	500
特殊多边形孔处理	500
缩放图案	500
直接打印到设备	500
使用增强元文件 (EMF) 打印	501
在应用程序中实现打印功能	501
常规打印技巧与方法	502
打印地图中的图例	503
已知打印问题的解决方案	506
平台独立问题	506
特定平台问题	507
附录 F: 样式查找	510
填充图案	511
了解索引编号结构	511
直线样式	525
矢量符号	525
MapInfo 箭头	526
MapInfo 制图	526
MapInfo 杂类	527
MapInfo 油气	527
MapInfo 遮罩	527
MapInfo 房地产	528
地图符号	528
MapInfo 符号	528
MapInfo 交通	529
MapInfo 天气	529
定制符号	530
MapXtreme 图标	532
附录 G: 定义 MapInfo Codespace	535
定义 MapInfo Codespace	536
附录 H: 坐标系的元素	541
投影及其参数	542
投影 (Projection)	543
投影基准面	546
单位	555
坐标系原点	556
基准面转换	557
定制基准面	558
定义定制基准面	558
National Transformation v.2 (NTv2)	563

关于坐标系和投影的信息	565
附录 I: 用户定义的元数据	567
元数据和 MapCatalog	568
支持 TableInfoServer 查询的用户定义的元数据	568
ColumnHints 属性	568
附录 J: 迁移到 MapXtreme	571
比较 MapXtreme 和 MapX 的对象模型	572
特定对象模型实现的不同之处	572
附录 K: 本地化工具包	582
本地化工具包	583
系统要求	585
如何使用本地化工具包	585
创建卫星程序集	586
从命令行进行构建	586
卫星程序集的私有钥匙标记	587
附录 L: 术语表	588
术语	589

MapXtreme 简介

欢迎开发人员使用 Pitney Bowes Software 最新提供的 .NET 编程技术。为了支持 Microsoft 公司的 Windows .NET 框架，MapXtreme 打造了单一的对象模型，可用于开发或扩展桌面、传统客户机/服务器环境或 Web 方式地图绘制应用程序。

对于那些已经认识到数据可视化和地图绘制有助于制定更好的商业决策、管理资产以及更有效地运作的公司，MapXtreme 是一个功能强大的应用程序开发工具。MapXtreme 设计用于需要将位置分析或定义放入桌面、客户机/服务器或基于 Web 的产品的公司。MapXtreme 是一个功能强大的分析工具包，可以用来制定关键的商业决策，例如销售办事处的最佳位置、如何最有效的运输产品以及如何管理和保护资产。开发人员可以使用 MapXtreme 来缩短开发时间并提高产品的性能、可靠性以及安全性。

在本章中：

- ◆ [MapXtreme 概述](#) 23
- ◆ [迁移到 MapXtreme](#) 25
- ◆ [学习使用 MapXtreme](#) 27

MapXtreme 概述

MapXtreme 是 Pitney Bowes Software 的主要 Windows 软件开发工具包，具有 .NET 开发经验的开发人员使用该工具可以创建功能强大的位置增强型桌面和客户机/服务器应用程序。

开发人员可在这个 SDK 中使用熟悉的 .NET 编程语言开发应用程序，在桌面和 web 部署之间共享和重用代码，使用标准协议访问大量数据源中的数据以及更多其它功能。

这些均可通过 MapXtreme 的对象模型实现，该对象模型是在 Microsoft 的 .NET Framework 上开发的完全受管代码 API。Framework 的 Common Language Runtime (CLR) 提供了实现简化开发的基础。

MapXtreme 包括以下组件和功能：

- **产品框架：**MapXtreme 对象模型使用 Microsoft .NET Framework 4.5.2 构建而成。有关详细信息，请参阅[对象模型概述](#)。
- **开发环境工具：**Visual Studio 中提供的大量模板、控件、示例代码和工具有助于开发 Windows 窗体和 ASP.NET 应用程序。您可以利用对象模型将这些组件进行扩展，从而获得更高级的功能。
MapXtreme 2004 包括了两个数据管理实用程序，Geodictionary 管理器用于管理将在应用程序中使用的表，而 Workspace 管理器用于管理工作空间以方便用户使用和移植。请参阅[第 5 章：Web 应用程序、控件和工具](#)和[第 7 章：桌面应用程序、控件、对话框和工具](#)。
- **强大的地图绘制和分析能力：**地图创建和显示、数据访问、主题图地图绘制、光栅和网格处理、对象处理和表示等等。
- **可伸缩的基础结构：**Session 对象池和缓存功能可使 web 应用程序获得出色的性能。通过将信息保存为 MapXtreme 的基于 XML 的工作空间格式来维护会话和用户信息。请参阅[第 9 章：运用核心 MapXtreme 类](#)。
- **运行时部署：**MapXtreme 使用 Windows Installer 技术（合并模块），开发人员可以使用该技术来安装或重新发布在已部署应用程序中使用的运行时组件。请参阅[部署应用程序](#)。
- **丰富的文档：**产品文档已作为组件完全集成在 Visual Studio 开发环境中。MapXtreme 学习资源页面让您能够随时访问所有产品资源，包括此版本中的新特性和改进功能。可以通过“开始”菜单访问该页面。

主要功能

MapXtreme 可以帮助您方便有效地构建 Windows 窗体或 ASP.NET Web 应用程序。不论是设计应用程序的基本绘制地图功能还是增加基本地图绘制功能以支持现有的应用程序，所使用的框架和工具都是相同的。以下是 MapXtreme 功能的概述：

如果您是 Pitney Bowes Software 地图绘制产品的新用户，请务必参阅[第 3 章：地图绘制概念](#)以获取基础知识的帮助信息。

对于升级到 MapXtreme 的开发人员，请参阅发行说明以了解本产品中的新增内容和变化。另请参阅[迁移到 MapXtreme](#) 以了解 MapX 特性与 MapXtreme .NET 特性的对应关系。

Feature *	目的
表、图层、图元	<p>MapXtreme 中的地图由地理图元（例如点位置、边界和街道网络）组成。图元信息存储在表中并在地图中显示为图层。</p>
数据访问	<p>MapXtreme 支持来自多种数据源的数据，包括空间和非空间 RDBMS、MS Access、dBase 和 ASCII 以及自带的本地 MapInfo Table (.TAB)。所有数据操作都通过 MapInfo.Data namespace 执行。数据操作包括添加和删除表，插入、更新和删除来自各种数据源的记录。</p> <p>.NET Dataset Provider 支持：允许将任何 ADO.NET 数据集提供程序视作 MapInfo.Data 中的表予以处理。这样，您可以使用外部不可制图的数据。</p>
Web 服务	<p>MapXtreme 为您提供客户端和 API，方便您访问几种流行的 Web 服务：地理编码、路径规划、WMS 和 WFS。</p>
选择和搜索	<p>该公共地图绘制操作通过使用属性或空间查询，可以找到与条件相匹配的数据。</p>
主题地图绘制	<p>对数据进行分析是最常使用的一种方法，可以通过主题地图显示可视的关系和基础数据。MapXtreme 支持创建和使用六种主题：范围、单值、分级符号、点密度、饼图和条形图。</p>
标注	<p>MapXtreme 提供了成熟的标注功能，不仅可以使用名称或其它信息（列数据或表达式）对图元进行标注，还可以在标注本身创建范围和独立值主题图，用来表示信息而不必仅依赖于文本。</p>
地图样式	<p>标注是其中一种地图样式，可以在地图上以任意方式对地图样式进行控制。样式还指地图图元的颜色、图案、字体、直线样式和符号，MapXtreme 中的许多地方（包括图元、修饰即地图标题、对话框和文本）都使用这些样式。</p>
地理处理和分析	<p>该功能是指从现有图元中产生新的图元，例如将邮政编码边界进行组合从而创建销售区域。该功能还指使用图元的位置坐标了解更多与其它图元关系的信息。例如，在点周围半径 5 英里范围内创建缓冲区以找出落在该缓冲区内的其它点。</p>
投影和坐标系	<p>在二维地图上可以有多种方式来表示地点。了解数据的坐标系让您正确地排列图元，从而获得精确的显示和度量。MapXtreme 支持多种投影和坐标系，并为创建自己的投影和坐标系提供了信息。</p>

* 对于个别语种或区域，可能不支持本产品中的部分功能或工具。详细信息请与当地的客户服务代表联系。

迁移到 MapXtreme

下表将 MapX 和 MapXtreme v3.0 的特性和功能与 MapXtreme .NET 的特性和功能进行了比较。使用产品的任何新体系结构时，注意对等的功能可能不精确。使用该列表中右边一列的 MapXtreme 主题可以了解进一步的详细信息，这些信息位于本《开发人员指南》和联机帮助以及对象模型的其它章节中。MapX 对象模型和 MapXtreme 中等价功能的详细列表位于[附录 J：迁移到MapXtreme](#)。

MapX5.0 /MapXtreme3.0	MapXtreme
Map 对象	Map 类: 保存图层集合。 MapControl: 查看窗体上的地图的方法。 MapInfo.Mapping 命名空间
MapXBroker	Session 类: 所有基于 MapXtreme 的应用程序的起点。 相关主题: MICommand、Catalog、Pooling MapInfo.Engine 命名空间
以图层为中心的模型	以表为中心的模型 相关主题: 表元数据 (TableInfo 类)、Feature 类、Column 类 (MI_Geometry、MI_Style 和 MI_Key)、MapInfo ADO.NET 数据提供程序。 MapInfo.Data 命名空间
数据集和数据绑定	使用 Table.AddColumn() 方法将临时列添加到 Table 中。 相关主题: Geodictionary Manager MapInfo.Data 命名空间
Geosets Geoset Manager	工作空间 (.MWS): XML 格式。支持 Geoset。 相关主题: Workspace 管理器。 MapInfo.Persistence 命名空间
Annotations	修饰: 同一地图中 Legend、Title 和 Scalebar 或其他类似的用户定义对象。 MapInfo.Mapping 命名空间

MapX5.0 /MapXtreme3.0	MapXtreme
主题地图绘制	<p>与主题地图类型相同。主题不再是图层。</p> <p>相关主题: ModifierTheme (分级符号、饼图和条形图主题)、对象主题 (范围、单值、点密度主题)。</p> <p>MapInfo.Mapping.Thematic 命名空间</p>
图元图层和图元集合	<p>FeatureGeometry: 全部几何体现在都是对象。其中包括 point、multipoint、curve、multicurve、polygon、multipolygon、ring。</p> <p>几何体包括矩形、有圆角的矩形、椭圆、旧式的弧线和旧式的文本。</p> <p>相关主题: CoordSysFactory (已注册的坐标系)、对象处理 (请参阅下面的 FeatureProcessor)</p> <p>MapInfo.Geometry 命名空间</p>
FeatureFactory	<p>FeatureProcessor 类: object processing Buffer、Combine、Intersection、ConvexHull。</p> <p>相关主题: FeatureGeometry 类中的 Difference (先前的 Erase)。</p> <p>MapInfo.Geometry 命名空间</p>
工具	<p>可为桌面工具 SelectRegion 分配鼠标按键并配置鼠标滚轮使之具备缩放功能。</p> <p>MapInfo.Tools 命名空间</p>
光栅和网格图像	<p>新的表结构、RasterImageInfo 和 GridImageInfo。</p> <p>相关主题: 可控制的样式: 亮度、对比度、颜色/灰度、半透明度和单色透明度。</p> <p>MapInfo.Raster 命名空间</p>
标注对象和标注集合	<p>LabelLayer: 允许对标注和图层进行单独排序。LabelSource: 来自用于进行标注的数据源的信息。</p> <p>MapInfo.Mapping 命名空间</p>
Selection 对象	<p>Selection 类: 图元列表的多图元集合, 连接到表。</p> <p>MapInfo.Engine 命名空间</p>

MapX5.0 /MapXtreme3.0	MapXtreme
Spatial Server 连接	MI ADO.NET 数据提供方, MapInfo SQL 语言。 MapInfo.Data 命名空间
Style 对象	Style 类: 新对象模型。Style 现在成为对象, 不再是其它对象的属性, 信息存储在列 MI_Style 中。 相关主题: MapInfo.Mapping 命名空间中的 FeatureStyleModifiers 和 FeatureOverrideStyleModifiers。 MapInfo.Styles 命名空间

学习使用 MapXtreme

MapXtreme 为入门和经验丰富的 .NET 开发人员提供了大量知识。本节描述了多种支持方式, 使用户可以迅速进入角色, 将对开发日程表的影响降至最低。

支持资源

MapInfo 致力于为新的 MapInfo 开发人员和长期客户提供支持服务。我们提供了多种工具和以下资源, 可以帮助您进行转换或在 Visual Studio 环境中迅速展开工作。

MapXtreme 学习资源页面

安装 MapXtreme 后可以通过“开始”菜单访问, “学习资源”页面集中了关于 MapXtreme 的各类信息, 其中包括最佳开发实践、代码示例、教程 Web 应用程序以及指向 MapInfo 网站上所有文档和在线资源的链接等。访问该页面能够使您逐步熟悉 MapXtreme 并帮助您继续开发适应业务需要的地图绘制应用程序。

文档和帮助系统

MapXtreme 开发人员指南概述了 MapXtreme 开发环境和命名空间。Visual Studio 集成的帮助系统提供了更多特定的 API 级别的信息。用户需要使用这些工具开发与 MapInfo 强大的地图绘制组件集成的 Windows 桌面和基于 Web 的应用程序。请将对本文档的意见发送至 software.support@pb.com。

-
-  上述电子邮件地址不适用于特定于软件的问题, 以及有关包含在本文档中的主题的问题。请将这种问题发送至“技术支持”(见下文)。
-

如果您刚开始使用 MapInfo 地图绘制产品, 则一定要阅读第 3 章: 地图绘制概念。

要获取最新的版本信息, 请务必从 [Pitney Bowes Software](#) 网站下载 MapXtreme 发行说明的副本。

技术支持

Pitney Bowes Software 为 MapInfo 软件产品的用户提供了强大的技术支持。我们的“技术支持”部门为已注册的 MapInfo 软件用户提供技术上的帮助 — 因此不需要精通我们产品的方方面面就可以获得所需的结果。有关技术支持的信息，请访问 Pitney Bowes Software 网站 www.pb.com/software。

入门指南

本章提供了首次安装、配置和部署 MapXtreme 应用程序所需的全部信息。

在本章中：

◆ 安装要求	30
◆ 安装类型	31
◆ MapXtreme 许可	32
◆ 安装之前	38
◆ 将 MapXtreme 安装到系统中	40
◆ 升级到 MapXtreme	43
◆ 将 Web 站点迁移到 64 位 Web 应用程序中	43
◆ 在 Visual Studio 中创建应用程序	46
◆ 构建 ASP.NET Web 应用程序但不使用模板	50
◆ 部署应用程序	52

安装要求

Pitney Bowes Software Inc.已经过测试，支持 MapXtreme 的以下方面。

体系结构	<ul style="list-style-type: none">• 64 位• 32 位
操作系统	<ul style="list-style-type: none">• Windows 8 和 8.1 (x64)• Windows 7 Ultimate (x86, x64)• Windows Server 2012 和 2012 R2 (x64)• Windows Server 2008 R2 64• Windows Server 2008 with SP2 (x86 , x64)
开发框架和 IDE 支持 *	<ul style="list-style-type: none">• Microsoft .NET Framework 4.52• Visual Studio 2013• Visual Studio 2012 Upgrade 4
浏览器	<ul style="list-style-type: none">• Internet Explorer 8.0 及更高版本**• Firefox 3.5 及更高版本• Chrome 20 及更高版本
对于 Web 应用和部署：	<ul style="list-style-type: none">• IIS 8 及更高版本（Windows 8/8.1、Windows Server 2012 R2）• IIS 7（Windows Server 2008 和 Windows 7）
支持的数据库	<ul style="list-style-type: none">• Microsoft Access 2007 和 Excel 2007• Microsoft Access 2003• Oracle 11G (11.1.0.6.0 和 11.1.0.7.0)• Oracle 10G、10GR2,• Microsoft SQL Server 2012 (配有 SQL Native Client 11)• Microsoft SQL Server 2008 (配有 SQL Native Client 10)• MapInfo SpatialWare® 4.9/4.9.2 for Microsoft SQL Server 2005• MapInfo SpatialWare® 4.9.2 for Microsoft SQL Server 2008 (32 位版和 64 位版)
对于数据访问：	<ul style="list-style-type: none">• MDAC 2.8

* 推荐的开发环境 (IDE)。使用其他开发环境时，MapXtreme 安装程序不会将其模板、示例和帮助系统集成到开发环境中。

** 无论您的默认浏览器设置如何，MapXtreme 学习资源页面都将自动显示在 Internet Explorer 中。这不会更改您的默认浏览器设置。

i MapXtreme 不支持 Microsoft Visual Studio 的 Express Edition。

i MapXtreme 不支持 Borland Delphi 开发框架，也不支持 Cold Fusion 或 HAHT Web 开发框架。

最低系统要求

内存	Windows 7 和 8: 1 GB 内存 (32 位), 2 GB 内存 (64 位) Windows Server 2012 R2: 1 GB 内存 Windows Server 2008: 512 MB 内存
处理器	Windows 8、Windows 7: 1 GHz 处理器 Windows Server 2008/2012: 1 GHz 处理器
显卡	支持至少 256 色的显卡

安装类型

MapXtreme 提供两种安装类型：一种用于开发环境 (SDK)，一种用于部署环境 (Runtime)。每种类型都可以从产品光盘浏览器中选择。

开发 (SDK) 安装

“开发安装”在计算机上安装 MapXtreme 软件开发工具包 (SDK)。选择此选项，用于开发桌面和 web 应用程序。安装后，该 SDK 将自动集成到 Microsoft Visual Studio 并与 .NET Framework 结合使用。SDK 提供了用于简化开发工作的 C# 和 VB 应用程序模板。

有关如何安装 SDK 的说明，请参阅将 [MapXtreme 安装到系统中](#)。也可从安装对话框上的“帮助”按钮获得指示。

部署（运行时）安装

“部署安装”选项安装“位置运行时环境”，它放置 MXTRuntimeNCP.exe（或软件防复制保护版本的 MapXtreme 的 MXTRuntimeSCP.exe）中。有关安装运行时安装程序的说明，请参阅[用运行时安装程序部署](#)。

并行安装和使用

可以同时将多个版本的 MapXtreme 安装到系统上。每个版本的 MapXtreme 都会安装到各自的目录下。

还可以针对较早版本的 MapXtreme 构建桌面和 Web 应用程序并在后续版本上运行。

如果 ASP.NET 应用程序是由不同版本的 MapXtreme 构建的，则可以在同一台计算机上运行多个 ASP.NET 应用程序。请为每一种版本的 MapXtreme 创建应用程序池并将相应的 ASP.NET 应用程序放入其中。在命令提示符下键入 *iisreset* 以重新启动 IIS，或回收应用程序指定的应用程序池。当应用程序在其自身的进程空间中运行时，将会加载相应的 MapXtreme 版本。

这并不会影响使用其他版本的 MapXtreme 创建的桌面应用程序。每个桌面应用程序始终在其自身的进程空间中运行。

MapXtreme 许可

所有的 MapXtreme 安装都需要一个许可证 (.lic) 文件。某些许可在安装中提供，而其它许可必须从 Pitney Bowes Software Inc. 获取。以下主题介绍了许可类型以及如何从 Pitney Bowes Software 获取一个许可。

许可证类型

根据您的目的和需求，有三种类型的许可证可用于 MapXtreme — SDK（开发人员）、运行时（部署）和试用许可证。

SDK 许可证

开发人员可通过 SDK（软件开发工具包）许可证 (MapXtremeSDK.lic) 创建桌面和 Web 应用程序。使用此许可证开发的应用程序将在地图窗口左上角显示一个小的水印。

运行时许可证

所有运行时桌面和 Web 安装都需要运行时或部署许可证。

- 对于使用 NCP 版的 MapXtreme 构建的桌面运行时部署，该许可证（例如，MapXtremeDesktop.lic）可在多个安装中使用。
- 对于使用 SCP 版的 MapXtreme 构建的桌面运行时部署，每个安装都需要一个许可证（例如，MapXtremeDesktopIntl.lic）。
- 对于使用 NCP 或 SCP 版的 MapXtreme 构建的 Web 运行时部署，每个安装都需要一个许可证（例如，MapXtremeWeb.lic）。

有关在分发中打包许可证的详细信息，请参阅[用于部署的运行时许可证](#)。

试用许可证

试用许可证 (**MapXtremeTrial.lic**) 在 SDK 和 MapXtreme 的评估版本中提供。该许可证在安装之日起 60 天后过期。使用试用许可证开发的应用程序将在地图上显示一个水印，该地图还可指示使用许可证的剩余试用天数。

获取许可证

用于开发的 **SDK** 许可证

对于没有安装复制保护的软件开发工具包 (**NCP SDK**) 的用户，将会在进行安装时自动收到永久 **SDK** 许可证 (**MapXtremeSDK.lic**) 以及试用许可证。有了永久许可证后只要愿意，您就可以一直使用 **SDK**。不再需要其他 **SDK** 许可证。

对于安装了复制保护的 **SDK (SCP SDK)** 的用户，则只能收到试用许可证。要永久使用 **SDK**，必须从 **Pitney Bowes Software** 获取永久 **SDK** 许可证。如要执行此操作，请运行 **MILicenseNodeID.exe**（可在 **Program Files\Common files\MapInfo\MapXtreme\7.x.x\License Tool** 下找到）以生成特定于机器的唯一硬件 ID，该 ID 包含在名为 **MILicenseNodeID.txt** 的文本文件中。**Pitney Bowes Software** 使用此信息为您的特定安装生成自定义许可证。有关获取许可证的完整详情，请参阅产品包装盒中的《软件复制保护声明》，也可参阅以下目录中的文件：**Program Files\MapInfo\MapXtreme\7.x.x\Documentation\PDF\SoftwareCopyProtectionNotice_SC_P.pdf**。请根据《声明》中您所在地区的相应电子邮件或传真号联系 **Pitney Bowes Software**（或其代表）。当您收到永久许可后，请在首次使用部署的应用程序之前先将该许可复制到 **Program Files\Common Files\MapInfo\MapXtreme\7.x.x** 文件夹中。

评估版用户需要购买产品后才能收到永久许可证。

用于部署的运行时许可证

MapXtreme 要在每个部署中包括运行时许可证。要执行该操作，则必须在程序包中提供许可或包括客户可用于获得自身许可的许可工具。运行时许可证可从 **Pitney Bowes Software** 获取。此许可证类型还可以从地图中删除所有的水印。

需要的运行时许可证类型取决于正在生成的应用程序类型和使用的 **MapXtreme** 的版本。在下表中查看具体情况：

部署类型	MapXtreme 的 SDK 版本	节点锁定？	如何获得运行时许可证
桌面	无复制保护 (NCP)	否	联系客户服务部门以获得一个或多个桌面的运行时许可证。
桌面	复制保护 (SCP)	是	为每个桌面生成节点 ID 文件并提交给客户服务部门。

部署类型	MapXtreme 的 SDK 版本	节点锁定？	如何获得运行时许可证
Web	NCP	是	为每个桌面生成节点 ID 文件并提交给客户服务部门。
Web	SCP	是	为每个桌面生成节点 ID 文件并提交给客户服务部门。

要生成部署计算机的节点 ID，请运行许可证工具 **MILicenseNodeID.exe** 并将生成的文本文件提交给客户服务部。许可证工具与 MapXtreme SDK 一起安装在 **Program Files\Common Files\MapInfo\MapXtreme\7.x.x\License Tool** 中。请参阅 MapXtreme 程序包中的《软件复制保护》许可证应用程序文档或参阅以下目录中的文件：
Program Files\MapInfo\MapXtreme\7.x.x\Documentation\PDF。

许可证工具位于合并模块 **MapInfoMXTConfig_7.x.x.msm** 中。将 **MapInfoCoreEngine.msm** 包括在分发中时，您的用户将自动获得该工具。请参阅[使用运行时安装程序部署应用程序的步骤](#)。

① 通常，Web 许可证不能用于桌面部署，反之亦然。如果是 SCP 版本，则许可证（Web 或桌面）对于您正在部署的计算机是节点锁定的。

当联系 Pitney Bowes Software 以获取许可证时，请指明 MapXtreme 的版本。

许可证文件的位置

使用下面的图表可确定您的许可证位于何处，应该将其放置在何处：

许可证类型	许可证名称	许可证位置
开发人员		
SDK NCP	MapXtremeSDK.lic	Program Files\Common Files\MapInfo\MapXtreme\7.x.x
SDK SCP	MapXtremeSDK.lic	Program Files\Common Files\MapInfo\MapXtreme\7.x.x
运行时		
桌面 NCP	MapXtremeDesktop.lic	在与应用程序可执行文件相同的文件夹中，以免与其他应用程序出现潜在的冲突
桌面 SCP	MapXtremeDesktopIntl.lic	在与应用程序可执行文件相同的文件夹中，以免与其他应用程序出现潜在的冲突

许可证类型	许可证名称	许可证位置
Web NCP	MapXtremeWeb.lic	Program Files\Common Files\MapInfo\MapXtreme\7.x.x
Web SCP	MapXtremeWeb.lic	Program Files\Common Files\MapInfo\MapXtreme\7.x.x
试用		
SDK NCP	MapXtremeTrial.lic	Program Files\Common Files\MapInfo\MapXtreme\7.x.x
SDK SCP		
评估		

故障诊断

如果在使用 MapXtreme 许可时遇到问题，请使用下表来帮助分析和解决问题。对于任何问题，总是检查正确的许可证是否安装在正确的位置（请参阅[许可证文件的位置](#)）。

许可问题的故障诊断

问题说明	可能原因	解决方案
“许可错误 - 无法初始化 MILicensing 对象。”	当 MapXtreme 许可服务未运行时会出现此情况。 64 位应用程序的 MapXtreme 许可要求在本地应用程序主机上运行 Windows 服务。会在 MapXtreme 安装过程中自动安装该服务，将使用 Services.msc 管理 Windows 服务。	验证安装的 MILicensingService 的状态为“已启动”。
“没有找到 MapXtreme 200x 的有效许可证”	可能在使用远程桌面时出现。需要一个新的具有激活了终端服务的许可证。	重置服务器，将新的许可证放在正确的目录中，然后重新启动 IIS 和 ASP.NET 工作进程。
	许可证文件损坏	从计算机中删除许可证，重新生成许可证，将新的许可证放在正确的目录下，然后重新启动 IIS。
	需要权限	确保 ASP.NET 用户帐户具有对 Temp 目录和 Framework 目录的“读/写”访问权限（请参阅 已部署 Web 应用程序需要访问临时目录的权限 ）。
COM+ 激活错误	通常是错误的或损坏的许可证文件。它可防止 COM+ 对象进行初始化，因此在 COM+ 中会导致低级别的操作系统故障。	从计算机中删除许可证，重新生成许可证，将新的许可证放在正确的目录下，然后重新启动 IIS。
如使用从 Pitney Bowes Software 网站下载的试用版，则可能出现许可证异常（mapinfo.coreengine.dll 中出现的 System.ComponentModel.LicenseException）。	因为检测到系统时钟被拨前，MapXtreme 试用许可证无效。	在不同的计算机上安装试用版或购买一个有效的许可证。

许可问题的故障诊断（续）

问题说明	可能原因	解决方案
试用版中的合并模块 (MSM) 丢失 (\Common files\Merge Modules)。	由于试用版并不意味着可以进行部署，所以试用版中不包括合并模块。	请购买一个运行时许可证。
卷影复制服务可让许可证不被找到。	与 Veritas 备份相关。有可能根据备份软件更改网络。备份进程是锁定文件的过程，可能不允许访问。	检查备份进程是否正在按与 Exponare 进程相同的 ID 运行。如果是，则终端服务可能存在问题。新的许可证将能够解决此问题。
MapXtreme 文件夹（来自以前版本）可让有效的许可证不被找到。	无效的许可证文件可能出现在旧目录中，将导致故障。	查找和删除旧的无效许可证。将新的许可证放到正确的目录中。
使用 Exponare 分发的 MapXtreme 许可证可解锁一些站点，但不能解锁另一些站点。	系统中出现多张网卡。	获取正确的许可证文件。将新的许可证放到正确的目录中。
许可证文件如果没有同时位于 MapXtreme\7.x.x 文件夹和 Web 应用程序文件夹中，则无法解锁 MapXtreme。	有关如何运行服务器和使用哪些用户 ID 的权限错误。	设置正确的权限（请参阅 已部署 Web 应用程序需要访问临时目录的权限 ）。
MapXtreme 许可证文件将大约在 15 日之后过期。	终端服务	确保您正在使用可提供终端服务的最新许可证生成器内部版本。
MapXtreme 许可证节点 ID 发生更改，并在虚拟 PC 移动时导致 Exponare 许可证还原到试用许可证。	移动虚拟 PC 会更改网络节点 ID，这样会使许可证失效。	获取新许可证
在安装和第一次使用之后，用户将收到一条错误消息“您的 MapXtreme 试用许可证已过期”。	安装生成的运行时（部署）许可证，而不使用已有的 NCP SDK 许可证。	使用随 NCP SDK 提供的永久 SDK 许可证。

已知的许可问题

下面是已知的 MapXtreme 许可问题。请联系技术支持来帮助您处理这些问题（请参阅[技术支持](#)）。

- 防病毒软件会删除许可证电子邮件中的内容，并在该许可证文件中留下 Null。
- 有时，许可证生成器会创建损坏的许可证文件。

安装之前

下面是安装 MapXtreme 之前需要了解的内容。

管理员特权

要安装 MapXtreme，您必须是计算机的管理员，或者当前用户必须是 Administrator 组的成员。这既适用于 SDK 安装也适用于运行时安装。

-
- ① 对于 Windows Vista、Windows 7 和 Windows Server 2008，常规的管理权限不足，因为安装程序必须注册 COM+ 对象。Vista 用户必须右键单击 Setup.exe 并选择“用管理员帐户运行”来获取运行该安装程序的系统权限。
-

首先安装 .NET Framework 和 Visual Studio

在安装 MapXtreme 之前，确定您安装了 .NET Framework 和适用于此框架的 Visual Studio 环境。

-
- ① 您还可以使用 Visual Studio 之外的其他部署环境，但不会将模板、示例和帮助系统集成到环境中。
-

支持 IIS 7/8

MapXtreme 支持 Internet Information Services (IIS) IIS 7 和 IIS 8 下的 web 部署。

对于 IIS 7，要求操作系统为 Windows Server 2008 或 Windows 7。对于 IIS 8，要求操作系统为 Windows Server 2012 或 Windows 8。MapXtreme 不支持 Windows XP 中的 web 部署。

《开发人员指南》中所提及的 IIS 是指 IIS 7/8。

Windows Server 2008 和 Windows 7 Ultimate 中提供了 IIS 7（尽管非必需安装）。MapXtreme 支持经典模式和集成管线模式的 IIS 7 和 IIS 8。

安装 MapXtreme 之前，请按以下步骤配置 IIS 7/8。这些步骤适用于 Windows Server 2008 和 Windows 7。

1. 启用 Windows 身份验证和匿名身份验证。
 - a. 转到“控制面板”>“管理工具”，右键单击 IIS，然后选择“用管理员帐户运行”。
 - b. 选择“默认 Web 站点”。
 - c. 在 IIS 组下，双击“身份验证”。
 - d. 右键单击“匿名身份验证”，然后选择“启用”。对“Windows 身份验证”执行同样的操作。
2. 启用“Web 管理工具”。
 - a. 转到“控制面板”>“程序和功能”。

- b. 单击“打开或关闭 Windows 功能”。此时将打开“Windows 功能”对话框。
 - c. 选中“Internet 信息服务”复选框。
 - d. 双击（或展开）“Web 管理工具”，然后选中其下面的所有复选框。
3. 启用 World Wide Web 服务。
- a. 在“Windows 功能”对话框中，双击（或展开）“World Wide Web 服务”，然后选中下面列出的复选框。
 - b. 应用程序开发功能 - 全部选中。
 - c. 常用 HTTP 功能：默认文档、目录浏览、HTTP 错误、静态内容和 WebDAV 发布
 - d. 健康状况和诊断：HTTP 记录、请求监视
 - e. 性能：静态内容压缩
 - f. 安全性：请求筛选和 Windows 身份验证

MapXtreme Web 控件和 IIS

MapXtreme 的 Web 控件始终会自动修改 ASP.NET 的 web.config 文件，以包含所需的模块和处理程序。将完全支持 IIS7 集成管线模式，还将自动修改 web.config 文件以在 system.webServer 节点下包含必需的代码。为了保持与之前 ASP.NET 应用程序的兼容性以便在 IIS6 或 IIS7 的“经典”管线模式下运行如果将 MapInfo.Engine.Session.PipelineMode 属性设置为“集成”，则只能在 web.config 文件中输入集成的管线代码。当将任何 ASP.NET 项目加载到系统上安装了 MapXtreme 的 Visual Studio 中时，都会将此属性添加到 web.config 文件的“appSettings”节点中。最初，会注释掉此属性。只需取消注释进行适当的编辑，即可使应用程序在 IIS7 的集成管线模式下运行。将值更改为“经典”，或者只是重新注释该属性，以便注释掉 system.webServer 节点，以实现与经典管线模式的向后兼容。

MapXtreme 的默认安装目录

MapXtreme 是 64 位应用程序，默认情况下，会安装在 C:\Program Files (64 位计算机) 或 C:\Program Files (x86) (32 位计算机) 下。

这两个默认路径都包含在 MapXtreme 示例应用程序的 web.config 文件中。如果您将 MapXtreme 安装到其他位置，必须编辑 Web.config 文件才能指向让示例正确运行的位置。



安装的 Windows 如果不是美国英语 (ENU) 版，MapXtreme 会将默认安装目录 (C:\Programmers\) 视为自定义安装位置。您必须编辑示例 Web.config 文件来指向您的安装目录，如下面的示例所示。

```
<configuration>
<appSettings>
<add key="MapInfo.Engine.Session.Workspace"
     value="C:\Programmer\MapInfo\MapXtreme\7.x.x\Samples\Data\World.mws"
/>
</appSettings>
</configuration>
```

其它安装特性

MapXtreme 提供了需遵循的联机安装指示。也可以在安装过程中的安装对话框上通过“帮助”按钮访问这些指示。

MapXtreme 提供了大量与各个世界位置有关的样本数据。要安装这些数据，请选择“从光盘浏览器安装样本数据”。通过选择“定制”选项，可以控制要安装的数据量。如选择“完全安装”选项（默认），文件夹 (Program Files\MapInfo\MapXtreme\7.x.x\Samples\Data) 下将安装大约 450 MB 的世界数据集。

-
- i** 无需运行该数据安装程序就可以使用 MapXtreme 包含的示例应用程序。为实现该目的，会将基本示例数据自动安装到 \Data 文件夹。
-

MapXtreme 光盘浏览器还提供了到 PDF 版《开发人员指南》的链接。

将 MapXtreme 安装到系统中

安装 MapXtreme：

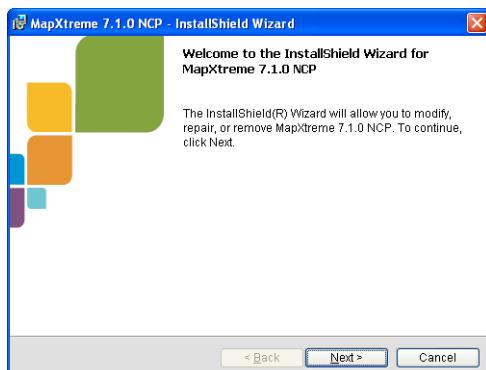
1. 将 MapXtreme 产品介质置于磁盘驱动器中。
2. 在光盘浏览器主页面，单击安装。此时将显示“安装说明”页面。
3. 选择开发安装以安装 SDK，或选择部署安装以安装运行时版本。

-
- i** “部署”安装需要运行时许可证。请参阅[用于部署的运行时许可证](#)。
-

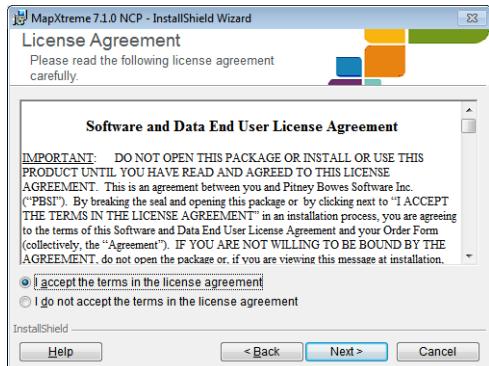
4. 选择“安装 SDK”或“安装运行时”。在“欢迎”对话框中，单击下一步继续。对于部署安装，则跳到**步骤 10**。

-
- i** 也可以查看安装指示并从该页面安装样本数据。
-

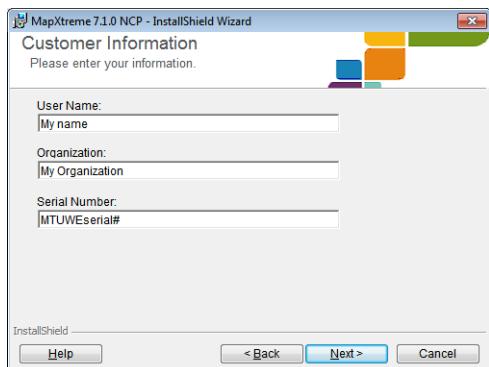
5. 在“安装程序欢迎”对话框中，阅读面板中的信息并单击下一步以继续。



6. 选择接受“许可协议”。单击下一步。此时将显示“客户信息”对话框。

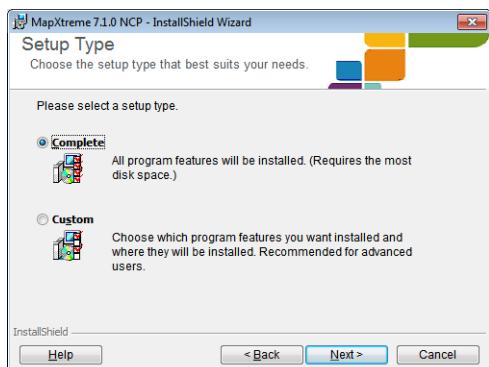


7. 在“客户信息”对话框，将用户名和公司名输入适当的字段，如有必要，还需要输入位于产品包装上的序列号。单击下一步。此时将显示“安装类型”对话框。

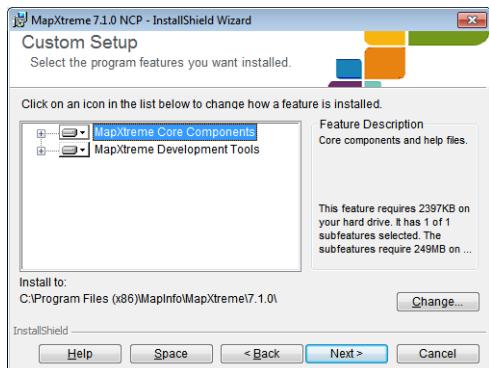


8. 在“安装类型”对话框，选择完全安装或自定义。如要安装特定功能或将功能安装到除默认位置 (C:\Program Files\MapInfo\MapXtreme\7.x.x) 之外的目录，请选择“自定义”。单击下一步。如果选择“完全”，则继续步骤 10。如果选择“自定义”，则继续步骤 9。

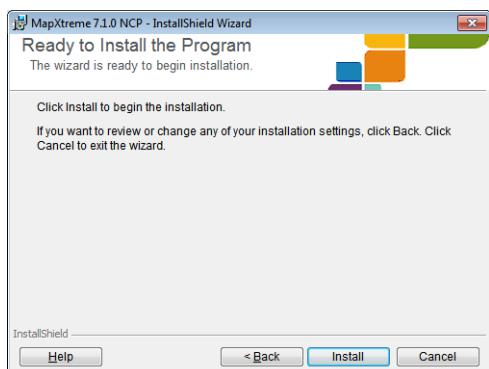
i 如果将 MapXtreme 安装到非默认位置或非英文 (ENU) 版的 Windows XP 时，则需要编辑想要在安装后运行的所有样本 Web 应用程序的 web.config 文件。有关编辑 web.config 文件的指示，请参阅示例 Web 应用程序的 ReadMe.rtf 文件。



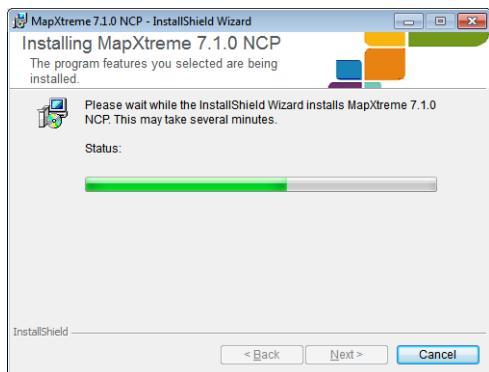
9. 如果在上一步中选择了“自定义”，则选择要安装的组件，或单击更改按钮来指定新的安装路径。单击下一步。



10. 在“准备安装程序”对话框中，单击安装。

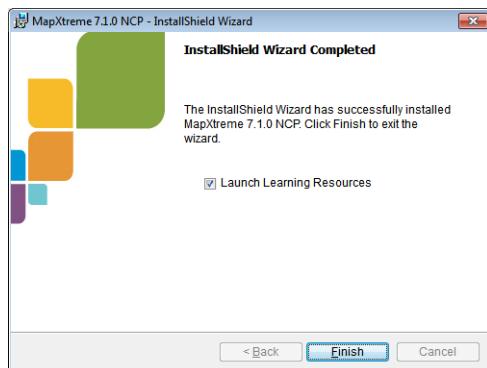


11. “安装 MapXtreme”对话框启动。



12. 如果不想查看 MapXtreme 学习资源页面，请在 InstallShield Wizard 的“完成”对话框中取消选中启动学习资源复选框，然后单击完成退出软件安装程序。

如果采用的是 SDK 安装，“MapXtreme 学习资源”页面随时都可以通过 Windows“开始”菜单进行访问。它不可用于运行时安装。



升级到 MapXtreme

MapXtreme 安装到自己的目录中，其格式为 \<installdir>\MapInfo\MapXtreme\7.x.x.x，此处 X.x.x 是当前版本号。它不会覆盖先前版本。这样就可以同时安装有不同发行版本的产品。

对于并行安装，请将原许可证文件复制到最新安装的位置
(Program Files\Common Files\MapInfo\MapXtreme\7.x.x)。

注意，使用并存安装时，必须在打开不同类型项目时关闭 Visual Studio。对于 Web 应用程序，还需重置 Internet 信息服务 (IIS)。在控制台窗口中或在“开始 > 运行”菜单选项中执行 **iisreset.exe** 命令。

将 Web 站点迁移到 64 位 Web 应用程序中

MapXtreme 支持创建 64 位 Web 应用程序。如果有一个现有的 Web 站点，您需要将它们迁移到某个 Web 应用程序以便利用 64 位处理功能。您可以继续构建基于 MapXtreme 的 Web 站点，该站点为 32 位站点（请参阅第 43 页中的 [更新现有的 Web 站点](#)）。

将 Web 站点迁移到 Web 应用程序中时，没有 MapXtreme 所特有的内容。有关详细信息，请参阅以下 Microsoft 主题：[Web 应用程序项目与 Web 站点项目](#)，然后转至：[转换 Web Visual Studio 中的站点项目和 Web 应用程序项目](#)。

更新现有的 Web 站点

如果您希望更新 32 位的 MapXtreme Web 站点以使用最新的程序集，请遵循本节中的说明。

要运行使用旧版本 MapXtreme 创建的 Web 站点，必须编辑应用程序的 Web.config 文件，以指向新版本的程序集。



程序集位于以下位置处：C:\Windows\Microsoft.NET\assembly\GAC_32。

至少，您必须包含 MapInfo.CoreEngine.dll、MapInfo.CoreEngine.Wrapper.dll、
MapInfo.CoreTypes.dll 和 MapInfo.WebControls.dll。

完成编辑后，保存 Web.config 文件并重新生成 Web 站点。

检查 Web 控件的 Copy Local 属性设置为 False。请参阅第 44 页中的 [将 Copy Local 属性设置为 False](#)。

更新现有的桌面应用程序

可以重新编译使用旧版本 MapXtreme 创建的桌面应用程序使之能够在当前版本中使用。控件可能需要重新增加到窗体中。

但必须先完成几项工作。

- 将 Copy Local 属性设置为 false
- 将新程序集添加到项目
- 将程序集重定向到新程序集

将 Copy Local 属性设置为 False

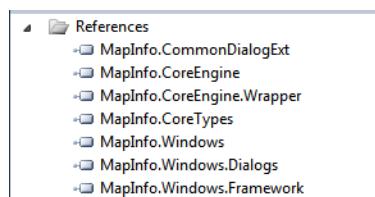
确定控件的 Copy Local 属性已设置为 False。已知的问题是：当您将 MapXtreme 桌面控件拖放到 Windows 窗体时，将会加载 Copy Local 属性已设置为 True 的引用。MapXtreme 引用必须指向全局程序集缓存 (GAC) 中存在的程序集，而不是本地容器路径（当 Copy Local 设置为 True 时发生）。

将 MapXtreme Web 控件拖放到窗体时，或者构建的控制台应用程序不基于 MapXtreme 模板时，ASP.NET Web 应用程序也存在相同的状况。此处也可以采用同样的解决方法：将 Copy Local 属性设置为 FALSE。

使用 MapXtreme Web 应用程序模板或基于 MapXtreme Web 站点模板的示例应用程序时，这种情况不会发生。

将新程序集添加到项目

下图显示了用于 MapXtreme desktop 应用程序的程序集。



将 MapXtreme 程序集重定向至新版本

要使用当前版本的程序集，必须重定向应用程序。Microsoft 提供几种程序集重定向机制。有关详细信息，请参阅 Microsoft 的 [.NET Framework 开发人员中心](#)。

应用程序配置文件

推荐用户使用应用程序配置文件来实现程序集的重定向。配置文件必须与应用程序位于同一目录并以应用程序命名。例如，必须为 *myApp.exe* 指定配置文件 *myApp.exe.config*。

应用程序配置文件将重载发布程序策略文件中的设置。

要重定向程序集，必须标注版本号和当前版本的公钥标记并将其添加到应用程序配置文件中。版本号的格式为 X.x.x.x，例如 7.3.0.140。

MapXtreme 程序集位于以下位置处：C:\Windows\Microsoft.NET\assembly\GAC_32。

发布程序策略文件

包含重定向设置的发布程序策略文件可与程序集一起安装在 GAC 中。不过不支持 Pitney Bowes Software 配置。

计算机配置文件

在计算机配置文件中指定重定向设置会导致所有引用程序集的应用程序使用升级版本。由于计算机配置文件将覆盖应用程序配置文件和发布程序策略文件的设置，因此应小心使用这个重定向方法。

在 Visual Studio 中创建应用程序

使用 **MapXtreme**，很容易就可以将地图添加到应用程序。提供的 Visual Basic.NET 和 Visual C# 项目模板允许创建简单的地图绘制应用程序而无需编写任何代码。

此外，还提供了可供研究的示例应用程序和 web 应用程序，用于实验并适应自身情况。有关分步说明如何使用这些示例应用程序的教程，请参阅[附录 A：如何创建和部署 MapXtreme 应用程序](#)。

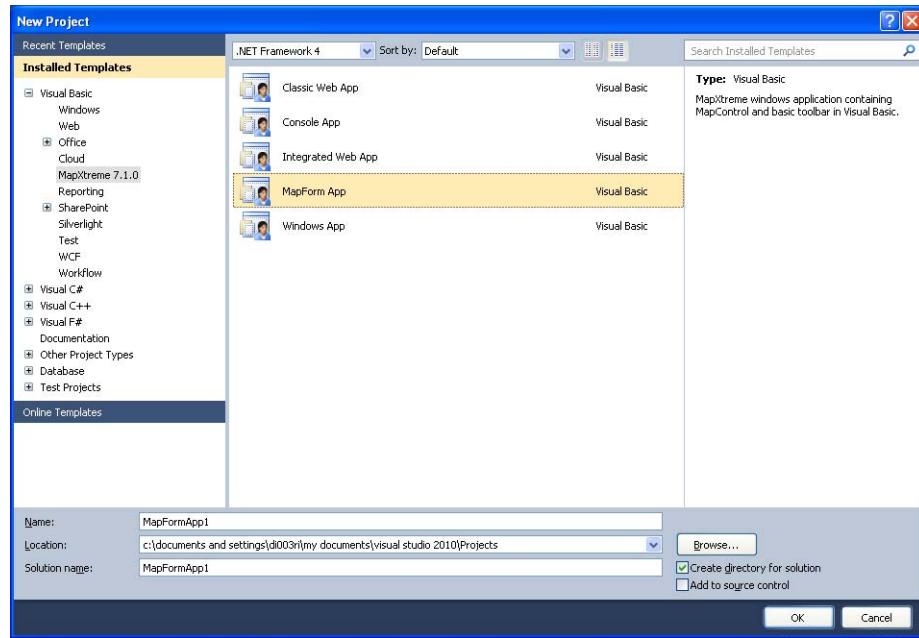
MapXtreme 还附带了一组教程应用程序，帮助您了解如何在 Web 应用程序中加入有用的地图绘制功能。每个教程应用程序都附有文档说明该应用程序是如何生成的。使用“开始”>“程序”菜单中的“学习资源”查看文档。教程应用程序均包含在名为 **MapXtremeTutorials.sln** 的 Visual Studio 解决方案中，该解决方案位于 **MapXtreme** 安装目录的 **\Tutorials** 文件夹中（默认位置是 **c:\Program Files\MapInfo\MapXtreme\7.x.x\Tutorials**，其中 **7.x.x** 表示版本号）。

以下步骤概述了生成简单桌面地图绘制应用程序的步骤。有关创建 Web 应用程序的步骤，请参阅[ASP.NET Web 应用程序](#)。

地图应用程序

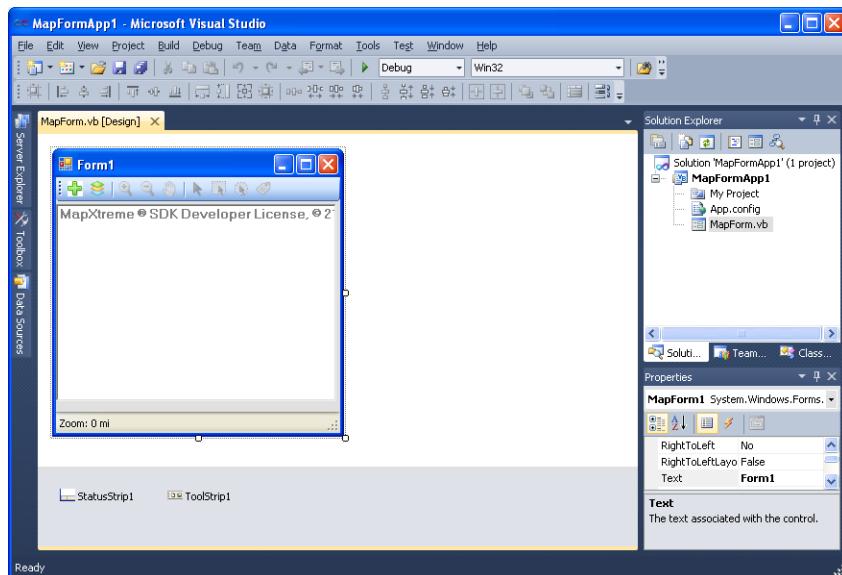
i 此示例为 Visual Basic.NET 专用。要创建 Visual C# 地图应用程序，请在以下步骤中使用 Visual C# 代替 Visual Basic。

1. 从 Visual Studio 的“文件”菜单中，选择新建项目。此时将显示“新建项目”对话框。
2. 在“新建项目”对话框的“已安装的模板”框架的 Visual Basic 文件夹下，选择 Windows。
3. 单击“新建项目”对话框的“模板”框架，选择 **MapXtreme 7.x.x MapForm** 应用程序。



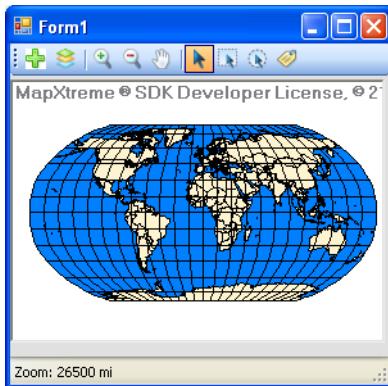
4. 选择合适的名称并单击确定。MapXtreme 创建应用程序。

在“解决方案资源管理器”下，双击**MapForm.vb**，将显示**MapForm.vb [设计]**。



5. 在“调试”菜单中，单击开始调试运行应用程序。

6. 单击打开表图标并加载数据。示例数据的默认位置为：
`Program Files\MapInfo\MapXtreme\7.x.x\Samples\Data`。



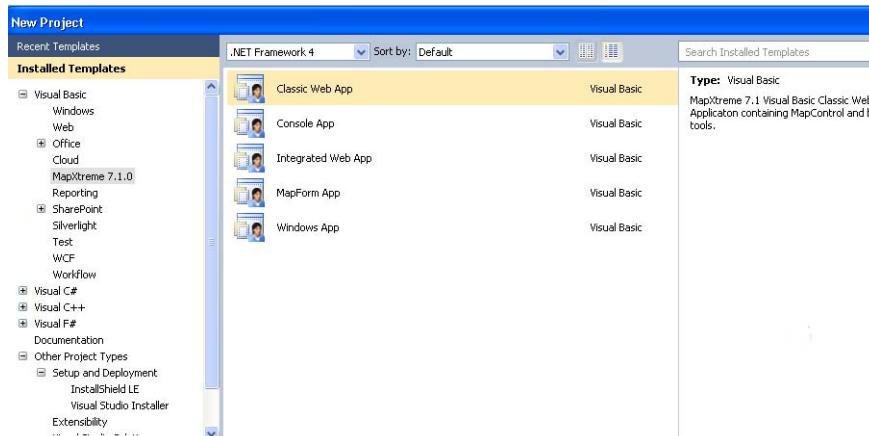
7. 可以像使用任何其它 Pitney Bowes Software 地图绘制应用程序一样，也可以使用工具栏的控件来操控地图。

ASP.NET Web 应用程序

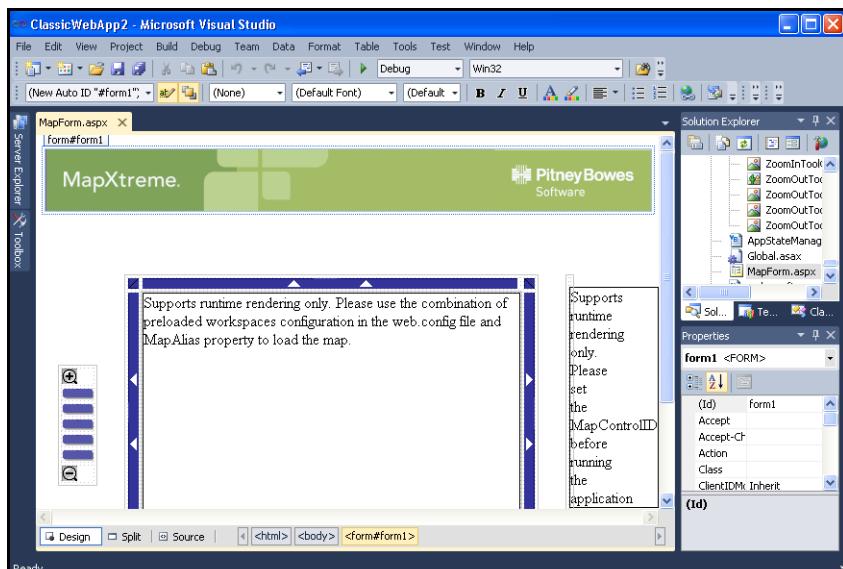
以下步骤概述了使用 MapXtreme Web 应用程序模板创建简单 ASP.NET Web 应用程序的步骤。IIS 7/8 预配置了此模板。

此示例为 Visual Basic.NET 专用。要创建 Visual C# ASP.NET 地图应用程序，请在以下步骤中使用 Visual C# 代替 Visual Basic。

1. 从 Visual Studio 的“文件”菜单中，单击新建项目。此时将显示“新建项目”对话框。
2. 从“已安装的模板”列表中，选择“Visual Basic”和“Windows”。从可用的 web 模板中，选择 MapXtreme 经典 Web 应用程序或 MapXtreme 集成 Web 应用程序。



3. 设置应用程序和解决方案的名称以及项目的位置。单击确定。
4. 在“解决方案资源管理器”下，双击 **MapForm.aspx** 以查看 MapControl 和一些工具的设计视图。



5. 生成项目。
6. 在 Visual Studio 的“调试”菜单中，单击开始调试运行应用程序。



7. 可以像使用任何其它 Pitney Bowes Software 地图绘制应用程序一样，也可以使用工具栏的控件来操控地图。

如果无法运行应用程序，请务必检查系统是否正在运行 ASP.NET 状态服务（控制面板 > 管理工具 > 服务 > ASP.NET 状态服务）。

MapXtreme 控件

使用其中一个模板创建基本应用程序后，就可以使用各种在 Toolbox 中提供的 MapXtreme 控件扩展。

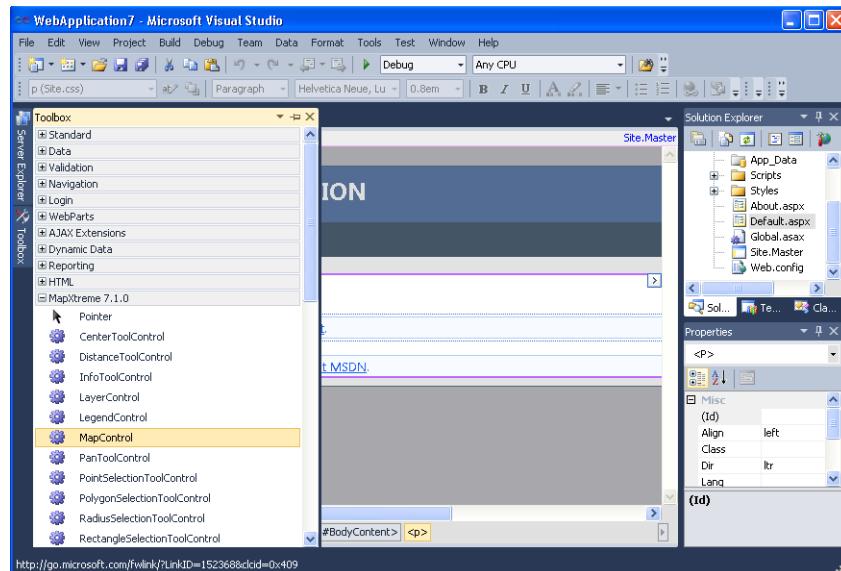
对于使用 Windows 窗体构建的桌面地图应用程序，可以使用 MapXtreme Windows 控件。任何 Visual Studio Toolbox 的 MapXtreme Windows 控件选项卡中的控件都可以添加到窗体。请参阅第 7 章：桌面应用程序、控件、对话框和工具。

对于 MapXtreme ASP.NET Web 应用程序，可以使用 MapXtreme Web 控件。任何 Toolbox 的 MapXtreme Web 控件选项卡中的控件都可以添加到窗体。有关详细信息，请参阅第 5 章：Web 应用程序、控件和工具。

构建 ASP.NET Web 应用程序但不使用模板

不使用 MapXtreme 模板也可以构建 ASP.NET 地图应用程序。例如，可以按以下步骤创建 Visual Basic Web 应用程序：

1. 从 Visual Studio 菜单中选择文件 > 新建。在 Visual Studio 的“已安装的模板”下，导航到您首选编程语言下的 Web 部分。选择通用 ASP.NET Web 应用程序模板，然后单击“确定”。
2. 从 Toolbox 中的 MapXtreme Web 控件组选择控件并将其拖到窗体。该操作会将 MapXtreme 程序集作为引用添加到项目中：MapInfo.WebControls、MapInfo.CoreEngine、MapInfo.CoreTypes 和 MapInfo.CoreEngine.Wrapper。还会使用程序集信息更新 web.config 文件。



3. 在这种情形下，您将注意到，设计器中 MapXtreme Web 控件和工具将显示红色的 X 而不是其图标。要正确显示这些图标，请从其中一个 MapXtreme 示例应用程序复制 MapXtremeWebResources 文件夹，然后将其粘贴到包含 web.config 和 default.aspx 文件的项目中。关闭然后重新打开网页查看这些图标。

4. 要在 IIS 7 经典模式下运行 web 应用程序，请将以下代码复制到 Web.config 文件中。要在集成的管线模式下运行，请跳转至步骤 6。

```
<system.web>
    <compilation debug="true" targetFramework="4.0">
        <assemblies>
            <add assembly="System.Design, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A" />
            <add assembly="MapInfo.CoreEngine, Version=7.3.0.140,
Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" />
            <add assembly="MapInfo.CoreEngine.Wrapper, Version=7.3.0.140,
Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" />
            <add assembly="MapInfo.CoreTypes, Version=7.3.0.140,
Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" />
            <add assembly="MapInfo.WebControls, Version=7.3.0.140,
Culture=neutral, PublicKeyToken=0a9556cc66c0af57" />
        </assemblies>
    </compilation>
    <sessionState mode="StateServer"
stateConnectionString="tcpip=127.0.0.1:42424"
sqlConnectionString="data source=127.0.0.1;userid=sa;password="
cookieless="false" timeout="20" />
    <httpHandlers>
        <add verb="*" path="MapController.ashx"
type="MapInfo.WebControls.MapController, MapInfo.WebControls,
Version=7.3.0.140, Culture=neutral, PublicKeyToken=0a9556cc66c0af57" />
    </httpHandlers>
    <httpModules>
        <add type="MapInfo.Engine.WebSessionActivator,
MapInfo.CoreEngine, Version=7.3.0.140, Culture=neutral,
PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator" />
    </httpModules>
</system.web>
```

5. 保存 Web.config。生成项目并运行应用程序。
6. 要在 IIS 7 或 IIS 8 集成管线模式下运行 web 应用程序，请在 <appSettings> 部分中添加以下粗体键值：

```
<appSettings>
    <!--Use this setting to set config sections for Classic or
Integrated Pipeline Mode--&gt;
    &lt;add key="MapInfo.Engine.Session.PipelineMode" value="Integrated" /&gt;
&lt;/appSettings&gt;</pre>
```

将以下代码复制到 Web.config 文件中 <appSettings> 部分的下方。

```
<system.webServer>
    <modules>
        <add name="WebSessionActivator"
type="MapInfo.Engine.WebSessionActivator, MapInfo.CoreEngine,
Version=7.3.0.140, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" />
    </modules>
    <handlers>
```

```
<add name="MapController" verb="*" path="MapController.ashx"
      type="MapInfo.WebControls.MapController, MapInfo.WebControls,
      Version=7.3.0.140, Culture=neutral, PublicKeyToken=0a9556cc66c0af57" />
    </handlers>
</system.webServer>
```

7. 保存 Web.config。生成项目并运行应用程序。

部署应用程序

在希望宿主应用程序的服务器计算机上安装 **MapXtreme** 组件有两个基本的策略：1) 使用随附的运行时安装程序，或者 2) 创建自己的安装程序并添加相应的合并模块 (MSM)。

MapXtreme SDK 包含 NCP 版本的运行时安装程序 **MXTRunNCP.exe**。SCP 版本的 SDK 包含 **MXTRunSCP.exe**。

用运行时安装程序部署

将运行时安装程序（**MXTRunNCP.exe** 或 **MXTRunSCP.exe**）用作定制安装过程的一部分。这会安装 **MapXtreme** 程序集并创建需要的注册项。它还包含必需的 .NET Framework v4.5.2。包含同时适合 Web 和桌面应用程序的运行时程序集和文件。此外，还必须将从 **Pitney Bowes Software** 或经销商处获得的运行时许可证文件安装到以下路径：**Program Files\Common Files\MapInfo\MapXtreme\7.x.x**。请参阅[用于部署的运行时许可证](#)。

对于基于 Web 的应用程序，使用运行时安装程序则更好。所有运行时组件都安装在它们的默认位置。没有用户配置的选项。

使用运行时安装程序的一个缺点是会安装不必要的文件（取决于不同的应用程序）。此外，如果要安装多个启用 **MapXtreme** 的应用程序，则需要在使用运行时安装程序时维护自己的引用计数。通过直接用自己的安装程序使用 MSM，引用可以自动维护。

使用运行时安装程序部署应用程序的步骤

要运行安装程序，您必须拥有该计算机的管理员权限。要安装到 Web 服务器上还需要 IIS 权限。

1. 从 **MapXtreme** 产品介质中选择安装。此时将显示“安装选项”页面。
2. 选择部署安装，然后选择安装运行时环境。安装向导显示。
3. 遵循提示继续安装。
4. 按照您的需要部署 Windows 或 Web 应用程序（例如，在 Visual Studio 中创建部署项目并添加应用程序）。
5. 在 Program Files 下的 \Common Files\MapInfo\MapXtreme\7.x.x 中或 Program Files(x86) 下安装运行时许可证文件，具体取决于您安装的是 64 位运行时还是 32 位运行时。

如果想从命令行执行运行时安装程序，则遵循以下指示。可执行文件位于产品介质光盘根目录下的 **\Install\InstallRuntime** 文件夹中。

- 要用最简练的 UI 执行运行时安装程序，请按照以下示例执行运行时安装程序：

```
MXTRunNCP.exe /v"/qb"
```

- 要自动执行运行时安装程序，请按照以下示例执行运行时安装程序：

```
MXTRunNCP.exe /v"/qn"
```

如果使用的 **MapXtreme** 是软件防复制保护的版本，则运行时可执行文件是 **MXTRunSCP.exe**。

用自己的安装程序进行部署

使用 Windows Installer 技术 (MSI) 创建自己的安装程序并包括 **MapXtreme** 合并模块。合并模块 (MSM 文件) 是一个独立的程序包，包含了安装组件必需的所有文件、资源、注册表项和安装逻辑。合并模块无法单独安装，它们必须置入 MSI 文件中。如果要精确调控安装的组件或想创建自己基于 MSI 的安装程序，则可以使用该策略。

使用 MSM 的一个缺点是：如果 Pitney Bowes Software 提供了产品更新，那么您就必须创建自己的补丁程序或更新的安装程序。

使用 MSM 的好处是可以控制要安装的组件且引用计数可以自动维护。特殊版本的程序集仅被复制到全局程序集缓存 (GAC) 中一次，且使用这些程序集为每个应用程序维护引用计数。如果随后删除一个应用程序，则引用计数（按 1 递减）将把程序集保留在 GAC 中。使用这些程序集的最后一个应用程序被删除后，程序集本身将被删除。

有多种开发人员工具可以帮助您创建 MSI 安装程序。例如 InstallShield Developer (Apresso Software Corporation) 和 Visual Studio (Microsoft)。请参阅 Microsoft **MSDN 库**中的“Windows 安装程序”主题。

MapXtreme 合并模块

以下是 MapXtreme 的合并模块。必要时将这些模块包括在安装程序 MSI 中。有关使用合并模块的详细信息，请参阅[打包桌面应用程序](#)。

注意：每个 MSM 文件名中的 7.x.x 均代表产品的当前版本号。MSM 位于 \Program Files\Common Files\Merge Modules 中。

合并模块的名称	包括的程序集	目的	何时需要
定制符号 MapInfoCustSymb_7.x.x.msm	无	包含定制符号	应用程序要使用常用位图符号时需要
桌面 MapInfoDesktop_7.x.x.msm	MapInfo.Windows MapInfo.Windows.Dialogs MapInfo.Windows.Framework MapInfo.CommonDialogExt	包含具有 .NET 控件的程序集（用于 C# 和 VB .NET 桌面应用程序）	应用程序使用“桌面”工具时需要
字体 MIFonts_7.x.x.msm	无	包含 MapInfo 字体，如 Symbols、Cartographic、Real Estate、Arrows、Miscellaneous、Oil&Gas、Transportation、Weather 和 Shields。	应用程序要使用常用 TrueType® 字体时需要

合并模块的名称	包括的程序集	目的	何时需要
地图化 MapInfoCoreEngine_7.x.x.msm MapinfoMXTConfig_7.x.x.msm MapInfoCoreEngineIntl.msm * MapInfoCoreResJPN_7.x.x.msm † MapInfoCoreResCHN_7.x.x.msm ‡	MapInfo.CoreEngine MapInfo.CoreEngine.Wrapper MapInfo.CoreTypes MapInfo.WMS.Client MapInfo.Windows.Printing MapInfo.Ellis.ExtensibleDataProvider MapInfo.LinearReferencing MapInfo.Ogc MapInfo.WorkspaceManager.Extension	提供核心地图绘制功能。它们也为这些程序集安装常用配置和默认引用文件。	必需
Web 控件 (Web Controls) MapInfoWeb_7.x.x.msm	MapInfo.Web MapInfo.WebControls	安装 .NET 程序集 MapInfo.Web，大量“Web 资源”将用在基于 Web 的应用程序中。 它还启动 ASPNetState 服务，并为“web 资源”创建虚拟目录（如果已安装 IIS）。	任何应用程序使用常用 web 控件时需要
Web 服务客户端 MapInfoServices_7.x.x.msm	MapInfo.Services	包含用于地理编码和路径规划的程序集	应用程序需要地理编码和路径规划时需要

合并模块的名称	包括的程序集	目的	何时需要
WFS MapInfoWFS_7.x.x.msm	MapInfo.WFS.Server	包含 WFS 所需的程序集。	应用程序使用 Web 图元服务用于数据变换时需要。
WMS MapInfoWMS_7.x.x.msm	MapInfo.WMS.Server	包含 WMS 所需的程序集。	应用程序使用 Web 地图服务用于检索数字图像时需要。

* 使用 MapXtreme SCP 构建部署应用程序时包括。

† 使用 MapXtreme JPN 构建部署应用程序时包括。

‡ 使用 MapXtreme CHN 构建部署应用程序时包括。

用自己的安装程序部署应用程序的步骤

以下是使用自己的安装程序安装 MapXtreme 时必须执行的任务：

1. 包括所需的 MSM。

默认情况下，Visual Studio 和 InstallShield 在 C:\ProgramFiles\Common Files\Merge Modules 中查找要包括的合并模块。这样就可以快速生成安装程序，不必重新配置开发环境。

2. 包括运行时许可证文件。

部署 web 应用程序

如果要为 web 应用程序创建自己的部署，则需要手动添加 MSM。Web 安装项目不检测 Web.config 文件中引用的程序集；因此，它不会将程序集（及其对应的 MSM）检测为从属项。要手动添加 MSM，请右键单击 Web 安装项目并选择“添加”>“合并模块”。选择 MapInfoCoreEngine_7.x.x、MapInfoMXTConfig_7.x.x、MapInfoWeb_7.x.x 以及任何其它所需的 MSM。

部署访问数据的应用程序

任何包括数据访问的 Visual Basic 或 Visual C# 应用程序都对 Microsoft Data Access Components (MDAC) 2.7 版或更新版本具有依赖关系。安装应用程序之前 MDAC 必须安装在目标计算机上，否则应用程序会失败。

在代理服务器背后部署 MapXtreme Web 应用程序

如果您在代理服务器背后部署您的 Web 应用程序，请确定将 MapXtreme 服务器 URL 添加到代理服务器的绕过缓存列表。这将允许 MapXtreme 为每个请求交付动态地图，这是根据设计意图进行的。

代理服务器依赖于显示的缓存图像。但是，MapXtreme 的图像是针对每条请求构建的，因此，在 Web 应用程序位于代理服务器背后时，将不会向缓存中发送任何图像。Web 应用程序显示一个红色的 X 替代地图图像。

已部署 Web 应用程序需要访问临时目录的权限

对于部署的 web 应用程序，如果使用集成安全系统，则请确保访问该站点的所有用户都具有访问临时目录及任何其它必需资源的权限。MapXtreme 在 ASP.NET 处理空间中执行，该处理执行从 IIS 进程传入的安全标记。必须将权限授予任何可能登录并访问该目录的用户。如果使用匿名访问，则必须将访问需要访问临时目录的权限授予 `IUSR_LocalMachineName` 系统帐户。MapXtreme 会从当前的 TEMP 环境设置中获得临时目录。

应用程序数据文件

应用程序数据文件是应用程序使用的不可执行的文件。MapXtreme 安装和使用以下应用程序数据文件。有关定制这些文件位置的详细信息请参阅[应用程序数据文件的位置](#)：

文件类型	文件名
缩写文件	MAPINFO.W.ABB
画笔文件	MAPINFO.W.PEN
投影文件	MapInfoCoordinateSystemSet.xml
矢量符号文件	MapInfo.fnt
定制符号目录	CustSymb
Nadcon 文件	*.las、*.los
jgd2000 文件	jgd2000.*

默认情况下，MapXtreme 应用程序在以下目录寻找数据文件：

- Program Files\Common Files\MapInfo\MapXtreme\7.x.x — 这是 MapXtreme 安装程序存放这些文件的目录。
- 用户的应用程序存放的目录。对于 Windows 应用程序，这是存放 .exe 文件的目录。对于 web 应用程序，这是存放 Web.config 文件的目录。

部署安装故障诊断

部署安装如出现故障，诊断时请考虑以下问题：

- 部署应用程序时使用了什么（例如：运行时、SDK、Setup.exe、NCP/SCP？）
- 许可证文件在 C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x 目录下吗？
- 计算机上是否有其他 .lic 文件？如果有，请将其删除。
- 部署计算机上的客户是否使用了管理员的身份登录？
- 部署计算机上是否安装了 4.5.2 Framework？
- 部署计算机上的 CoreEngine 是否注册为 COM+ 应用程序？
- 计算机上是否安装了 Visual C++ 10.0 CRT（运行时）？

地图绘制概念

创建地图绘制应用程序之前，理解基本的地图绘制概念及其如何在 **MapXtreme** 中实现这些概念会对您有所帮助。本章讨论了学习 **MapXtreme** 时将遇到的常用概念。

在本《开发人员指南》的最后，我们提供了[附录 L: 术语表](#)，其中包含了有关地图绘制和编程的术语，相信会对您有所帮助。

在本章中：

- ◆ 地图绘制和 **MapXtreme** 60
- ◆ 使用 **MapXtreme** 进行地理编码 64
- ◆ 使用 **MapXtreme** 进行路径规划 64

地图绘制和 MapXtreme

地图绘制应用程序的核心元素是地图。本章概述了使用 **MapXtreme** 构建应用程序时可能遇到的最重要的地图绘制术语。此外还介绍了 **MapXtreme** 对象模型中对应的命名空间，方便您快速获得需要的技术信息。这些主题包括：

- ◆ 地图
- ◆ 表
- ◆ **Layers**
- ◆ **Features**
- ◆ 标注和图例
- ◆ **Themes**
- ◆ 工具
- ◆ 工作空间
- ◆ 坐标系和投影

地图

地图显示地图图元之间的空间关系，例如城市边界、客户位置或输电线路。地图可以让您从视觉上确定图元的方向及它们所表示的内容。除了图元外，地图上的元素可包括标注、标题、图例和主题。主题是根据某项涉及图元的操作和地图上的信息创建的。

地图包含在 **MapControl** 中。**MapControl** 还提供了查看地图的基本工具（平移、放大、缩小、居中）。

可通过多种方法来创建地图：

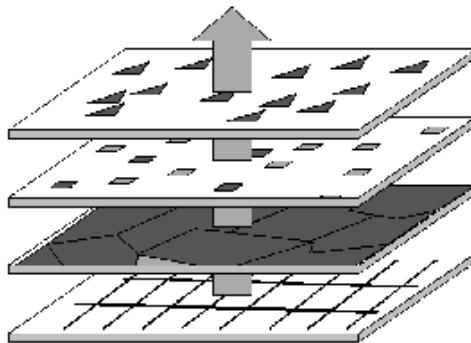
- 使用 **MapXtreme Workspace** 管理器来构建和保存地图工作空间。（请参阅 **Workspace** 管理器的特性）。
- 请使用 **MapXtreme** 模板，该模板提供了可拖放到 **Visual Studio** 窗体上的 **MapControl**（对于桌面应用程序，请参阅第 7 章：桌面应用程序、控件、对话框和工具，对于 web 应用程序，请参阅第 5 章：Web 应用程序、控件和工具）。
- 使用 **MapXtreme** 对象模型通过编程方式在应用程序中构建地图绘制（请参阅 **MapFactory** 和开发人员参考（联机帮助）中的 **MapInfo.Mapping** 命名空间）。

表

表包含要显示在地图上的数据。表保存描述图元的信息的行和列，包括它们的几何体、样式和属性。**MapXtreme** 支持各种数据源中的表，包括本机表 (**MapInfo .TAB**)、关系数据管理系统 (RDBMS)、**dBase**、**MS Access**、**ASCII** 和 **ESRI ShapeFile**。特性表包括光栅、网格、无缝、视图、WMS 和 **ADO.NET**。可通过 **TableInfo** 类查看表的类型。可以通过 **MapInfo.Data** 命名空间中的 **Catalog** 打开或关闭表。请参阅第 8 章：使用数据。

Layers

地图由图层组成。图层包含地图图元，例如，邮政编码边界、学校或街道网络。了解图层的顺序，这点非常重要。最底部的图层是第一个绘制的，而最顶部的图层是最后绘制的。应将其图元会遮蔽其它图层图元的图层分别放置到较低的位置。例如，边界区域图层应放置到点图层下面。



MapXtreme 中的图层可以表示除图元外的内容。图层可以是光栅或网格图像、无缝地图（连接的地图），包含标签或用户绘制的图元，还可以包含对象主题，例如饼图主题。可以通过简单的定位将图层分组，并简化其图元的动画。主要接口是 **IMapLayer**。有关详细信息请参阅 [Layers](#)。

Features

图元由它们的几何体、样式、数据源、键和属性描述。通常，图元为表中的行。支持的几何体包括包含给定区域的闭合对象（**Polygons**、**MultiPolygons**、**Rings**、**Rectangle**、**RoundedRectangles** 和 **Ellipses**）；表示数据某一位置的点对象（**Points**、**MultiPoints**）；以及包含给定距离的直线对象（**Curves**、**MultiCurves** 和 **LegacyArcs**）。

计算机化地图其中一项主要用途就是获取关于图元的信息。在 MapXtreme 中，可通过多种方法将图元返回到 **FeatureCollections**，例如，使用构架从空白创建，使用选择集工具或方法选择，或通过搜索这些满足特定标准集合的工具或方法的 **Catalog** 选择。

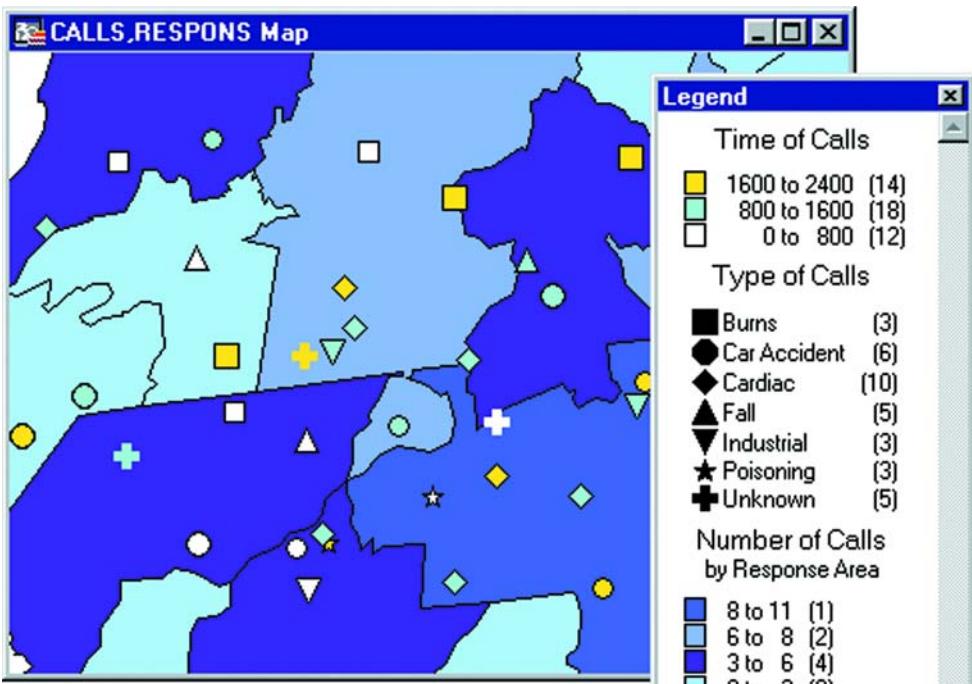
Feature 类位于 **MapInfo.Data** 命名空间。

标注和图例

如果地图中没有描述显示内容的元素，则这种地图用处不大。地图需要标注和图例等文本。上述提及的标注属于名为 **LabelLayer** 的图层类型。这样，您可以控制标注的各个属性，例如可视性、位置、样式和内容。用于处理标注的 MapXtreme 类包括 **LabelSource**、**LabelProperties** 和 **LabelModifiers**。请参阅 [MapInfo.Mapping 命名空间简介](#)。

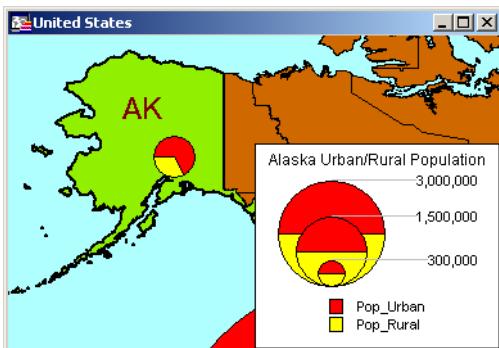
其它文本元素也可以用在地图中帮助传递正确的消息。图例是以编码方式描述图元的制图元素。例如，图例可以描述边界（例如学校地区）、直线（例如电力线路网络）或点（公司办事处位置）。图例也包含标题，用于描述地图表示的全部内容。

MapXtreme 中，图例连同地图标题和比例栏都是 Adornments 类的一部分。Adornments 位于 MapInfo.Mapping 命名空间中。



Themes

计算机上的电子地图不仅可以直观地显示地图图元之间的空间关系，而且还可以分析与图元关联的基础数据，从而了解到与看到的内容有关的详细信息。通常的分析技术是创建基于图元图层的主题，该图元图层的数据以特定的方式排列。例如，范围主题显示的颜色区中，每个颜色表示了地图上满足相同标准的图元。分级符号主题可用于表示人口的分布，例如最大的符号表示最多的人口。



此外也可以为标注创建主题。例如，使用范围标注主题表示城市之间的相对人口数量。最大的标注表示有最多人口的城市。

`MapInfo.Mapping.Thematics` 命名空间包含的类以 `Feature` 图层上的样式重载和 `Object` 主题实现来主题。对象主题添加新的图层时修饰符主题更改样式。所有主题都实现 `ITheme` 接口。

工具

多数地图绘制应用程序提供工具栏按钮（工具）的分类，用于一般绘制任务（例如在地图上绘制直线）和导航任务（例如放大）。MapXtreme 提供大量地图绘制工具，此外，还可以创建自己的自定义工具。

这些工具分为桌面工具和 web 工具，每个工具的 API 都包含在各自的命名空间中（`MapInfo.Tools` 用于桌面工具，`MapInfo.WebControls` 用于 web 工具）。

有关 MapXtreme 中桌面工具的详细信息请参阅 [MapXtreme 桌面工具 API](#)。有关 web 工具的详细信息请参阅 [第 5 章：Web 应用程序、控件和工具](#)。

工作空间

尽管并非严格的地图绘制概念，此处还是包括工作空间，因为它们使所有地图元素的处理更简单。MapXtreme 支持基于 XML 的工作空间格式，扩展名为 `.MWS`。其中全是地图的设置。工作空间的格式在[附录 C：理解 MapInfo 工作空间](#) 中解释。MapXtreme 提供名为 `Workspace` 管理器的实用程序来帮助您构造工作空间并保存工作空间以备后用。请参阅[第 24 章：Workspace 管理器](#)。

坐标系和投影

坐标系和投影是两个重要的地图绘制概念，应该对它们有基本的理解。投影是指如何将地图显示在图纸地图或计算机屏幕等平面上，而坐标系说明的是如何将地图图元在空间上安排。开发应用程序，尤其是开发空间精度和准确度很重要的应用程序时，坐标系和投影都需要重点考虑。

将球面的对象显示在平面上时，投影是减少失真发生的一种方法。主要需要权衡两个方面：保持图元的面积相等和保持图元的真实形状。有多种不同类型的投影，每种投影都可用于减少给定区域上的失真量。某些投影保持形状，而其它投影保持面积、距离或方向的精确度。

坐标系是一组参数，告知如何解释对象的位置座标。这些参数之一是投影。坐标可以有两种类型：球面坐标或笛卡尔坐标。球面坐标描述在 `Earth` 曲面上的位置，而笛卡儿坐标以二维空间来描述平面位置。二者均用 `x` 和 `y` 坐标表示。计算图元的距离或面积时就会出现差异，表示街道或河流等真实 `Earth` 位置时用球面坐标，而表示大脑解剖或棋盘等相对位置时用笛卡儿坐标。

开发应用程序时需要重点考虑地图要使用的坐标系。包含距离和面积计算的分析操作（例如缓冲、路径规划和查询）使用坐标系和投影来得出正确的结果。

坐标系和投影类是 `MapInfo.Geometry` 命名空间的一部分。有关详细信息请参阅[第 16 章：空间对象和坐标系](#)。

使用 **MapXtreme** 进行地理编码

以上讨论的地图都使用提供附加信息的数据，该附加信息不显示在地图上。例如，商店位置的表不仅包括将商店放置在正确地图位置的地理坐标，还可以包含与位置有关的数据，例如商店营业时间、客户服务电话号码和经理姓名。这让应用程序可以分析和得出那些会丢失在表的行和列中的信息。

通常，定制数据的表与参考图层一起包括在地图上，例如街道、城镇边界和河流等表示真实区域环境的图元。通常购买这些参考图层后就可以显示在地图上。**Pitney Bowes Software** 销售大量世界各地的参考数据。此外，**MapXtreme** 提供了超过 400 MB 的与世界位置有关的示例数据。要进行安装，请在**MapXtreme** 产品光盘浏览器上选择“安装样本数据”。使用“自定义”安装选项安装大量所需的数据集。

不过像您的商店位置或呼叫中心区域等定制数据还不能显示在地图上。表必须包含地理坐标，这样地图绘制引擎才能知道在何处绘制对象。将坐标分配到数据的过程称为地理编码。任何包含地理信息（例如地址或邮政编码）的数据表都可以地理编码。在包含相同位置的已地理编码的表上，该过程包括匹配定制的表。如果地址匹配完成，则地理编码表的坐标将分配到定制的数据。此外，定制的数据可以在地图上查看。

通常，地理编码是创建地图过程中的前期步骤。作为地图绘制应用程序的开发人员，您需要考虑要在地图上显示的数据类型及其地理编码的需要。

MapXtreme 框架提供使用地理编码客户端的类，用于访问 **Pitney Bowes Software** 的服务器地理编码产品。有关地理编码的详细信息请参阅 [第 19 章：地理编码](#) 和联机帮助中的 **MapInfo.Geocoding** 命名空间（可通过 Visual Studio 访问）。

使用 **MapXtreme** 进行路径规划

MapXtreme 开发人员可以使用的另一个组件是路径规划。驾驶导航型应用程序和那些与运输或布置电缆的规划路径有关的应用程序均需要使用路径规划。通常的目标是用最短距离或最短旅行时间来定位路径。

MapXtreme 提供四种类型的路径规划：点到点路径规划、多点路径规划、矩阵路径规划和等值线路规划。每种类型都提供了大量选项，用于按需创建适当的路径规划网络。

像地理编码所提供的功能一样，**MapXtreme** 允许开发人员在他们的应用程序中使用预先构建的路径规划客户端，这些应用程序可以与 **Pitney Bowes Software** 的路径规划服务器产品交互。请参阅 [第 20 章：路径规划](#) 和 **MapXtreme** 联机帮助中的 **MapInfo.Routing** 命名空间。

4

理解 MapXtreme 体系结构

本章着重介绍 MapXtreme 体系结构的设计，方便您根据开发需求做出精明的选择。理解产品的体系结构有助于创建出有效使用产品特性和功能的应用程序。

在本章中：

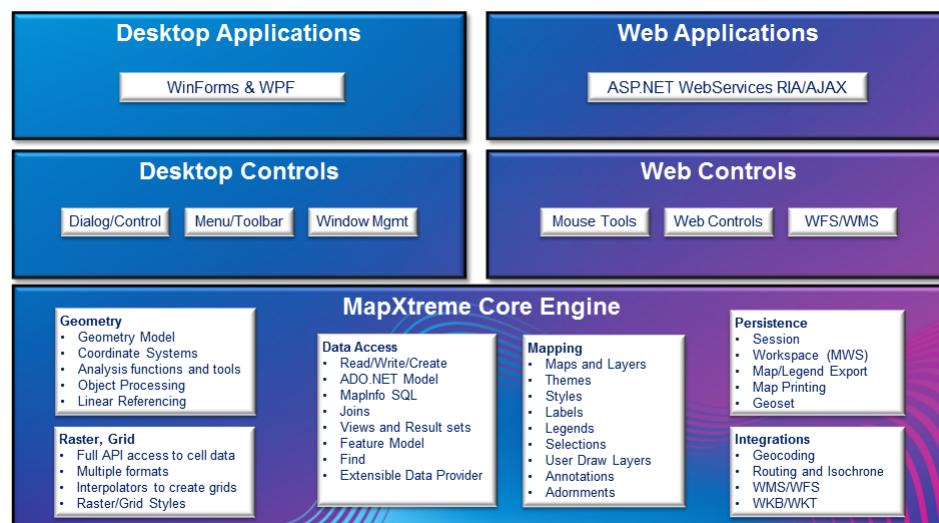
◆ MapXtreme 体系结构	66
◆ 对象模型概述	67
◆ 应用程序体系结构	70
◆ Web 应用程序体系结构	71
◆ 桌面应用程序体系结构	73

MapXtreme 体系结构

MapXtreme 的构建基础是 Microsoft .NET 框架，使用了 Microsoft .NET 框架包括的功能。Pitney Bowes Software 的这种创新让您可以在 .NET 框架上开发应用程序。我们还努力结合本公司的 Windows 产品的功能和易用性，使这些产品成为一套综合性的对象模型。要为可以预测到的未来技术开发基于 Windows 的产品，那么对象模型就是 Pitney Bowes Software 合作伙伴、客户以及 Pitney Bowes Software 自身的基础。

使用类似的代码，您可以开发既能在桌面又能在 Web 上部署的应用程序。如果开发桌面应用程序，则可以调整应用程序，只需要更改很少的代码即可用于后续的 web 部署。

下图说明了 MapXtreme 体系结构。MapInfo.CoreEngine.dll 程序集和 MapInfo.CoreTypes.dll 程序集包含了大多数核心地图绘制和数据访问功能。Core Engine 的上面是 MapInfo.Windows 和 MapInfo.Web 命名空间，这些命名空间包含了控件、工具和其它特定于每种部署环境的功能。任何从 MapXtreme 对象模型开发的应用程序都在 MapInfo.Windows 或 MapInfo.Web 命名空间之上构建。



MapXtreme 体系结构

对象模型概述

MapXtreme 对象模型由大量命名空间组成。.NET 命名空间是分类系统，用于区分具有相同名称的其它对象和特定的类、方法和属性。通过使用命名空间，.NET 开发人员可以避免对象名称与其方法和属性名称发生冲突。

MapXtreme 包含主要接口和类的彩色框图，并通过命名空间分隔显示它们如何彼此相关。可以通过“学习资源”页面查看 PDF 版本的框图，该页面可通过 Windows “开始”菜单访问。具体而言，可以从以下路径访问“学习资源”：开始 > 所有程序 > MapInfo > MapXtreme > 学习资源。

下表包含了在 MapXtreme 对象中实现的多个命名空间。每个命名空间的概述包括在后续章节中。本手册包含了与命名空间及其使用有关的详细信息，以下每个概述说明都包含了对本手册某些部分的引用。

- **MapInfo.Data** 命名空间
- **MapInfo.Data.Find** 命名空间
- **MapInfo.Engine** 命名空间
- **MapInfo.Geometry** 命名空间
- **MapInfo.Mapping** 命名空间
- **MapInfo.Persistence** 命名空间
- **MapInfo.Raster** 命名空间
- **MapInfo.Styles** 命名空间
- **MapInfo.WebControls** 命名空间
- **MapInfo.Tools** 命名空间
- **MapInfo.Geocoding** 命名空间
- **MapInfo.Routing** 命名空间

在 MapXtreme 的“程序员参考”中，完整的对象模型由命名空间组织并集成在 Visual Studio 中。

如果已经使用了 MapX 或非 .NET 版本的 MapXtreme (MapXtreme for Windows v3)，请务必查看[附录 J：迁移到 MapXtreme](#) 比较两个产品的对象模型。

MapInfo.Data 命名空间

MapInfo.Data 命名空间包含了实现 MapInfo 数据提供方的类和接口。对象模型具有多个类用于访问数据。可以根据存储的数据格式而使用特定的类来访问数据。此外，我们现在实现了 ADO.NET，用来访问以其它类不包含的格式包含的任何数据。有关 MapInfo.Data 命名空间的详细信息，请参阅[第 8 章：使用数据](#)和[第 11 章：从 DBMS 访问数据](#)。

MapInfo.Data.Find 命名空间

MapInfo.Data.Find 命名空间包含了用于搜索数据的类。该命名空间通过指定可以绘制地图的表和执行搜索的列（必须带有索引）来简化对象的搜索。有关 MapInfo.Data.Find 命名空间的详细信息，请参阅[第 13 章：查找位置](#)。

MapInfo.Engine 命名空间

MapInfo.Engine 命名空间包含了所有直接与核心功能有关的类，该核心功能驱动基于 MapXtreme 的所有应用程序。其中包括核心 Session 类，该类是所有 MapXtreme 应用程序的起点。有关 MapInfo.Engine 命名空间的详细信息，请参阅[第 9 章：运用核心 MapXtreme 类](#)。

MapInfo.Geometry 命名空间

MapInfo.Geometry 命名空间是一种可扩展的层次结构，基于 OGC (Open GIS Consortium) 标准、坐标系可互操作性和对象处理。MapInfo.Geometry 命名空间包含了用于创建和编辑 Geometry 对象的类、接口和枚举。有关 MapInfo.Geometry 命名空间的详细信息，请参阅[第 16 章：空间对象和坐标系](#)。

MapInfo.Mapping 命名空间

MapInfo.Mapping 命名空间包含了用于创建、显示和导出地图、图层、修饰符和标注的类、接口和枚举。有关 MapInfo.Mapping 命名空间的详细信息，请参阅[第 12 章：将地图绘制功能增加到应用程序](#)。

MapInfo.Mapping.Legends 命名空间

MapInfo.Mapping.Legends 命名空间包含用于创建并显示 Cartographic 和 Thematic Legends 的类、接口和枚举。有关详细信息请参阅[图例](#)和[使用主题和图例](#)。

MapInfo.Mapping.Thematics 命名空间

MapInfo.Mapping.Thematics 命名空间包含了作为图层样式和图层本身实现主题的类。对象主题添加新的图层时，主题可应用到 Modifier 主题来更改样式。所有主题都实现 ITheme 接口。有关 MapInfo.Mapping.Thematics 命名空间的详细信息，请参阅[第 14 章：使用主题和图例](#)。

MapInfo.Persistence 命名空间

MapInfo.Persistence 命名空间包含了支持基于 XML 工作空间读写的类，用于启用对地图绘制工作空间的保存和检索。请参阅[附录 C：理解 MapInfo 工作空间](#)。

MapInfo.Raster 命名空间

MapInfo.Raster 命名空间公开了 Pitney Bowes Software C/C++ Raster 和 Grid API 的全部功能。可使用 MapInfo.Raster.RasterRead 打开光栅图像进行查询。可使用 MapInfo.Raster.GridRead 打开网格图像进行查询。可以使用 MapInfo.Raster.HillshadeWrite 将 Hillshading 增加到现有的网格。相关类包括 MapInfo.Raster.RasterInfo 和 MapInfo.Raster.GridInfo。有关 MapInfo.Raster 命名空间的详细信息，请参阅[第 17 章：运用光栅和网格](#)。

MapInfo.Styles 命名空间

MapInfo.Styles 命名空间强调了 Styles 对象模型。Style 类是所有样式的基类。有关 MapInfo.Styles 命名空间的详细信息，请参阅[第 15 章：为地图设置样式](#)。

MapInfo.WebControls 命名空间

MapInfo.WebControls 命名空间为使用 MapXtreme ASP.NET 应用程序的 Visual Studio 模板提供支持。此命名空间中具有 MapControl 和 LayerControl 设计阶段增强功能。有关 MapInfo.WebControls 命名空间的详细信息，请参阅[第 5 章：Web 应用程序、控件和工具](#)。

MapInfo.Windows 命名空间

MapInfo.Windows 命名空间包含了实现各种窗口控件及其必需组件的类，用于在 Windows 应用程序中开发窗体。Windows.Dialogs 命名空间包含了实现 Windows 应用程序中使用的各种对话框和对话框组件的类。有关 MapInfo.Windows 命名空间的详细信息，请参阅[第 7 章：桌面应用程序、控件、对话框和工具](#)。

MapInfo.Tools 命名空间

MapInfo.Tools 命名空间包含了用于创建和实现在桌面地图应用程序中使用的多种工具的类。有关 MapInfo.Tools 命名空间的详细信息，请参阅[MapInfo.Tools 命名空间概述](#)。

MapInfo.Geocoding 命名空间

MapInfo.Geocoding 命名空间包含了定义 MapXtreme 客户端用于地理编码的类、接口和枚举。支持使用 MapInfo 地理编码服务器或 MapInfo Location Utility 服务的地理编码（中文版中没有此功能）。要执行地理编码，则必须有用于运行地理编码服务器或 Location Utility 服务的 URL。地理编码服务器和 Location Utility 服务的接口是类似的，因为它们对于地理编码请求、约束、响应、结果代码、输入和候选地址都使用相同的类。请参阅[第 19 章：地理编码](#)。

MapInfo.Routing 命名空间

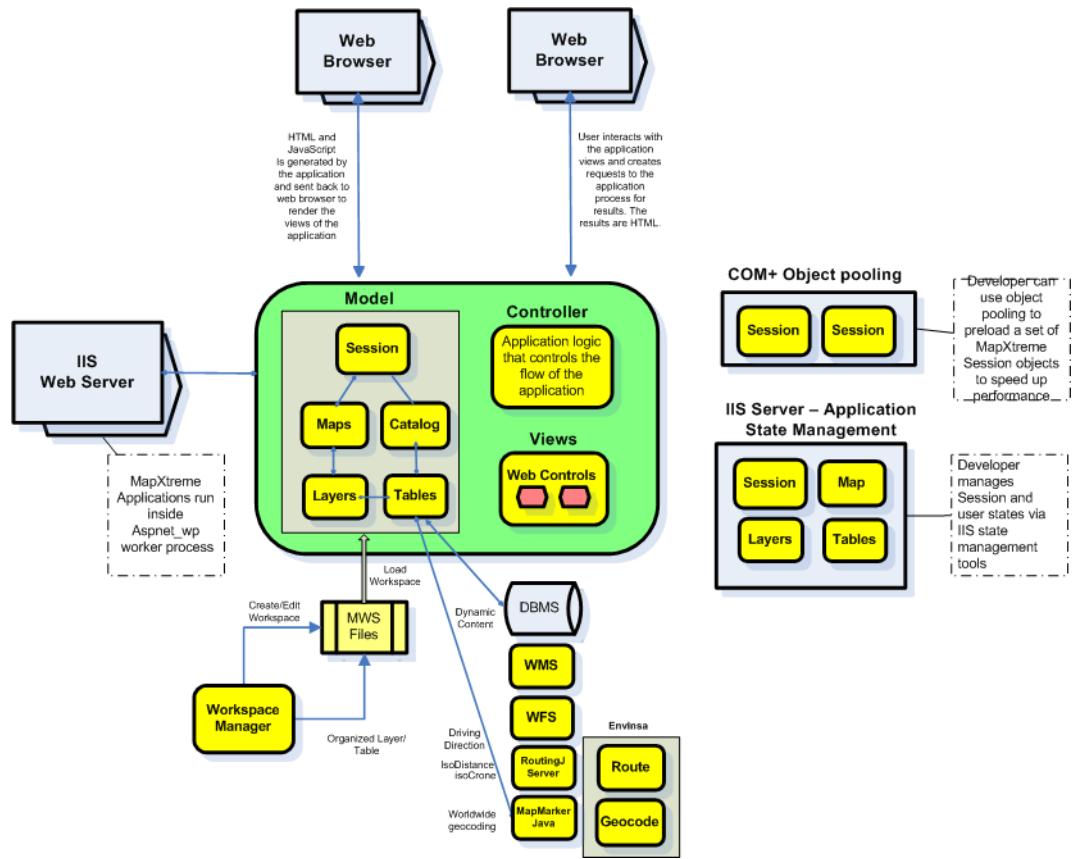
MapInfo.Routing 命名空间包含的类、接口和枚举用于让 .NET 客户端进行 Routing。MapInfo.Routing 命名空间包含的类支持点对点、多点、矩阵和等值线路径规划。为最短时间或最短距离而优化。它与其它 MapInfo 路径规划服务器产品进行交互。此外还具有避免特定点的功能。路由结果可以包括能在地图上显示的详细指示和（或）路由几何体。请参阅[第 20 章：路径规划](#)。

应用程序体系结构

在了解 MapXtreme 命名空间概况后，下一步就要考虑计划构建的应用程序的体系结构。

使用 MapXtreme，可以同时构建 web 应用程序和桌面应用程序。以下部分说明了 web 和桌面应用程序可能具有的体系结构。设计基于模型-视图-控制器范例，该范例将应用程序的数据模型、用户接口和控制逻辑分隔到三个独立的组件中。这样就可以在对其它组件产生最小影响的情况下修改某一组件。[第 5 章：Web 应用程序、控件和工具](#)和[第 7 章：桌面应用程序、控件、对话框和工具](#)提供了关于主要设计元素的重要附加信息以及在计划 MapXtreme 应用程序时要考虑的决策。

Web 应用程序体系结构



体系结构描述

上述 Web 应用程序考虑到以下组件和功能：

- MapXtreme Web 应用程序
- Microsoft .NET 基础结构
- 地图构建工具
- 数据访问

MapXtreme Web 应用程序

典型的 MapXtreme web 应用程序包含视图（表示层）、模型（与数据源和应用程序内部数据模型进行交互）和控制器（控制应用程序流程的业务逻辑）。

MapXtreme 提供了集成到 Microsoft Visual Studio 的 web 模板，帮助您创建初始的 web 应用程序。

构建 web 应用程序的相关教程，请参阅[生成 Web 应用程序](#)。

通过将 MapXtreme web 控件拖放到 Visual Studio web 窗体来构建视图。通过使用 MapInfo.Engine 和 MapInfo.Data 命名空间下的对象，构建内部数据结构并与外部数据源、基本地图和动态内容进行交互。使用控制器代码将视图和数据联系在一起并向用户提供活动顺序，以便有效地利用应用程序来解决业务需求或问题。

Microsoft .NET 基础结构

MapXtreme 在 Microsoft .NET 4.5.2 Framework 下运行。使用 MapXtreme 构建的应用程序以 ASP.NET 应用程序的形式在 IIS 的工作进程中运行。

Microsoft ASP.NET 框架为高性能企业应用程序的开发人员提供了 COM+ 对象池，以便能够预装如工作空间等对象。MapXtreme 的对象模型在该框架下能够非常有效地运行。该框架还提供了应用程序状态管理工具（如 StateServer 和 SQL Server）及自动和手动状态管理控件。[第 6 章：了解状态管理](#)提供了关于这些主题的重要信息。

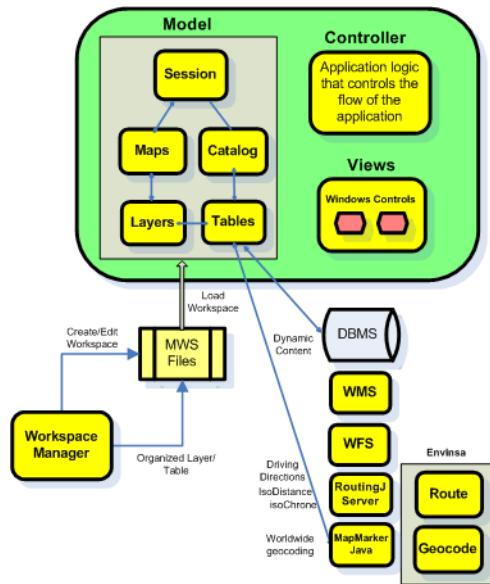
地图构建工具

使用 MapXtreme Workspace 管理器创建应用程序的基本地图。通过这些工具可以管理每个地图图层并控制其缩放级别、标注、样式、主题图和修饰，以提供对应用程序的精确表示。信息保存到基于 XML 的工作空间，以方便日后检索。请参阅[第 24 章：Workspace 管理器](#)。

数据访问

该体系结构的一个要素是访问动态数据内容的能力。MapInfo.Data 命名空间中的对象提供了这种能力。数据内容可来自大量数据源，如 WMS、WFS、远程数据库管理系统、GPS 中的实时种子或 MapInfo 路径规划服务中的驾驶导航。要充分利用各种数据，可以同时使用来自不同数据源的数据。请参阅[第 8 章：使用数据](#)。

桌面应用程序体系结构



桌面应用程序的体系结构与其模型-视图-控制器设计中 web 应用程序的体系结构类似。

独立的组件用于表示层、应用程序模型和业务逻辑。使用 **Workspace** 管理器构建所需的基本地图。使用 **Windows** 控件和对话框赋予应用程序丰富的用户体验。也可同时使用来自各种数据源的动态数据内容，控制应用程序的流程和逻辑。

有关详细信息，请参阅[第 7 章：桌面应用程序、控件、对话框和工具](#)。

Web 应用程序、控件和工具

本章汇总了大量关于使用 **MapXtreme** 提供的 Web 控件、工具和便于构建 ASP.NET Web 应用程序的信息。

在本章中：

◆ Web 应用程序请求 / 响应周期	75
◆ MapXtreme Web 应用程序组件	75
◆ MapXtreme Web 控件和工具	78
◆ Web 控件的体系结构	81
◆ 使用 MapXtreme Web 控件	84
◆ 管理 Web 控件	85
◆ 创建定制工具	86
◆ 使用和分发定制 Web 控件	88
◆ 将 InfoTool 添加到 Web 应用程序	89
◆ ASP.NET AJAX 和 MapXtreme Web 应用程序	90
◆ MapXtreme 瓦块句柄	92
◆ HTML/XHTML 验证问题	95
◆ 将 Post-back Web 控件 迁移到 JavaScript Web 控件	96
◆ Web 控件专题	97

Web 应用程序请求/响应周期

要规划和构建有效的 web 应用程序，需要切实地了解客户端（浏览器）与服务器（web 应用程序）之间的后台交互。简而言之，web 应用程序是可通过因特网或内联网用 web 浏览器进行访问的软件应用程序。用户可通过与 Web 页面上的元素进行交互（向 Web 服务器发送 HTTP 请求进行处理）的方法以 HTML 页面的形式访问应用程序的功能。Web 服务器然后向用户回复响应，以满足其请求。

MapXtreme web 应用程序通常可为用户提供地图图像及一些与地图交互的工具。用户只需点击“放大”工具显示不同的地图视图，即可轻易地完成一次请求/响应周期。放大请求将在后台被发送至服务器。服务器将对该请求进行处理，并以刷新后的地图图像（显示新视图）作出响应。

有关 MapXtreme web 应用程序体系结构的信息，请参阅[第 4 章：理解 MapXtreme 体系结构](#)。

有关构建 MapXtreme 应用程序的教程，请参阅[附录 A：如何创建和部署 MapXtreme 应用程序](#)。

有关创建 ASP.NET Web 应用程序的讨论，请参阅 MSDN 库中的[ASP.NET Web 应用程序项目](#)。

MapXtreme Web 应用程序组件

以下各节涵盖了典型 MapXtreme web 应用程序的主要组件，包括：

- [MapXtreme Session](#)
- 背景地图
- [MapControl](#)
- 地图工具

MapXtreme Session

MapXtreme Session 是所有 MapXtreme 应用程序的起点。其管理 MapXtreme 应用程序所需资源的初始化。通过 MapXtreme Session，还可访问诸如 Data.Catalog、MapFactory、CoordSysFactory 和 Selections 等其他重要对象。

要访问 MapXtreme Session，请调用 `MapInfo.Engine.Session.Current()` 方法。进程中的每个线程均有一个 MapXtreme Session 对象。每个线程只能有一个 MapXtreme Session 且该会话不可以在多个线程之间共享。请参阅[Session 接口](#)。

Web 应用程序可为每个用户提供一个 MapXtreme Session 对象，或以对象池的形式为多个用户提供多个 MapXtreme Session。有关两种开发模型的讨论，请参阅[第 6 章：了解状态管理](#)。

背景地图

图像形式的背景地图是 MapXtreme web 应用程序最为直观的组件。其为该应用程序用户提供了关于地图图元的信息，并显示了这些地图图元与其他地图图元之间的关系。背景地图通常由诸如管理边界、网络街道和兴趣点之类的参考图层组成。与该应用程序相关的自定义数据，例如，表示办公位

置的点、手机信号发射塔或 ATM 机，均是附加图层。参考图层和自定义图层并不根据用户与该应用程序的交互而发生变化。通常发生变化的是地图显示。用户可放大某特定位置，以便根据其在 web 请求中提交的标准创建覆盖地图的主题图影线。

背景地图是以其初始或“干净”状态展示给用户的。在设计池化应用程序时，该状态是一个重要考虑因素，因为应用程序必须处理用户状态的变化。

背景地图是从基于 XML 的工作空间 (.mws) 预载进应用程序的。工作空间是在应用程序的 web.config 文件中标识的。请参阅[初始地图看起来应该是什么样子？](#)。MapXtreme web 模板和示例应用程序提供的 web.config 文件包括样本数据的硬编码路径。如果您的 web 应用程序是基于任意这一种，请确保根据自己的数据调整路径。

MapXtreme 随附的 Workspace 管理器实用程序是有助于构建背景地图的桌面应用程序示例。请参阅[第 24 章：Workspace 管理器](#)。

MapControl

MapControl 是包含 Map 对象实例的 MapXtreme web 服务器控件。MapControl 是 MapXtreme 应用程序中的主要 web 控件，其显示背景地图并向地图上的工具交互作出响应。在后台，可以通过使用 MapControl 的 MapAlias 属性从 MapFactory 获得该 Map 对象。地图渲染后导出为图像，并以控件 标记的形式返回浏览器。

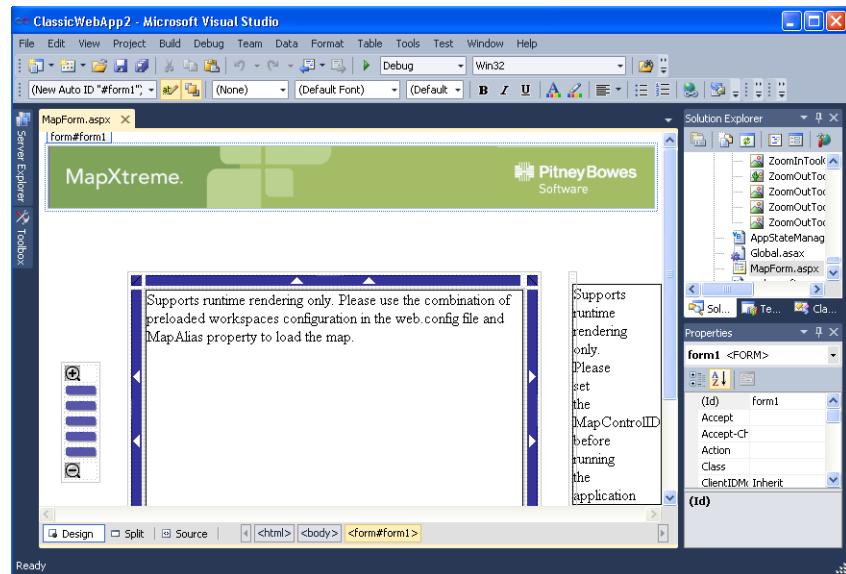
如同 MapXtreme Web 应用程序模板一样，MapXtreme 随附的示例应用程序带有内置 MapControl。但在设计模式中，所显示的地图并未被渲染。这是因为从 Web MapControl 获取地图时，服务器上肯定有 Web 应用程序在运行，为地图图像提供了一些动态生成的 javascript。它仅在运行时/调试时可用。



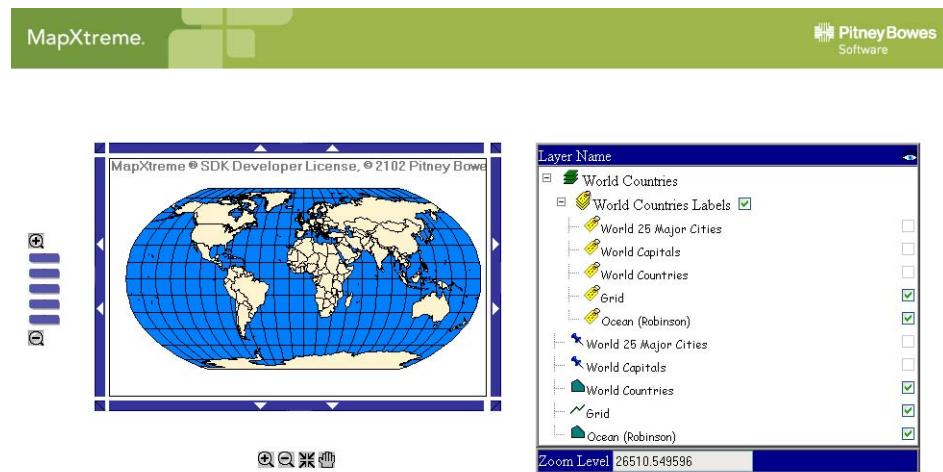
如果您将 Visual Studio Visual Basic 或 C# ASP.NET 模板作为起点，要将 MapXtreme 资源纳入项目中，则必须将 MapControl 和工具手动添加至 Web 窗体。

有关详细信息，请参阅[MapXtreme Web 控件和工具](#)。

图：设计阶段中的 MapXtreme Web 应用程序视图



图：运行时的 MapXtreme Web 应用程序视图



地图工具

MapXtreme 提供的多种地图工具有助于导航背景地图并与其进行交互。这些工具包含于 Visual Studio 工具箱中。将这些工具拖放至 web 窗体即可使用它们。

大多数这些工具嵌入在 MapXtreme Web 应用程序模板和示例应用程序中，包括：

- 居中、平移和缩放基本工具
- 缩放级别介于 500 至 12,500 地图单位之间的 ZoomBarTool
- 用于以 45 度增量进行直线平移的定向工具 (N, NE, E, SE, S, SW, W, NW)

LegendControl、Distance 工具和选择工具位于 Visual Studio 工具箱的 MapXtreme 部分。

有关工具的讨论，请参阅 [Web 控件和工具的描述](#)。

通用 WebTool 是所有 MapXtreme 地图工具的基本工具，也位于工具箱中。该工具可用于将定制行为添加至某工具中。请参阅[创建定制工具](#)。

状态管理和对象池功能

构建稳定、可扩展且可带给用户超凡体验的 MapXtreme web 应用程序是您的目标。MapXtreme 提供的配置有助于管理应用程序的状态，并使其随用户数量的添加而扩展。

经过配置，MapXtreme web 应用程序模板和示例应用程序可手动管理状态。这意味着，仅用户在与应用程序进行交互期间作出的更改才得以保存和恢复。编写应用程序及保存/恢复需求所特有的代码即可完成此操作。

为了更有效地为不断潜在增长的应用程序用户提供服务，可将模板和示例应用程序配置为使用 Microsoft COM+ 对象池服务，在该服务中 MapXtreme Session 对象是可用的并可在多个请求之间共享。

规划 web 应用程序时需要慎重考虑状态管理和对象池需求。请参阅[第 6 章：了解状态管理](#)。

MapXtreme Web 控件和工具

MapXtreme web 控件和工具内嵌于 Web 页面中。Web 控件通过 web 工具对交互作出响应。Web 页面在服务器上对通过工具捕获的信息进行处理，并通常将其作为新的地图图像返回至客户端。

MapXtreme 可提供三个 web 控件（MapControl、LayerControl 和 LegendControl）和多个地图工具。Web 控件可显示内容（例如，地图）、地图和标注图层及其属性列表或标识图层意义的图例。这些工具与 MapControl 进行交互，以更改地图视图并选择用于进一步分析的地图图元等。有关 web 控件和工具的描述，请参阅 [Web 控件和工具的描述](#)。

Web 控件和工具可从 Visual Studio 中多个位置获取：

- MapXtreme 标题下的 Visual Studio 工具箱。将这些控件拖放至 Web 窗体，以便将映射功能添加至项目中。

- **Web 应用程序模板：**从 MapXtreme Web 应用程序模板（Visual Studio 的“文件”>“新建项目”菜单）开始构建不含编码的地图原型。
- **示例应用程序：**各类任务的示例可在 \Samples\Visual Studio 20x\Web\Features 文件夹中找到。也可根据根据自身需求参考或扩展所提供的源代码。

Web 控件和工具的 API 在 **MapInfo.WebControls** 命名空间中。请参阅《开发人员参考》以了解更多信息。

支持 JavaScript 的部分页面更新

MapXtreme web 控件和工具组合使用了 web 页面和 JavaScript 以通知 MapXtreme 服务器要执行的操作。每个地图工具均指定了 **JavaScript ClientInteraction**（定义必须执行的操作，如单击鼠标、绘制矩形、线段或多边形等）和 **JavaScript ClientCommand**（向服务器发送 URL 请求以处理该命令，如平移、缩放或选择对象等）。

这些支持 JavaScript 的工具并不会在每次使用中触发全页回贴。每次使用工具后，通常只刷新地图图像。欲了解该开发模型，请参阅[地图工具体系结构](#)。有关事件处理、错误管理和状态管理的信息，请参阅[管理 Web 控件](#)。

每当使用某工具时，先前的 MapXtreme 控件（v6.5 之前的版本）均要求回贴调用 **Page_Load** 和 **Page_Unload**。这些工具适用于向后兼容，建议不要将其用于新的开发项目。请参阅[将 Post-back Web 控件迁移到 JavaScript Web 控件](#)。

可自定义

如果所提供的工具并不能提供能满足您的要求的功能，可考虑对其进行修改。此操作可能象更改内置工具属性一样容易，也可能象编写您自身的 JavaScript 和服务器端类以扩展其功能一样复杂。Web 控件和工具的源代码可在 **Samples\MapXtremeWebControlsSourceCode** 文件夹中找到。请参阅[创建定制工具](#)。

MapXtreme 提供的 ASP.NET AJAX 示例应用程序演示了如何使用 MapXtreme web 地图绘制应用程序中 Microsoft 的 ASP.NET AJAX 控件。请参阅[ASP.NET AJAX 和 MapXtreme Web 应用程序](#)。

Web 控件和工具的描述

以下是可用的 MapXtreme Web 控件和工具。

Web 控件 (Web Controls)	描述
MapControl	<p>可用于显示 Map 对象实例。每个 Map 对象均有一个 MapAlias，例如 Map1。</p> <p>在运行时，MapControl 可显示源自于 MapFactory 的地图，只需使用该地图的 MapAlias 属性即可。通过导出地图图像来绘制地图，并在 HTML 标记中引用该图像。如果 MapAlias 属性未指定或者无效，则从 MapFactory 中选择第一个地图。在设计阶段设置 MapControl MapAlias 属性。</p>
LayerControl	<p>可用于以树视图结构显示地图图元图层和标注图层。该控件可以打开或关闭特定图层的可视性并显示只读的当前缩放值。可视性更改仅在使用 LayerControl 的应用程序处于活动状态时永久有效。</p>
LegendControl	<p>LegendControl 可用于显示特定 MapControl 的图例。返回的图例是非交互式图像。默认导出格式是 GIF。</p> <p>在设计阶段，只需使用其 LegendAlias 或其在地图图例列表中的索引来指定要显示的图例。也可使用页面上的 JavaScript 对此进行设置。JavaScript 也可用于显示和隐藏图例。可排列可滚动的图例。</p> <p>主题图例和制图图例在 LegendControl 中均受支持。</p> <p>Web 示例应用程序 Legend Control 支持该控件。它演示了如何根据当前的 Web 控件体系结构创建和使用自定义的 LegendControl 以及如何通过使用 JavaScript 向服务器发送请求来创建主题并显示图例，而无需刷新整个页面。有关详细信息，请参阅“程序”菜单中的 MapXtreme 学习资源浏览器或 MapInfo 网站的“支持和培训”。展开“学习资源”链接并单击“示例应用程序”。</p>
地图工具	描述
CenterTool	可用于通过单击地图而重定地图的中心。
DistanceTool	可用于通过单击地图来获取两点或多点之间的距离。
NavigationTools	您可按照固定方向平移地图：North、South、East、West、NorthEast、NorthWest、SouthEast 和 SouthWest。
PanTool	可用于通过以任意方向拖动地图在窗口中重新定位地图。

PointSelectionTool	用于在单击地图时可以选择图元（离该点最近的图元）。
PolygonSelectionTool	可用于选择其质心位于多边形内的所有图元。使用鼠标单击多边形中的节点，即可在地图上绘制选择区域。双击会封闭/结束该多边形。
RadiusSelectionTool	可用于选择其质心位于半径范围内的所有图元。使用鼠标单击圆中心和边界的表示，在地图上绘制半径。可通过点击和拖动操作在地图上绘制选择半径。
RectangleSelectionTool	可用于选择其质心位于矩形内的所有图元。可通过点击和拖动操作在地图上绘制矩形选择区域。
ZoomBarTool	可用于将地图缩放至介于 500 与 12,500 地图单位之间的一系列预设置的级别。
ZoomInTool	可用于通过单击或选择某矩形区域来缩放地图。
ZoomOutTool	可用于绘制表示在地图上视图要缩小到的矩形。
InfoTool 示例 Web 应用程序	示例 Web 应用程序中包括 InfoTool 示例。该示例说明了如何基于 Web 控件体系结构创建并使用自定义的地图工具。

Web 控件的体系结构

MapXtreme Web 控件体系结构遵循 ASP.NET 模型来创建 Web 应用程序。通用体系结构是模型-视图-控制器 (MVC) 设计模式，在该模式中，web 应用程序表示模型，web 页面 (HTML, JavaScript) 表示视图，响应信息请求的 MapXtreme Server 则表示控制器。

用户通过 web 控件和工具与 web 应用程序进行交互，捕获数据并将指示和命令发送至可处理和返回该等信息的服务器。

在 MapXtreme 中，在初始化期间渲染 web 页面时会渲染控件和工具。完成初始化后，使用工具时仅渲染地图图像。这些支持 JavaScript 的工具是对 v6.5 之前版本 web 控件的改善，这些控件在每次进行工具操作时均会触发全页回贴。

MapXtreme 控件提供了下列行为和功能：

- 背景地图是通过预定义的工作空间加载的。在设计阶段，可以将 MapControl MapAlias 属性设置为在预装的工作空间中定义的地图的别名。在运行阶段，可将相应的地图加载进 MapControl，以便用户能使用地图工具与其进行交互。
- 第一次渲染页面时会调用 Page_Load 和 Page_Unload。如果会话是新的，Page_Load 将应用程序的状态初始化为默认状态，如果会话不是新的，则将恢复状态。Page_Unload 恢复任何变更的状态，以期望另一次请求。

- **StateManager** 在应用程序中作为类实现，并会将该类的实例放入 ASP.NET 会话中。**SaveState** 和 **RestoreState** 方法从该对象调用。每次使用工具时均会调用 **SaveState** 和 **RestoreState** 方法。若状态管理使用 **Manual** 模式，则 **StateManager** 类实例必须位于该会话中。（在 **Web.config** 文件中将 **MapInfo.Engine.Session.State** 设置为 **Manual**。）有关状态管理的更多信息，请参阅**状态管理 (State Management)**。
- 错误处理在 **application_error** 事件处理程序的 **global_asax.cs/.vb** 文件中执行。请参阅**错误管理**。
- 可通过将请求发送至服务器的客户端 **JavaScript** 命令对事件进行处理。服务器端命令类执行服务器端处理。请参阅**事件处理**。

Visual Studio MapXtreme Web 模板随附的 **MapXtremeWebResources** 文件夹包括有 **web** 控件和工具的附属文件。请确保在部署应用程序时纳入这些文件。

地图工具体系结构

MapXtreme 地图工具用于以某种方式与地图进行交互，例如，规划另一个视图或选择地图的某区域以收集用于进一步分析的数据。MapXtreme Web 应用程序模板中提供了基本导航工具集。这些工具及用于选择地图图元和创建图例的其他工具，均可在 Visual Studio 工具箱中找到。

地图工具由客户端组件和服务器端组件组成。客户端的工具拥有一个 **JavaScript** 交互组件和一个 **JavaScript** 客户端命令组件。服务器端的工具拥有服务器命令类组件。

客户端地图工具组件负责：

- 绘制和鼠标操作（例如橡皮圈矩形和鼠标单击）
- 从鼠标操作收集数据（例如，获取放大操作的屏幕坐标）
- 将 url 请求发送至服务器

服务器地图工具组件负责：

- 执行工具的业务逻辑（例如，计算两点之间的距离）

客户端地图交互

客户端地图工具交互是通过 **JavaScript** 类命令实现的。任何用户接口均可使用这些通用类在任何 HTML 元素上执行交互。可扩展 **Interaction.js** 中的基本 **Interaction** 类，以创建诸如 **ClickInteraction** 和 **RectInteraction** 之类的所有独立交互。用于交互的构造函数是：

```
Interaction(elementID, onComplete)
```

其中，**elementID** 是地图的 **IMG** 标记，**onComplete** 是完成交互时所调用的函数。

例如，**Interaction** 类可在地图上绘制橡皮圈矩形并收集其中包含的所有点图元。

客户端命令执行

地图工具拥有负责执行特定任务的客户端 **JavaScript** 命令对象。可扩展 **Command.js** 中的基本 **Command** 类，以创建诸如 **PanCommand** 和 **ZoomCommand** 之类的独立命令。交互完成后，工具会执行客户端命令。**Command** 类的构造函数是：

Command(name, interaction)

其中，name 指服务器端 Command 类名称，interaction 指在客户端交互期间收集到的数据。

服务器在接收到客户端 Command 生成的 URL 请求后处理响应以显示新的地图。

此交互对象可以是空。这意味着没有客户端交互，即没有可以自动发出命令的交互，例如平移、缩放、点选择等。命令仍然可以发出，但必须通过编程方式才能发出。

服务器端命令体系结构

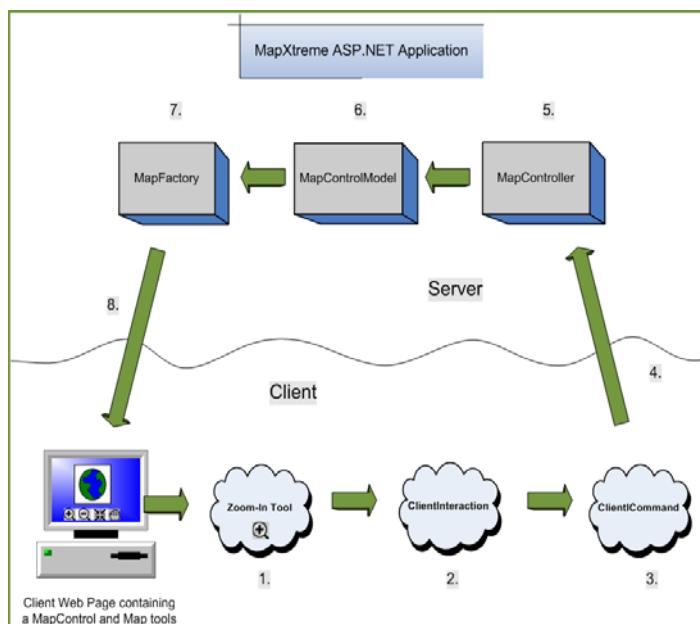
在服务器端，`MapInfo.WebControls.MapControlModel` 类会处理从客户端发出的 URL 请求。该类中包含的方法可用于执行诸如平移和缩放之类的基本地图导航命令及用于在绘制的矩形和半径中选择点、多边形和图元的选择命令。`InvokeCommand` 方法定位指定的工具命令并调用其 `Execute` 方法。`Execute` 会先调用 `RestoreState`，然后调用 `Process`，最后调用 `SaveState`。

要执行该类未提供的命令，必须编写可扩展 `WebControls.MapBaseCommand` 类的 Command 类。

地图工具的功能

下文描述了 web 地图工具（此情况下指“放大”工具）的典型请求/响应周期。可使用该体系结构创建定制地图工具。请参阅[创建定制工具](#)。

图中的编号与下述各阶段相对应。



1. 用户可使用“放大”工具在 MapControl 中显示的一般欧洲区域周围绘制矩形。
2. 称为 RectInteraction 的工具的 ClientInteraction 属性收集可定义该矩形的屏幕坐标。RectInteraction 是在 Interaction.js 中定义的。

3. 工具的 ClientCommand 属性 MapCommand 创建 URL 请求，并将其分配至 MapControl 的 image.src 属性。ClientCommand 是在 Command.js 中定义的。URL 请求如下：
`MapController.ashx?Command=Zoom&Width=300&Height=300&ZoomLevel=1200&Points=1,50,100&MapAlias=Map1`
4. URL 请求将被发送至服务器。
5. MapController 接收并调用 MapControlModel。MapController 源自于 System.Web.IHttpHandler。
6. MapControlModel 解析 URL 请求并调用 ZoomInCommand 类。
7. ZoomInCommand 从 MapFactory 获取地图，并对地图执行放大操作。地图图像将得以更新以反映新的视图。
8. 可将地图图像导出至图像中，将其作为数据流写入到 HTTP 响应中，并将其返回至客户端。

使用 MapXtreme Web 控件

要在 web 应用程序中使用 MapXtreme Web 控件：

1. 请执行下列任一项操作：
 - a. 从 Visual Studio “文件” > “新建项目” 创建 MapXtreme Web 应用程序，并选择 MapXtreme Web 应用程序模板。MapControl、LayerControl 和地图导航工具均是内置的。
 - b. 打开任一个示例 web 应用程序，然后根据自己的需求对其进行修改。
 - c. 将所需工具从 Visual Studio 的 MapXtreme Web 7.x.x 工具箱拖放到 Web 窗体中。
 - d. 从 Visual Studio “文件” > “新建项目” 创建 MapXtreme Web 应用程序，并选择通用 ASP.NET Web 应用程序模板¹。
2. 如果工作空间包含要预载到 MapControl 中的地图图层，请在应用程序的 web.config 文件中指定该工作空间的路径和名称。不管您在步骤 1 中作出何种选择，均需要执行此操作。

模板和示例的 web.config 文件包含了安装的示例数据的默认位置的路径，如下所示：

```
<configuration>
  <appSettings>
    <add key="MapInfo.Engine.Session.Workspace" value="c:\Program
      Files\MapInfo\MapXtreme\7.x.x\Samples\Data\World.mws" />
  </appSettings>
</configuration>
```

应用程序需要的多个地图可能存在于一个或多个工作空间中，每个地图的 MapAlias 属性均是唯一的。要在 web.config 文件中预载多个工作空间，请在每个工作空间的完整路径之间使用分号。

-
1. 在此情况下，您会注意到 MapXtreme Web 控件和工具在 Designer 中显示为红色 X 而不是其图标。要正确显示这些图标，请从其中一个 MapXtreme 示例应用程序复制 MapXtremeWebResources 文件夹，然后将其粘贴到包含 web.config 和 default.aspx 文件的项目中。关闭然后重新打开网页查看这些图标。要避免此手动步骤，请改用选项 “a”。

3. 请确保设置 **MapControl** 的 **MapAlias** 属性。**MapControl** 在模板中的默认 **MapAlias** 为 **Map1**。如未设置该属性，**MapXtreme** 渲染的第一幅地图可能并非您期望的地图会话。
要确定 **MapAlias**，请在文本编辑器中打开 **.mws** 文件，并查找 **MapDefinition** 元素。**MapAlias** 存储为别名属性。
也可从 **Workspace** 管理器中查找 **MapAlias**。当鼠标指针在地图节点（图层列表中的顶部节点）上悬停时，别名出现在 **ToolTip** 中。
4. 请为所有地图工具、**LayerControl**、**LegendControl** 和自定义控件（如有）设置 **MapControlID** 属性，以指向合适的 **MapControl**。

管理 Web 控件

使用 Web 控件的重要部分就是对其进行有效管理。本节解释了如何执行：

- 事件处理
- 错误管理
- 状态管理 (**State Management**)

事件处理

可通过将请求发送至服务器的客户端 JavaScript 命令处理地图工具事件。在服务器端，源自于 **MapBaseCommand** 的命令类为该工具执行处理。

通常，服务器端处理的结果会发回客户端。如使用 **MapControl**，它将是图像；如使用 **LayerControl**，它将是 XML。然后，Web 页面只有一部分会通过客户端 JavaScript 和命令结果一起更新（例如，新的地图图像在平移后显示）。

错误管理

可通过多个方法处理 Web 控件中的错误处理，且错误处理完全特定于应用程序。因此，本节仅说明处理错误的其中一种方法。由于客户端所期望的响应是图像，因此可以使用详细消息捕获说明并将响应与图像以及错误消息一同发送回去。因此，**MapControl** 现在将会包含错误消息。

我们的示例应用程序随附的 **Global.asax** 文件演示了处理应用程序错误的实例。如果该应用程序在处理地图图像请求时遇到错误，**Application_Error** 方法就会创建包含错误消息的图像，并将其返回至客户端。

有关错误管理的详细信息，请参阅 **MSDN** 站点上关于使用 **ASP.NET** 进行错误处理的内容。

状态管理 (State Management)

`MapInfo.WebControls.StateManager` 抽象类包括 `SaveState` 和 `RestoreState` 方法。在处理前后，`MapXtreme` 地图工具会分别调用 `RestoreState` 和 `SaveState`。由于状态管理是特定于应用程序的，因此您的责任是在应用程序的具体类中实现这些方法。这将允许用户控制恢复和保存哪些内容以及如何恢复和保存这些内容。

有关状态管理的详细信息请参阅[第 6 章：了解状态管理](#)。

创建定制工具

要创建定制工具，可修改或添加内置工具的行为，或编写您自己的定制命令和工具。

例如，您可能想要修改“放大”工具以放大并通过单击选择某图元。该工具需要包含了代码的服务器命令类，以放大和执行选择。因为我们提供了所有服务器命令类的源代码，所以，您可轻易地修改 `ZoomIn` 命令类以添加选择代码。

如果我们的源代码未为您的自定义提供起点，则必须编写您自己的命令和工具类。源代码位于 `MapXtreme` 安装中，它的路径如下：`\Samples\MapXtremeWebControlsSourceCode`。

规划自定义时，请记住下列 `MapXtreme` 工具体系结构。`MapXtreme` 地图工具由下列部分组成：

- 继承于 `MapInfo.WebControls.WebTool` 的客户端工具类。
- 控制工具行为的工具类属性，包括：
 - 描述了工具与 `MapControl` 之间交互（点击和绘制矩形等）的 `JavaScript`。
 - 为工具创建 `url` 请求的 `JavaScript`。
- 从执行所需工具行为的 `MapInfo.WebControls.MapBaseCommand` 派生的服务器端命令。

在类中或在 `web` 页面上定义的工具的属性。`Web` 页面上按工具类名称引用了定制工具。

以下步骤所需的代码位于 `Samples` 文件夹下面，它的路径如下：

`\Web\Features\CustomTools\CustomToolsCS`。

要创建定制工具：

1. 将通用 `WebTool` 从 `MapXtreme` 工具箱拖至您的 `web` 窗体。如果想要扩展现有的行为，也可使用任一种现有的地图工具。
2. 在 `WebTool` 属性页面上，设置 `MapControlID`、`InActive/ActiveImageUrl` 和 `CursorImageUrl` 的属性。
3. 通过从下拉列表中作出选择，设置合适的 `ClientInteraction` 属性。

内置交互包括诸如点击、拖动及绘制线段、多边形、矩形和半径之类的鼠标操作，可满足大多数 `web` 应用程序的需求。请参阅您的项目中 `MapXtremeWebResources` 文件夹的 `Interaction.js`。

4. 通过从下拉列表中作出选择，设置合适的 ClientCommand 属性。

用于地图绘制、平移和缩放等的内置客户端命令可创建将发送至服务器的 URL 请求。有关这些命令的描述，请参阅您的项目中 MapXtremeWebResources 文件夹的 Command.js。

如果内置命令都不符合您的要求，可修改现有的 Command.js 或编写您自己的命令。自定义命令将步骤 3 中的交互名称视为输入。有关如何通过单击从服务器获取多个响应的示例，请参阅 CustomToolsCS 或 CustomToolsVB 实例中的 CustomCommand.js。

5. 在 .aspx 页面中手动注册 JavaScript。在 web 页面正文中插入下列行。

```
<script language="javascript" src="CustomCommand.js"
type="text/javascript"></script>
```

6. 创建从 MapInfo.WebControls.MapBaseCommand 派生的新服务器命令类。包括执行客户端命令请求的行为的代码。或者，扩展现有的服务器命令类。

7. 在服务器命令类中，分配构造函数中服务器命令的名称。

```
namespace ToolsSample
{
    public class AddPinPointCommand : MapInfo.WebControls.MapBaseCommand
    {
        /// <summary>
        /// Constructor for this command, sets the name of the command
        /// </summary>
        /// <remarks>None</remarks>
        public AddPinPointCommand()
        {
            Name = "AddPinPointCommand";
        }
    }
}
```

8. 在服务器命令类中，添加执行命令业务逻辑的代码以覆盖 Process() 方法。

```
public override void Process()
{
    // Your code here.....
}
```

9. 在 webform1.aspx 的 Page_Load 方法中，将服务器命令添加至 MapControlModel 中的命令集合。

```
MapInfo.WebControls.MapControlModel controlModel =
    MapControlModel.SetDefaultModelInSession();
controlModel.Commands.Add(new AddPinPointCommand());
```

10. 通过创建新的程序集将工具添加至 Visual Studio 工具箱。请参阅[使用和分发定制 Web 控件](#)。

11. 根据步骤 3 和步骤 4 中对 JavaScript 所做的设置，将自定义工具拖放至 web 窗体，并将属性设置为 ClientCommand 和 ClientInteraction 的名称。

12. 按步骤 6 中的定义，设置服务器 Command 的属性。

13. 通过指定该控件所处的程序集和命名空间，在 web 窗体中注册标记前缀。

```
<%@ Register TagPrefix="cc1" Namespace="MapInfo.WebControls"
Assembly="MapInfo.WebControls, Version=4.0.0.476, Culture=neutral,
PublicKeyToken=0a9556cc66c0af57" %>
<%@ Page language="c#" Inherits="ToolsSample.WebForm1"
CodeFile="WebForm1.aspx.cs" %>
<%@ Register TagPrefix="cc2" Namespace="CustomizedWebTools" %>
```

如果 WebTool 的现有功能不能满足您的需求，可以编写自己的服务器命令类和 Javascript，以处理客户端的命令和交互。有关示例，请参阅[将 InfoTool 添加到 Web 应用程序](#)。

使用和分发定制 Web 控件

一旦创建了定制 Web 工具，必须将其纳入程序集中，以便使其存在于 Visual Studio 工具箱中或可将其分发至其他工具箱。

MapXtreme 提供 Web 控件的源代码，以便您进行学习或修改操作，也可以根据需要分发它们。Web 控件的源代码安装在 \Samples\WebControlsSourceCode 文件夹中。如要使用和分发已修改的 Web 控件，必须创建新的程序集，并在 Visual Studio 注册该程序集。

无论修改 MapXtreme 源代码还是重新创建自己的工具类，对于 MapXtreme web 控件程序集，请考虑下列重要因素：

- 程序集名称是 `MapInfo.WebControls.dll`，安装在全局程序集缓存中。该程序集具有特定的版本号，由模板和示例应用程序使用。
- 程序集中的控件安装在 Visual Studio 的工具箱中。
- 程序集具有对 `MapInfo.CoreEngine.dll`、`MapInfo.CoreEngine.Wrapper.dll` 和 `MapInfo.CoreTypes.dll` 的引用。
- 诸如图像和脚本之类的资源安装在 `C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x\MapXtremeWebResources` 文件夹中。
- 以下文件由 Web 控件使用：
 - * 表示工具操作的 GIF 图像（例如，`DistanceToolControlActive` 和 `DistanceToolControlInactive`）。
 - * 表示工具图标的 BMP 图像（例如，标注和选择箭头）
 - 定义工具交互和行为的 JavaScript（`Interaction.js`、`Command.js`、`LayerControl.js`、`LegendControl.js` 和 `Tool.js`）
 - * 在使用鼠标时显示图像的 CUR（光标）文件。

创建 Web 程序集

要创建定制 web 控件程序集：

1. 将自定义的 web 控件源文件复制到另一目录，以保留原始内容。

2. 从全局程序集缓存和 Visual Studio 工具箱中删除原始程序集。

程序集位于 `C:\Program Files\Windows\Assembly\GAC_32 or GAC_64` 中，具体取决于安装的是 32 位还是 64 位 MapXtreme。

3. 使用 `sn - k MapInfo.WebControls.snk` 创建强制命名键文件 (`.snk`) 并将该键文件复制到主项目文件夹（与项目文件位于同一级别）。

4. 更改 `AssemblyInfo.cs` 或 `AssemblyInfo.vb` 文件，以反映您的 web 程序集的版本号。

5. 在 Visual Studio 中打开项目，进行所需的更改并构建项目。新的程序集应位于 bin\Release 目录中，因此可以分发程序集的发行版。
6. 以下列语法作为模型，在全局程序集缓存和 Visual Studio 工具箱中注册新的程序集。有关详情，请参阅[全局程序集缓存工具](#)。
`gacutil /i MapInfo.WebControls.dll`
7. 使用新的程序集编写自己的应用程序。将新的控件从 Visual Studio 工具箱拖放至窗体。
8. 在 Web 应用程序的安装程序中，确保新的程序集安装在全局程序集缓存中。只要 Web 应用程序指向此版本的程序集，其就会使用新的控件。

可以考虑使用以下方案：

- 可以将 Web 控件项目和资源权限插入 Web 应用程序解决方案中。在这种情况下，更改资源（脚本和图像）的 URL 以从项目的根目录开始。这将防止创建虚拟目录。
- 不必使用全局程序集缓存和强制命名的程序集。可以将程序集的 Copy 属性设置为 true，并使程序集位于应用程序的 bin 文件夹中。

将 InfoTool 添加到 Web 应用程序

MapXtreme 提供的 InfoTool 示例应用程序可以进行修改以便用于 Web 应用程序。使用此工具可在用户单击地图时捕获信息，并获取从 Web 应用程序返回的信息。

按以下步骤修改 InfoTool 示例。

1. 进入示例文件夹（默认位置：C:\Program Files\MapInfo\MapXtreme\7.x.x\Samples\VisualStudio20xx\Web\Features\InfoTool\InfoTool\VB）找到 InfoToolCS 或 InfoToolVB Web 应用程序。
2. 将以下文件复制到项目文件夹并将它们添加到项目：
 - CustomCommand.js（InfoTool 文件夹的根目录下）
 - CustomizedCommands.cs 或 CustomizedCommands.vb（\App_Code 文件夹中）
 - stylesheet.css（InfoTool 文件夹的根目录下）
3. 将 PointSelect 工具添加到 Web 窗体，并将以下代码添加到窗体的 Page_Load 方法中：

```
MapInfo.WebControls.MapControlModel controlModel =
    MapControlModel.SetDefaultModelInSession();
controlModel.Commands.Add(new CustomWebTools.Info());
```
4. 使 PointSelect 工具的属性与示例应用程序中 InfoTool 的属性相匹配。在工具的“属性”窗口中设置属性。
这些属性包括：ClientCommand、ClientInteraction、Command、CursorImageURL、InactiveImageURL 和 MapControlID。
5. 将窗体切换到 HTML 视图，并将以下一行内容添加到 <form> 标记后面：

```
<script language="javascript" src="CustomCommand.js"
type="text/javascript"></script>
```

6. 添加类似于示例应用程序中的 <div>, 以保存从工具中获取的信息。

```
<div id="Info" class="infoDiv">  
    Div element to display selected feature information in html  
    table.</div>
```

7. 构建 Web 应用程序。

ASP.NET AJAX 和 MapXtreme Web 应用程序

MapXtreme 的 Web 控件和工具包括的 JavaScript 可为 Web 应用程序提供高效的请求/响应周期。每当使用地图工具时, JavaScript 交互和命令均可执行操作而无需对客户端触发全页面回贴。通常仅刷新地图图像。

Microsoft 的 ASP.NET AJAX 技术通过集成脚本库与 ASP.NET Framework 进一步提升了该行为。UpdatePanel 和 ScriptManager 是主要控件, 前者容纳了频繁刷新的服务器控件, 后者管理 web 页面的脚本活动。

MapXtreme 提供的示例应用程序演示了如何使用 MapXtreme web 映射应用程序中 Microsoft 的 ASP.NET AJAX 控件。该示例的位置如下:

..\\MapInfo\\MapXtreme\\7.x\\Samples\\VisualStudio2005\\Web\\Features。



AJAXDemo 示例需要在系统上安装 Microsoft ASP.NET AJAX Extensions 1.0 或更高版本。

以下部分介绍了将 AJAX 控件添加到 MapXtreme 应用程序的步骤。

在继续之前, 作为了解 AJAX 的一种练习, 不妨使用 AJAX 扩展随附的“支持 ASP.NET AJAX Web 站点”模板创建 Web 应用程序。请检查其 web.config 文件; httpHandlers 部分和 httpModules 部分包含了可能需要复制到应用程序的设置。

同时, 您应熟悉 AJAXDemo 示例 web 应用程序。随后的步骤参考了该示例中的 JavaScript 代码和 web.config 设置。

将 ASP.NET AJAX 控件添加至 MapXtreme Web 应用程序

要将 ASP.NET AJAX 控件添加至现有的 MapXtreme web 应用程序:

1. 使用 Visual Studio 的 Design 模式打开 Web 窗体。
2. 从 Visual Studio 工具箱的“AJAX 扩展”部分, 将 ScriptManager 控件拖至窗体。(放置 ScriptManager 的位置并不重要, 因为其在运行时是不可见的。)
3. 将 AJAX UpdatePanel 控件拖至窗体。

- 移动 UpdatePanel 内诸如 Button 控件之类的标准控件，以防止 Button 引起全页更新。



不要移动 UpdatePanel 内部诸如 MapControl 或 LayerControl 之类的 MapXtreme 控件。有关详细示例，请参阅 AJAXDemo 示例应用程序。

- 打开应用程序的 web.config 文件，并定位 httpHandlers 部分。根据 Web 应用程序内容的不同，httpHandlers 部分可能包括一个或两个条目，一个用于 MapController.ashx（如果您的应用程序包含 LayerControl），另一个用于 LayerController.ashx。

```
<httpHandlers>
  <add verb="*" path="MapController.ashx" ...>
  <add verb="*" path="LayerController.ashx" ...>
</httpHandlers>
```

- 从 AJAXDemo 示例应用程序打开 web.config 文件，并定位其 httpHandlers 部分，该部分包含了 ASP.NET AJAX 所需的附加条目：

```
<httpHandlers>
  <remove verb="*" path="*.asmx"/>
  <add verb="*" path="*.asmx" ...>
  <add verb="*" path="*_AppService.axd" ...>
  <add verb="GET,HEAD" path="ScriptResource.axd" ...>
  <add verb="*" path="MapController.ashx" ...>
  <add verb="*" path="LayerController.ashx" ...>
</httpHandlers>
```

如果 web.config 文件缺少头四个 httpHandler 条目中的任何一个，请从 AJAXDemo web.config 文件中复制缺失的条目，并将其粘贴在您的 web.config 文件中。（不必复制 LayerController.ashx 条目；如果您将 LayerControl 置于您的 Designer 模式的页面上，则将自动生成 LayerController.ashx 条目。）

- 定位 web.config 文件的 httpModules 部分。httpModules 部分很可能已经包含一个 MapInfo.Engine.WebSessionActivator 条目。从 AJAXDemo Web.config 文件中复制“ScriptModule”条目，使 httpModules 部分类似如下：

```
<httpModules>
  <add name="ScriptModule" type="System.Web.Handlers.ScriptModule...">
  <add type="MapInfo.Engine.WebSessionActivator...">
</httpModules>
```

如果 UpdatePanel 中的控件以某种方式影响地图，则需要将 JavaScript 添加至页面中，以便对地图图像强制执行更新。AJAXDemo 示例应用程序包含的示例 JavaScript 演示了在指定的 UpdatePanels 引起页面更新时如何更新地图图像。

- 以 Source 模式打开 AJAXDemo 应用程序的 MapForm.aspx 页面。复制 <script> 块，并将其粘贴至 aspx 页面。注意：必须将 <script> 块粘贴在 ScriptManager 标记的后面，因为脚本使用了 ScriptManager 提供的对象。
- 在 <script> 块中，删除 DisplayEventInfo 功能及对其作出的任何调用。DisplayEventInfo 功能是适用于 AJAXDemo 应用程序的调试工具，其他应用程序并不需要。

10. 如果重命名了 **UpdatePanel**，请编辑 **<script>** 块，以使用新的 **UpdatePanel** 名称。（**UpdatePanel** 名称作为第二个参数被传递至 **TargetPanelWasUpdated** 功能，默认的名称为“**UpdatePanel1**”。）

11. 如果在应用程序中使用了不止一个 **UpdatePanel**，但只想部分 **UpdatePanels** 影响地图，则应将每个 **UpdatePanel** 的 **UpdateMode** 属性设置为 **Conditional**。有关详情请参阅 **AJAXDemo** 示例应用程序随附的 **ReadMe** 文件。

有关 ASP.NET AJAX 扩展所需的 **web.config** 设置的详情，请查阅 Microsoft 的 **ASP.NET AJAX** 文档。

MapXtreme 瓦块句柄

MapXtreme 提供基于 REST 的瓦块句柄和公共 URL，它们可用于请求地图瓦块和信息。使用基于 REST 的句柄，可将地图请求的所有参数都嵌入一个 URL。

在如今的 Web 地图绘制领域，地图瓦块至关重要，因为它们可以预先渲染并予以保存，以便用户随时请求。您可以设计一个瓦块服务器，使它可以像存储静态图像一样存储基本地图，因为这样在用户会话期间就不需要更新。如果数据包含的信息仅针对用户（例如从查询返回的信息），在动态更改此数据时，这些地图将在扉页上生成。因为它们已分成瓦块状，所以只有在地图窗口上可见的瓦块（基于瓦块大小、地图窗口大小和缩放级别）才会返回。

有关 web 应用程序如何使用 **MapXtreme** 瓦块句柄的示例，请参阅《**MapInfo 开发人员代码交流**》中的“**MapXtreme** 瓦块句柄”示例。

MapXtreme 提供以下支持：

- 根据瓦块在地图中的行位置和列位置请求瓦块
- 请求可用的地图列表
- 请求地图描述
- 指定更佳瓦块性能的缓存说明

MapXtreme 瓦块句柄 API 在 **MapInfo.WebControls.Tiling** 命名空间中。详细信息，请参阅《开发人员参考》。瓦块句柄原代码在 **\Samples\WebControlsSourceCode** 文件夹中。

使用 MapXtreme 瓦块句柄

在最简单的形式中，需访问 **MapXtreme** 瓦块句柄并从瓦块服务器请求地图瓦块和信息：

1. 修改 **web.config** 文件以指向句柄，具体如下。

```
<httpHandlers>
    <add verb="*" path="TileServer/*"
        type="MapInfo.WebControls.Tiling.TileHandler" />
        <add verb="*" path="TileServer/*/*"
        type="MapInfo.WebControls.Tiling.TileHandler" />
            <add verb="*" path="TileServer/*/*/*"
            type="MapInfo.WebControls.Tiling.TileHandler" />
                <add verb="*" path="TileServer/*/*/*/*"
                type="MapInfo.WebControls.Tiling.TileHandler" />
                    <add verb="*" path="TileServer/*/*/*/*/*"
                    type="MapInfo.WebControls.Tiling.TileHandler" />
```

```
</httpHandlers>
```

2. 提供包含多个地图的工作空间 (.MWS)。

例如，对于便于用户了解其位置是在范围（手机网络、交易数据、学校地区等）内还是范围外的范围定位应用程序，则包括：

- 基本地图图层，专门针对服务器，可用为背景和引用地图。
- 覆盖地图，其中包括像点位置和道路网络这样的引用点和引用线。
- 范围图层，其中包含您的范围。

3. 使用以下 URL 格式请求可用的地图列表。

```
http://server/TileServer/maplist.{ext}
```

其中 ext 是扩展名，它表示返回信息的格式（当前仅支持 JSON）

此查询将返回 JSON 对象 (JavaScript Object Notation)，格式如下：

```
{
  [
    "Map1Alias",
    "Map2Alias",
    "Map3Alias"
  ]
}
```

4. 使用步骤 3 中所列的地图名称，通过以下 URL 格式请求有关地图的更多信息：

```
http://server/TileServer/{mapname}/description.{ext}
```

其中，mapname 是地图在瓦块服务器上获取元数据时的名称。值不区分大小写。

其中 ext 是扩展名，它表示返回信息的格式（当前仅支持 JSON）

它返回特定地图的元数据。

```
{
  "numberOfLevels": 20,
  "coordSys": "epsg:41001",
  "description": "Map of the World",
  "name": "World",
  "tileWidth": 256,
  "tileHeight": 256,
  "bounds": {
    "minX": -3.756380109844111E7,
    "minY": -4.497601034176671E7,
    "maxX": 3.773376466605809E7,
    "maxY": 4.49763662218225E7
  },
  "outputTypes": [
    "png"
  ]
}
```

5. 收集完步骤 3 和 4 中的信息后，在 URL 中提供所需的全部参数以请求地图（下面将解释）：

```
http://server/webapp/TileServer/{mapname}/{level}/{x;y}/tile.{ext}
```

例如：

`http://<server>/<mywebapp>/TileServer/WorldOverlay/3/0;0/tile.png`

这将请求包含 64 个瓦块的 WorldOverlay 地图提供左上方的瓦块。

参数说明详见下表。

参数	描述
服务器	您的 Web 服务器
webapp	服务器上运行的 Web 应用程序的名称。
TileServer	MapTiling 句柄实例的路径。它必须和 web.config "httpHandlers" 部分中的起始 "path" 条目相符。参阅以上内容
mapname	瓦块服务上的地图名称。
level	请求的瓦块的级别。从零开始。
x;y	请求的瓦块的 x 坐标和 y 坐标（从零开始）。 例如，位于第 3 级别时，地图由 64 个瓦块组成，它分为 8 行和 8 列。左上方瓦块的 x 参数和 y 参数为 0;0。左下方瓦块的参数则为 0;7。
ext	表示瓦块格式的扩展名（例如 .gif、png）。必须和支持的格式相符。

缓存

MapXtreme 瓦块句柄支持缓存常用瓦块，这样应用程序的性能就不会受影响。缓存说明包含在 web.config 文件中，就在 Microsoft 的 .NET Framework **HttpCacheability** 之后。

支持五种类型的缓存（加上一种“不需要缓存”选项）。这些枚举值可用来设置 Cache-Control HTTP 头信息。

NoCache	设置 Cache-Control: no-cache 头信息。如无字段名称，在满足请求前，命令可能会应用于整个请求；而且共享（代理服务器）缓存必须和原始 Web 服务器一同强制执行一次成功的重新验证。如有字段名称，命令仅应用于命名字段；其他的响应可能来源于共享缓存。
私有	默认值。设置 Cache-Control: private 可指定响应仅在客户端上可缓存，在共享（代理服务器）缓存上不可缓存。

公共	设置 <code>Cache-Control: public</code> 可指定响应在客户端和共享（代理）缓存上均可缓存。
服务器	指定响应仅在原始服务器上可缓存。它类似于 <code>NoCache</code> 选项。客户端将接收 <code>Cache-Control: no-cache</code> 命令，但文档在原始服务器上可缓存。它类似于 <code>ServerAndNoCache</code> 。
<code>ServerAndNoCache</code>	同时应用 <code>Server</code> 和 <code>NoCache</code> 的设置，以指定内容在服务器上可缓存，但所有其他方均不得缓存响应。
<code>ServerAndPrivate</code>	表示服务器和客户端可缓存响应，但所有其他方均不得缓存。不允许代理服务器缓存响应。

要指定缓存选项，请修改 `web.config` 文件以指向键 `MapInfo.Engine.Session.Cacheability`，具体如下。

```
<appSettings>
    <add key="MapInfo.Engine.Session.Cacheability"
value="private"></add>
</appSettings>
```

缓存结果失效日期

也可以设置一个缓存瓦块失效的日期。设置的失效日期一到，瓦块中的数据将需要刷新。通过设置瓦块的失效日期，应用程序的用户只会收到显示的最新信息。

要设置缓存结果失效日期，请将键添加到 `web.config` 文件，具体如下：

```
<appSettings>
    <add key="MapInfo.Engine.Session.CacheExpires"
value="4/1/2010"></add>
</appSettings>
```

值可以是 Microsoft `DateTime.Parse(String)` 方法可成功分析的任何字符串。

注意：根据 Microsoft 的限制，有效期不得超过一年。如超过一年则无效。

HTML/XHTML 验证问题

如果您通过验证服务创建了 `MapXtreme` web 应用程序并运行了生成的 HTML，则可能会看到下列验证错误，这具体取决于您所使用的 DOCTYPE 标记版本：

```
value of attribute "ID" invalid."_" cannot start name
```

该验证错误与隐藏的属性字段相关

```
id="__VIEWSTATE"
```

引起该验证错误的 `id` 属性不是 `MapXtreme` 而是 `ASP.NET` 输出的。

要解决该验证错误，需要替换.aspx 页面上的 DOCTYPE 标记。具体而言，如果将 DOCTYPE 标记更新为 XHTML DOCTYPE 标记，生成的页面将生效，即使已显示上述 id 属性。（ASP.NET 将打包 DIV 标记中的不良标记，将根据 XHTML DOCTYPE 对 DIV 标记中的所有标记进行验证。）举个例子，当您从 Visual Studio 的 ASP.NET 模板创建新的 Web 应用程序时，可能会使用为您生成的相同 DOCTYPE 标记：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

将 Post-back Web 控件迁移到 JavaScript Web 控件

从 v6.5 以前版本的 Web 控件¹无法自动迁移为 6.6 版及更高版本 MapXtreme 中的 Web 控件。每个应用程序和迁移过程都不相同，并且取决于应用程序与旧的 Web 控件的功能和设计的紧密联系程度。

建议分阶段进行迁移过程。在某些情况下，可能需要重新调整代码结构。迁移 Web 应用程序时需要考虑下列过程：

- 加载数据
- 替换控件
- 状态和事件管理

加载数据

使用预载的工作空间是通过 MapXtreme Web 控件加载数据的唯一方式。在该工作空间中 MapControl 会指向 MapAlias 并尝试显示地图，而不加载地图。预载的工作空间可在 web.config 文件中指定。请参阅[初始地图看起来应该是什么样子？](#)。

在 MapControl 用来显示多个地图图像的情况下，设置该数据以便所有地图位于同一位置，并更改 MapControl 的 MapAlias 以选择地图。

替换控件

可通过各种方式来替换控件，如通过从窗体中删除旧的 Web 控件并将新的 Web 控件拖放到窗体上，或通过创建新的 webforms、拖放新的控件，然后将旧窗体的功能添加到新的窗体。请确保为 MapControl 设置适当的 MapAlias 并为所有附属工具和 LayerControl 设置适当的 MapControlID。

1. 在 6.5 以前的 MapXtreme 版本中，web 控件要求回贴页面每当使用工具时调用 Page_Load 和 Page_Unload。这些控件已由更具效率的支持局部更新的 JavaScript 控件所替代。我们提供可向后兼容的 postback 控件，但它们被标记为“已过时”。有关这些控件的信息，可在 MapInfo.Web.UI.WebControls 命名空间中找到。有关 JavaScript 控件 API，请参阅 MapInfo.WebControls 命名空间。

状态和事件管理

在大多数情况下，状态管理的老方法是在 `Page_Load` 中恢复状态，在 `Page_Unload` 中保存状态。必须将该代码移至新的类，该类从 `StateManager` 派生到 `RestoreState` 和 `SaveState` 方法。

如果您的 web 应用程序手动处理状态，则必须执行 `StateManager` 并将其放入 ASP.NET 会话中。手动处理状态是最佳 `MapXtreme` 实践，以便您能仅恢复所需的信息。请参阅[第 6 章：了解状态管理](#)。

在 web 页面中使用 `Server.Transfer` 或 `Response.Redirect` 的情况下，该会话不再是新的。若 `StateManager` 先前不存在，则必须将其放入 ASP.NET 会话中。

Web 控件专题

以下各节介绍了 `MapXtreme` Web 控件的特殊用途：

- 在框架中使用 Web 控件
- 在表单元格中使用 MapControl
- Web 控件本地化

在框架中使用 Web 控件

`MapXtreme` Web 控件在框架中运行。有关如何创建框架、框架集及将页面分配至框架的信息，请参阅 Visual Studio 文档。

在框架中使用 Web 控件时，请记住框架指向 web 页面并且框架集包含一个或多个框架。可以考虑使用以下方案：带有 `MapControl` 的页面以及带有工具或取决于 `MapControl` 的 `LayerControl` 的其它页面。所有页面组成了框架集。

给定上述的方案后，将会应用以下规则：

- 对于附属的控件，必须手动将 `MapControlID` 输入到属性中。若同一页面上的 `MapControl` 与附属的控件具有相同的 ID，则工具和 `LayerControl` 将选取那个控件。
- 由于框架页面需根据特定顺序进行渲染，对于在第一个框架之后渲染的框架来说，`MapXtreme` 会话不会是新的会话。为确保 `StateManager` 位于正确位置，请执行以下代码。无论加载哪一个页面，都务必确保有一个已注册的 `StateManager` 类。在调用 `RestoreState` 前应执行以下代码。

```
' If the StateManager doesn't exist in the session put it in, else get it.  
If StateManager.GetStateManagerFromSession() Is Nothing Then  
    StateManager.PutStateManagerInSession(New AppStateManager())  
End If
```

在表单元格中使用 MapControl

由于 HTML 行为，一旦将某元素拖离另一元素，该元素就会调整为基本大小（主要是 0）。若将 MapControl 放入表单元格，则元素在 HTML 中没有绝对宽度和高度时可能会有问题。将元素拖离单元格时，该单元格会折叠起来，因此单元格大小变为 0。

要解决此问题，请明确设置 MapControl 高度和宽度。下面的示例说明如何在 HTML 中设置大小：

```
<table style="Z-INDEX: 101; LEFT: 32px; POSITION: absolute; TOP: 64px"  
borderColor="#ff00ff" border="1">  
  <TR bordercolor="#ff3366">  
    <TD bordercolor="#0066ff">  
      <cc1:mapcontrol id="Mapcontrol2" runat="server" Width="300px"  
Height="300px"></cc1:mapcontrol>  
    </TD>  
  </TR>  
  <TR>  
    <TD>  
      <cc1:pantool id="Pantool2" runat="server"  
MapControlID="MapControl2"></cc1:pantool>  
      <cc1:zoomintool id="ZoomInTool1" runat="server"  
MapControlID="Mapcontrol2"></cc1:zoomintool>  
      <cc1:zoomouttool id="ZoomOutTool1" runat="server"  
MapControlID="Mapcontrol2"></cc1:zoomouttool>  
    </TD>  
  </TR>  
</table>
```

Web 控件本地化

MapXtreme 提供 Visual Studio 解决方案，以方便开发人员转换和 Web 控件相关的文本字符串。该“本地化工具包”包含 MapXtreme 中所有运行时组件的资源项目。它不包括设计阶段资源。

在每个项目中，均包含可转换的英语资源字符串和强制命名键 (.snk) 文件。将编译到程序集中，以便集成到 MapXtreme 应用程序。MapXtreme Web 控件包含在名为 MapInfo.WebControls.resources 的项目中。

有关如何从本地化 Web 控件资源创建卫星程序集的说明，请参阅[附录 K: 本地化工具包](#)。

6

了解状态管理

本章介绍状态管理的概念和最佳实践，这是撰写 Web 应用程序时十分重要的一个主题。

在本章中：

◆ 概述	100
◆ 术语	100
◆ 什么是 状态管理	101
◆ InProc 开发模型	104
◆ 对象池的状态管理	106
◆ 手动状态管理详探	111
◆ 关于 MapXtreme Session 的深入讨论	116

概述

在设计和实现 **MapXtreme web** 应用程序时，需要重点考虑状态管理。当 **web** 站点访问者与您的 **web** 应用程序交互时，访问者进行更改的方式对获得成功的用户体验以及建立一个可伸缩的高性能应用程序而言殊为关键。

在深入探讨本主题前，请务必阅读下一节内容，以便继续学习本章后续内容。了解 **MapXtreme Session** 与 **HTTP** 会话、用户状态与应用程序状态、干净与不干净的 **MapXtreme Session** 对象之间的差异，以及更多内容，将帮助您更好地准备好从一开始就成功地规划并建立您的 **web** 应用程序。

术语

MapXtreme Session - `MapInfo.Engine.Session` 对象，用于保存 Catalog、`MapFactory` 和 `CoordSysFactory`。用户与 **MapXtreme Session** 对象的实例交互。

HTTP 会话 - `System.Web.HttpSessionState` 对象，用户在多次请求之间的更改就保存在这里。将为每个请求保存和恢复这些更改。用户的更改称为用户状态。

浏览器会话 - 唯一的用户与 **web** 应用程序交互的时间段。也称为 **ASP.NET** 会话。

InProc 开发模型 - 一种 **Web** 应用程序开发模型，每位用户都在其中拥有自己的 **MapXtreme Session** 对象。用户在浏览器会话过程中进行的任何更改都不会影响其他用户。整个 **MapXtreme web** 应用程序连同当前应用程序状态一起存储在内存中，并与此用户相关联。此模型对于用户数量已知的小型低负载 **Web** 应用程序（例如某个部门的内联网）很有用处。与可伸缩性更好的对象池开发模型相比较，用户在此模型中可共享 **MapXtreme Session** 和系统资源。请参阅第 104 页中的 **InProc** 开发模型。

对象池 开发模型 - 在该模型内，可从与 **web** 应用程序相关联的 **COM+** 提供多个 **MapXtreme Session** 实例，且这些实例会被激活以便为 **web** 请求提供服务。每个对象池的 **MapXtreme Session** 对象都用于处理来自多个 **web** 用户的请求。此模型比 **InProc** 模型更复杂，因为您的应用程序必须为每位用户管理 **MapXtreme Session** 的状态。不过这样可以更有效地利用系统资源。使用此模型来建立可伸缩的应用程序。请参阅第 106 页中的 **对象池的状态管理**。

状态管理 - **Web** 应用程序开发的常用术语，用于保存和恢复来自浏览器会话的信息。

背景地图 - 预装有 `web.config` 文件中定义的 **web** 应用程序的初始地图。该地图包含参考图层（如街道网络和邮政边界）和特定于应用程序的数据（如商店位置或手机信号发射塔）。此基本 `workspace` 将预装在 **MapXtreme Session** 实例中并可供使用，除非应用程序允许用户更改基本地图。请参阅第 102 页中的 **有哪些状态管理选项可用？**

应用程序状态 - 一种在内存中存储状态信息的 **ASP.NET** 定义的机制，适用于 **web** 应用程序的所有用户和会话。请参阅 **ASP.NET 应用程序状态概述**，网址为 <http://msdn2.microsoft.com/en-us/library/ms178594.aspx>。另请参阅第 115 页中的 **处理初始请求**。

用户状态 - MapXtreme Session 的已保存状态和用户多次请求之间的应用程序状态。用户对 MapXtreme Session 或应用程序进行的任何更改都必须保存到 HTTP 会话中。更改可能如让地图重新居中一般简单，也可能如创建主题地图的查询一般复杂。

起始状态 - 当用户访问对象池 MapXtreme Session 实例时，该会话实例的状态。有四种可能的起始状态：

- 对于到站点的新用户，来自对象池的可用 MapXtreme Session 实例是干净的，背景地图处于初始状态。
- 对于到站点的新用户，来自对象池的可用 MapXtreme Session 实例是不干净的（地图已由其他用户更改）
- 对于返回的用户，来自对象池的可用的 MapXtreme Session 实例是干净的。
- 对于返回的用户，来自对象池的可用 MapXtreme Session 实例是不干净的（地图已由此用户或其他用户更改）

MapInfo.Engine.Session.State - MapXtreme 定义的机制，可确定将要自动保存 MapXtreme Session 状态还是手动保存。自动会话状态意味着整个 MapXtreme Session 都会保存到 HTTP 会话中。自动会话状态在应用程序的 Web.config 文件中的键中设计：

```
<add key="MapInfo.Engine.Session.State" value="HttpSessionState" />
```

手动会话状态表示开发人员负责保存随用户而更改的状态。这是一种在建立可伸缩的对象池应用程序时使用的机制。

手动会话状态在应用程序的 Web.config 文件中的键中设计：

```
<add key="MapInfo.Engine.Session.State" value="Manual" />
```

StateManager - 一个具有方法和属性的 MapXtreme 类，可帮助在对象池 web 应用程序中保存和恢复用户状态。当 MapInfo.Engine.Session.State 在 Web.config 文件中设置为手动时，应用程序必须提供实现 SaveState 和 RestoreState 方法的 StateManager 类。请参阅第 113 页中的 实现 StateManager。

sessionState - 标准 ASP.NET Web.config 元素，用于配置使用哪种存储机制来保存用户状态。三种类型：

- InProc - 用户状态在 ASP.NET Session 的生命周期存储在内存中。
- StateServer - 用户状态保存在服务器上，以便今后检索用户更改。
- SQLServer - 用户状态保存到 SQL 服务器数据库中，以便今后访问。

什么是状态管理

多类型的 web 应用程序都需要执行状态管理 — 保存和恢复每个用户会话所处状态或情况的有关信息的处理过程。例如，当某个零售 web 应用程序提供购物车时，应用程序必须记住每个用户购物车的状态。

在 web 地图绘制应用程序中，如果您的应用程序允许用户单击地图进行放大，则应用程序需要记住某个用户可能在放大欧洲部分，而另一个用户在放大澳大利亚。记忆每个用户地图状态的代码被称为状态管理代码。

有哪些状态管理选项可用？

MapXtreme 向您提供如何管理状态的以下选项：

- 可以使用 **InProc** 状态管理，它易于编码；但是，它并非对所有应用程序都适用，因为它会加重服务器资源的负担。
- 可以使用带自动状态管理的对象池体系结构。此模型易于编码，但是生成的应用程序可能不够快，这视您的要求而定。
- 可以使用带手动状态管理的对象池体系结构。此模型需要您参与更多编码工作。您必须编写代码以保存和恢复包含用户状态的适当的 **MapXtreme Session** 对象。但是，此模型会生成最具可伸缩性的应用程序，因此它是拥有大量用户的应用程序的最佳选择。

每个选择都具有各自的利弊。在使用 **MapXtreme** 设计 web 应用程序时必须考虑许多因素。有些因素是 **MapXtreme** 设计决定，其它因素是 Microsoft 技术设计模式。本章将让您了解一些必须做出的决定，并展示如何做出符合应用程序类型的正确选择。

强烈建议您在撰写 web 应用程序前考虑状态管理。您选择的状态管理类型将影响撰写应用程序的方式。在撰写 web 应用程序时，如果之后决定采用不同类型的状态管理，则您可能需要重写应用程序的很大一部分。

撰写应用程序前应提出的问题

如果要创建简单的 **MapXtreme Web** 应用程序，您可以快捷轻松地进行创建。只需使用随 **MapXtreme** 提供的 Visual Studio 模板 ("MapXtreme Web Application") 即可创建全新的 Web 应用程序。

但是，对于较为复杂的 web 应用程序，在投入大量时间和精力为 web 应用程序进行业务逻辑编码前，您需要考虑若干问题，因为这些问题的答案将帮助您决定哪种类型的状态管理适合您。

有多少用户将访问您的应用程序？

需要回答的最重要的问题是：有多少人会访问这个站点？

越早确定自己的 web 应用程序将拥有的用户数量，就对您越有利。如果开发和测试用户数量极少的某个应用程序（例如作为试验性项目），稍后您可能会发现自己的应用程序在拥有很多用户时运行不佳。在这种情况下，您可能会发现自己需要更改应用程序的体系架构 — 这一更改可能会要求重写自己的应用程序。最好能够提前规划并事先预测用户负载，以便能够一开始就使用恰当的状态管理体系结构。

用户数目已知

或许您在创建一个内联网网站点，拥有该站点有限、定义好的潜在用户列表。例如，您可能会创建一个公司内特定部门使用的 web 站点。您可能知道访问站点的每个人的姓名。

在并发用户定义明确且数量有限的情况下，您可以考虑使用 **InProc** 非对象池开发模型。它是编码的最简易模型，但并非所有应用程序的恰当选择，因为该模型对 **web** 服务器资源有很多要求。它会对每个并发用户创建 **MapXtreme Session** 实例。但是，如果您的用户池是有限的且定义明确，则这些服务器要求可能不是问题。

用户数目未知

其它开发模型是当您不知道要访问站点的用户的数目时使用的模型。或许您希望或期待 **web** 站点随时间推移会吸引越来越多的访问者。在此情况下，使用 **InProc** 非对象池模型是不合适的，您应当规划使用手动捕获用户状态的对象池开发模型。在对象池开发模型中，您将创建有限数量的 **MapXtreme Session**，将通过重新使用池中的某个对象为每个用户请求提供服务。

对象池体系架构用于开发具有可伸缩性的应用程序。以后可能需要添加更多的服务器来处理附加负荷，因此应用程序必须了解如何保存用户当前状态，以及当下一个请求到达服务器时，何时及如何应用。这可能会发生在服务器群内相同或不同的服务器上。这种可伸缩性导致对系统如何存储状态、访问数据和响应多个请求而需要作出的许多选项和选择。因此，规划应用程序并创建一个强健的、支持分布式应用程序的体系结构是非常重要的。

初始地图看起来应该是什么样子？

开发应用程序时，必须判断哪些信息对于所有用户都是相同的。这包括地图图层、标注、标题，以及颜色编码或其他类型的主题图影线。它们称为背景地图。

要设置背景地图，请运行 **MapXtreme Workspace** 管理器（随 **MapXtreme** 安装的一种桌面应用程序），并将您的地图保存为工作空间文件（.mws 文件）。请参阅[第 24 章：Workspace 管理器](#)。

创建 .mws 文件后，编辑 **web** 应用程序中的 **Web.config** 文件，将参考包括到 .mws 文件中。以下 **Web.config** 摘录显示了相关语法：

```
<configuration>
  <appSettings>
    <add key="MapInfo.Engine.Session.Workspace"
      value="C:\MIDATA\EvalData\WorldDetail\World_Detail.mws" />
  </appSettings>
</configuration>
```

此标记指示 **MapXtreme Session** 只要创建 **Session** 新实例即加载此工作空间。

会允许用户以何种方式修改地图？

大多数 **web** 应用程序允许用户点击、拖动或执行其他操作来以某种方式修改地图。在应用程序中可以考虑以下方面。

是否允许用户执行以下操作：

- 放大、缩小或者将地图重新居中？
- 选择地图的某些功能（可能是通过直接单击地图）？
- 打开或关闭地图元素（例如清除复选框以关闭街道显示）？
- 创建和/或修改主题图影线（例如在地图上使用彩色编码显示数据）？

- 在地图上放置地图注释（例如符号标记）？

地图绘制应用程序中可支持所有这些操作。但是，如果您决定使用对象池模型来实现 web 应用程序，就应当注意对象池应用程序需要包括小心保存和恢复用户对地图进行的所有更改的代码。

例如，如果您的 web 应用程序允许用户在地图上放置注释，则需要编写代码来保存每位用户的自定义注释，然后编写代码来恢复带有各后续请求的自定义注释。

如果应用程序使用带手动状态管理的对象池模型，添加越来越多的功能以允许用户修改地图时，您的状态管理代码也会变得越来越复杂。当考虑允许用户以各种方式修改地图的功能时，也应记住留出时间对您的状态管理代码进行开发和测试。

InProc 开发模型

如果使用 InProc 开发模型，则为应用程序编码的方式颇为类似于给桌面应用程序编码的方式。使用 InProc 模型，每位用户都会有一个 MapXtreme Session 对象；这意味着每位用户都拥有自己的进程空间耗用，且资源不能共享。（MapXtreme Session 是 Catalog、MapFactory、CoordSysFactory 和其它 MapXtreme 对象的容器。）

InProc 模型是用于开发目的的简单模型。由于每位用户都有自己的 MapXtreme Session，可以在会话内进行更改，而无需干扰任何其他用户的会话。

例如，如果用户在地图上单击以进行放大，您的应用程序代码可以仅仅修改地图的对象缩放属性，而无需担心缩放级别中的更改是否会对其他用户造成不好的影响。因此对 InProc web 应用程序进行编码很容易。

将 web 应用程序想像为一家餐厅会有助于理解。Web 应用程序需要为多个用户同时提出的请求提供服务，正如餐厅需要侍者从多张餐桌接受客户点菜。

InProc 模型类似于为每张餐桌雇用一名侍者的餐厅。餐厅为每张餐桌雇用一名侍者的做法成本过高。但是，如果餐厅有能力为每张餐桌雇用一名侍者，那么这样做肯定能够为每张餐桌提供极佳的服务。此外，每张餐桌使用一名侍者的做法会让每位侍者的工作更轻松。由于每位侍者只为一张餐桌提供服务，侍者就无需浪费任何时间或精力来记住点菜单所属的餐桌。

InProc 开发模型的利与弊

如果选择 InProc 模型，编写 web 应用程序将更为轻松，因为您无需提供提供任何复杂的代码来保存和恢复带有各自后续请求的每个用户的状态。

但是，InProc 模型是不可伸缩的，因为它需要为每个并发用户配备专用的 MapXtreme Session。对于同时使用站点的数千用户，应用程序将需要维护服务器上数千个 MapXtreme Session，这会加重服务器的资源负担。如果预计会有大量并行用户，那么 InProc 模型并非应用程序的恰当选择。

InProc 管理：简介

以下示例说明了当应用程序使用 InProc 模型时如何创建和使用 MapXtreme Session。下文以简化的术语描述了事件发生顺序：

1. 用户启动浏览器并前往地图绘制应用程序 web 站点。
2. 服务器上创建了一个新的 MapXtreme Session。它将用于在此 ASP.NET 会话中为来自于此用户的 所有请求提供服务。
3. 将加载您的 Web.config 文件中指定的工作空间文件。结果是，MapXtreme Session 的 MapFactory 中有一个或多个地图对象。对于本示例，让我们假设工作空间包含一张地图，该地图最初显示整个世界。
4. 将对 web 页面的 HTML 进行渲染，并返回到客户端浏览器。页面的一部分为 HTML img 标记， 它会从服务器请求地图图像。在服务器上，将使用来自此用户 MapXtreme Session 的对象（尤 其是地图对象）来渲染地图图像，然后将其返回到用户的浏览器上。
5. 用户选择放大工具，围绕澳大利亚绘制了一个矩形框，以放大澳大利亚。
6. 在服务器上，应用程序会修改地图对象的缩放属性，将其放大为显示澳大利亚。应用程序会渲染 一张新的地图图像，然后将图像以数据流方式发送到客户端浏览器。
地图对象现在处于与其初始状态不同的状态下，其中心和缩放属性均已更改（以显示澳大利亚而 不是整个世界）。
7. 用户单击选择工具，然后单击地图上显示为澳大利亚的部分。
8. 应用程序在服务器上执行以下选择：选择澳大利亚。渲染一张新的地图图像以显示新选择。
现在，有两种方法可以从原始状态更改地图的状态：更改中心和缩放级别（在步骤 6 中），且已 出现选择。
9. 另一用户在其它地方启动了浏览器并前往您的地图绘制应用程序 web 站点。服务器上为第二个用 户创建新的 MapXtreme Session。该会话包含一张显示整个世界的地图。此地图对象与第一个用 户的地图是相互独立的。

注意，第二个用户看到的是世界地图，而不是放大为澳大利亚的地图。只有第一个用户的地图放大 到了澳大利亚。创建了第二个 MapXtreme Session（使用其自身的地图）来为第二个用户的请求提 供服务。因此第二个用户不会看到与第一个用户相同的地图。

配置应用程序使用 InProc 开发模型

Web 应用程序的 Web.config 文件通过设置来控制应用程序模型。对于 InProc 模型，Web.config 文 件设置为：

```
<!--Use this setting to turn Session pooling on/off (true/false)-->   
<add key="MapInfo.Engine.Session.Pooled" value="false" />  
  
<!--Use this setting to save Session state automatically  
(HttpSessionState) or manually (Manual)-->   
<add key="MapInfo.Engine.Session.State" value="HttpSessionState" />
```

```
<sessionState mode="InProc" stateConnectionString="tcpip=127.0.0.1:42424"  
sqlConnectionString="data source=127.0.0.1;userid=sa;password="  
cookieless="false" timeout="20" />
```

这些设置指定 `MapInfo.Engine.Session` 对象不集中到对象池，且 `MapXtreme Session` 对象的状态会自动存储在用户 HTTP 会话（键值 = `HttpSessionState`）中。

在 `sessionState` 元素中，我们已将模式属性设置为 `InProc`，以指定将所有内容保存在进程中。如果未指定 `sessionState`，则 `InProc` 为默认模式。

`MapXtreme Session` 实例的生命周期取决于会话状态配置。如果使用基于服务器的会话状态且 `Web.config` 文件中的 `ASP.NET sessionState` 元素设置为 `InProc`，则在 `ASP.NET` 会话的生命周期内 `MapXtreme Session` 实例在内存中高速缓存。

如果使用任何其它类型的会话状态配置（例如，基于客户端或基于服务器的 `sessionState` 元素设置为 `StateServer` 或 `SQLServer`），则为每个 `ASP.NET` 请求创建和处置 `MapXtreme Session` 实例。但是，这会带来性能方面的影响，应避免用作开发模型。如果采用 `InProc` 模型，则 `sessionState` 应设置为 `InProc`。

一旦决定要采用 `InProc` 会话的此选项，将为访问 `web` 站点的每个用户提供一个 `MapXtreme Session` 对象的副本及其所包含的所有内容。很显然，如果用户的数目增加，则内存占用区也会增大。

应用程序还将在每次创建 `MapXtreme Session` 时加载 `web.config` 中定义的工作空间。对于 `InProc` 模型，这意味着创建了新的 `MapXtreme Session`，并且当用户首次访问该站点时加载工作空间。

使用支持 `InProc` 开发模型的 `MapXtreme` 模板

如果 `Web` 站点是使用“目标”创建的，则请注意其不遵循模型。

通过使用 `InProc` 的模板来创建应用程序：

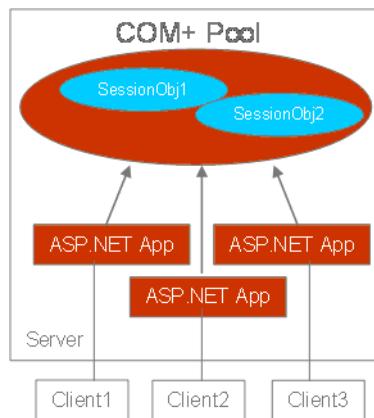
1. 修改 `Web.config` 文件，如[第 105 页中的 配置应用程序使用 InProc 开发模型](#)中所示。
2. 从 `MapForm1.aspx.vb` 或 `MapForm1.aspx.cs` 删除应用程序的 `Page_Load` 和 `Page_Unload` 方法，或注释掉由 `MapXtreme` 模板自动放置的手动状态管理代码。

对象池的状态管理

应用程序池中的状态管理是设计用于让应用程序可伸缩的一种开发模型。如果预计自己的应用程序将有较大和/或持续增长的用户数量，则对象池模型是比 `InProc` 模型更好。

什么是对象池？

在对象池模型中，您的应用程序会在服务器上创建有限数量的 `MapXtreme Session` 对象。随后会在应用程序中共享并重新使用这其中的每一个对象实例；每个 `MapXtreme Session` 都会处理来自多个用户的请求，如下图所示。



此图代表带有两个 **MapXtreme Session** 对象的 **COM+** 池的应用程序，它们由三个用户（**web** 客户端）访问。无论何时客户端提交请求，都会有两个 **MapXtreme Session** 对象的其中一个提供服务。

在此模型中，客户端没有专用的 **MapXtreme Session**。而是对 **MapXtreme Session** 对象进行共享和重用。可能产生以下事件序列：

1. **Client1** 可能请求地图放大显示非洲。此请求可能会使用 **SessionObj1** 进行处理。
2. **Client3** 可能请求地图放大显示澳大利亚。此请求也可能会使用 **SessionObj1** 进行处理。换句话说，每个会话都处理来自多个客户端的请求。没有一个客户端“拥有”**MapXtreme Session**，它们都是共享的。
3. **Client1** 可提交一个请求来平移地图以显示欧洲。此请求可能会使用 **SessionObj2** 进行处理。换句话说，不能保证特定客户端提交的每项请求都会由同一个 **MapXtreme Session** 或相同的地图对象提供服务。

对象池的利与弊

对象池帮助减少并发 **MapXtreme Session** 实例的数目，从而减少从零创建每个对象的耗用。需要对象时就从对象池中取出。如果不再需要对象则将对象放回对象池，等待下一个请求。**MapXtreme** 中的对象池意味着已经可以使用背景地图来创建和加载多个 **MapXtreme Session** 对象，且这些对象在收到 **web** 请求时都可用。

对象池帮助您用以下方法优化服务器的资源：

- 通过并行运行多个 **ASP.NET** 请求改进 **web** 应用程序的整个响应时间。
- 通过减少并发 **MapXtreme Session** 实例的数量来节约资源。
- 通过减少同时运行的请求（减少线程上下文切换）数量来最大化 **CPU** 的利用率。请求要占用大量 **CPU** 时（例如地图图像导出）这就显得尤其有用。通常推荐的池大小为每个 **CPU** 1-2 个会话实例。

但是，对象池应用程序可能会使应用程序更为复杂。将共享并重用 **MapXtreme Session**（及其地图）；因此，您需要采用一些步骤，以确保应用程序每次处理某个客户端请求时每张地图都会重设为适当的状态。

例如，假定客户请求德国地图。因为无法保证特定地图会放大到德国，所以不能简单地从池中获取 **MapXtreme Session** 然后使用其地图来渲染图像。服务于请求的地图必须设置为恰当的、已知的状态。作为应用程序开发人员，在使用对象以服务于当前请求前，必须管理对象的状态，以确保对象处于适当的状态。

保存对象池应用程序的状态

对象池开发模型向您提供管理对象池状态的两个选项。您可以使用自动状态管理，让 **MapXtreme** 管理当前用户状态，或者可以使用手动状态管理，控制必须为每个用户保存和恢复的更改。

自动状态管理

此情况意味着 **MapXtreme Session** 对象的状态将自动保存到 `web.config` (`StateServer` 或 `SQLServer`) 中定义的会话机制。应用程序的程序员不必用编程方式保存当前用户的任何状态信息，例如图层、主题图、打开表或当前缩放。将保存 **MapXtreme Session** 对象的整个状态。

与 `InProc` 模型相似，此选项保存系统资源，因为并不是每个用户都有一个 **MapXtreme Session**，而所有用户将共享对象池 **MapXtreme Session** 对象。缺陷是整个 **MapXtreme Session** 都将序列化为当前定义的状态机制。通常情况下，**MapXtreme Session** 对象在工厂中检索所有的对象，并将它们序列化为状态机制。这包括所有的地图和它们所包括的图层，所有打开的表定义和任何加载的投影。如果应用程序是使用自动状态管理，则 **MapXtreme Session** 对象不了解单个用户的状态，而是存储所有可用的信息。这包括在不同用户之间不会更改的图层，即背景地图。这会是一个花费时间的过程，可能会造成此类应用程序比 `InProc` 模型执行的要慢。有关序列化的信息请参阅[第 191 页第 9 章中的 串行化和永久化](#)。

手动状态管理

此配置意味着 **MapXtreme Session** 实例不会自动保存任何实例数据。程序员应在应用程序中写入相应代码，以保存和恢复每位用户的状态。**MapXtreme Session** 对象不会将其任何状态存储为当前定义的状态机制。

此配置提供性能和可伸缩性的最好效果。在此模型中，您使用的是池对象，**Session** 不保存任何状态，而池对象返回到“不干净”的 `COM+` 池。通过这些可以认识到池对象对当前用户执行的地图作出了修改，需要您的应用程序清除该状态，将其正确设置以备下一个用户使用。可以选择在 `web` 应用程序请求开始时清除，或在结束时清除。[第 113 页中的 实现 StateManager](#)。

使用对象池 **MapXtreme Session** 并手动管理用户状态，即可完成这一复杂的开发模型。您无从得知哪个请求会使用 **MapXtreme Session**；但同时，您只需使用特定于该用户的 **MapXtreme Session** 来满足用户的请求。因此，应用程序必须了解对象池 **MapXtreme Session** 的起始状态。

应用程序必须处理四种可能的起始状态。每种情况都会涉及用户状态和 MapXtreme Session 状态。

用户为 ...	池中的 MapXtreme Session 为
站点新用户	干净 *
站点新用户	不干净 †
返回站点	干净 *
返回站点	不干净 †

* MapXtreme Session 及其背景地图均处于初始状态。

† MapXtreme Session 和 / 或其背景地图包含其他用户的更改。

手动状态管理：简介

以下示例说明了当应用程序使用手动状态管理模型时如何管理对象池 MapXtreme Session。下文以简化的术语描述了事件的可能发生顺序：

1. 用户启动浏览器并前往地图绘制应用程序 web 站点。
2. 由于此应用程序使用对象池，将从对象池检索 MapXtreme Session 实例。
3. `RestoreState` 方法是从 `Page_Load` 调用的。
如果 MapXtreme Session “不干净” 并带有一些用户的更改，则应用程序会将地图设置回所需的起始状态；请参阅 `AppStateManager` 类中的 `RestoreDefaultState` 方法。
对于本示例，让我们假设起始状态下会显示整个世界。
4. 将对 web 页面的 HTML 进行渲染，并返回到客户端浏览器。页面的一部分为 HTML `img` 标记，它会从服务器请求地图图像。在服务器上，将使用来自此用户浏览器会话的对象（尤其是地图对象）来渲染地图图像，然后将其返回到用户的浏览器上。
5. 在 `Page_Unload` 方法中，应用程序通过调用 `AppStateManager.SaveState` 方法保存用户的状态。
6. 用户选择了放大工具，并围绕澳大利亚绘制矩形框。构建到放大工具中的 `Javascript` 向服务器提交请求，放大澳大利亚。
7. 在 MapXtreme 服务器上，应用程序必须再次从对象池中获取 MapXtreme Session 对象。请注意，对象池中返回的 MapXtreme Session 可能与服务于先前请求对象不是同一对象。此对象可以为干净或者不干净。
8. 应用程序在服务器上会调用 `AppStateManager.RestoreState` 方法，以恢复步骤 4 中保存的用户状态。这会将 MapXtreme Session 恢复为对此用户适合的已知状态。
9. 应用程序会修改地图对象的缩放属性，将其放大为显示澳大利亚。应用程序会渲染一张新的地图图像，然后将图像以数据流方式发送到客户端浏览器。同样，应用程序会在请求最后保存用户状态。
10. 用户单击选择工具，然后单击地图上的澳大利亚区域以选择澳大利亚。

11. 应用程序会再次从对象池获取 **MapXtreme Session**，然后再次调用 **RestoreState** 以便将池对象设置为已知状态。

12. 服务器上的应用程序会执行操作以选择澳大利亚。渲染一张新的地图图像以显示新选择。

13. 服务器上的应用程序会再次调用 **AppStateManager.SaveState** 以保存用户状态。由于本例允许用户进行选择，**AppStateManager** 的 **SaveState** 和 **RestoreState** 方法将需要写入相应代码以保存和恢复选择。

允许用户更改的 **MapXtreme session** 的任何一方面，例如图层、主题、查询、地图视图，都必须在代码中处理以保存和恢复每个项目。

手动使用对象池并管理用户状态的好处体现在它能提高效率，并允许应用程序伸缩。但是，就在每个请求结束时保存用户状态然后在用户的下一次请求开始时恢复状态这一方面，此模型还需要进行额外工作。

配置对象池应用程序使用手动状态管理

有关使用手动状态管理的对象池应用程序的示例，请参阅 **MapXtreme** 随附的 web 应用程序样本。以下是配置此类应用程序的要点。

使用以下设置配置 **Web.Config** 文件中的池对象：

```
<!--Use this setting to turn Session pooling on/off (true/false)-->
<add key="MapInfo.Engine.Session.Pooled" value="true" />
<!--Use this setting to save Session state automatically
(HttpSessionState) or manually (Manual)-->
<add key="MapInfo.Engine.Session.State" value="Manual" />
<sessionState mode="StateServer" stateConnectionString=
"tcpip=127.0.0.1:42424" sqlConnectionString="data source=127.0.0.1;user
id=sa;password=" cookieless="false" timeout="20" />
```

这两种设置可以组合为以下的选项：

- 对于池对象和自动状态管理，请将对象池设为真，然后将状态设为 **HttpSessionState**。
- 对于池对象和手动状态管理，请将对象池设为真，然后将状态设为手动。

将确定用户当前状态的逻辑放入 **Page_Load** 方法。请参阅第 113 页中的 实现 **StateManager**。

用户状态

应用程序从池中检索时必须考虑用户的当前状态和 **MapXtreme Session** 对象的状态。在此种情况下您必须决定：

- 是否使用 **SaveState** 方法将池对象清除后返回到池中？
- 是否将池对象保持原样，然后使用 **RestoreState** 方法来恢复任何状态？

正确的选择不一定很明显，因为各个用户的状态都各有不同，可能导致会话对象未清除就放回到池中。其它应用程序可能会通过等待 **Page_Load** 清除或检查 **Session** 是否需要清除而节省时间。

第 109 页中的 手动状态管理：简介部分的示例位于第二个选项之后：清除池对象（在 **RestoreState** 方法中）。

下一部分提供了有关如何保存状态和恢复状态的详细信息。

手动状态管理详探

本部分提供有关对象池应用程序如何才能执行手动状态管理的详细示例。我们将检查来自主题图示例 web 应用程序的相关部分，它是作为 **MapXtreme** 的一部分安装的。如果您尚未这样做，则在阅读本部分前可能需要运行主题图示例以熟悉其内容。

此处，**Thematic** 示例是本解决方案中包含的项目之一：

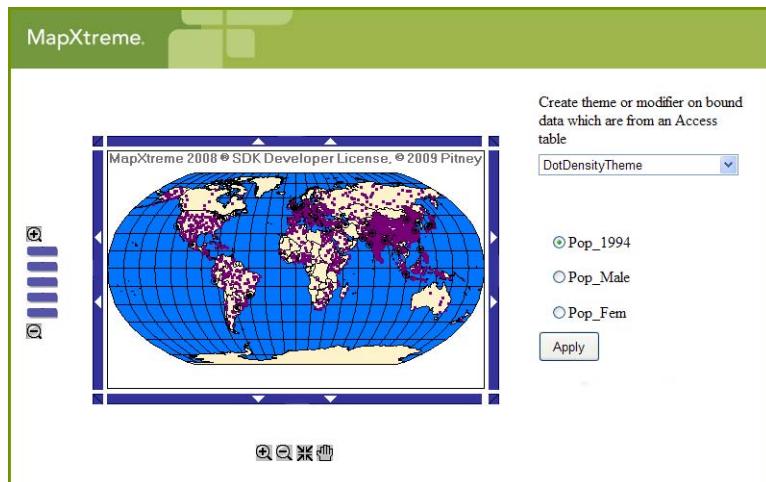
C:\Program Files\MapInfo\MapXtreme\7.x.x\Samples\VisualStudio20xx\Web\Features

本讨论包括以下主题：

- 主题图示例概述
- 应用程序设置
- 实现 **StateManager**
- 按照正常顺序对 **MapXtreme** 对象进行序列化
- 自动序列化 **MapXtreme** 对象
- 处理初始请求
- 处理后继请求

主题图示例概述

此应用程序显示从工作空间 World.mws（部分示例数据随 **MapXtreme** 安装）加载的世界地图。



web 表格上的选项让用户能够在地图上创建不同类型的主题图影线。这些主题会修改 world.tab 表中各个区域的外观（国境线）。主题的属性数据来自于 Microsoft Access 数据库 eworld.mdb（包括在 Visual Studio 项目中）。

在本示例中，当 MapXtreme Session 创建时已预装有 world.mws 工作空间文件；但是，当客户端访问应用程序时，随后会增加来自 eworld 的列。

-
- ① 要获得更高的性能，可将属性列信息放到工作空间文件中，这样就会在 MapInfo Session 创建时预装所有需要的数据。
-

应用程序设置

首先，查看 Web.config 文件的以下部分内容，该文件包括了相关的应用程序设置：在 Web.config 文件顶部附近可以找到这些设置：

```
<configuration>
<appSettings>
    <!--Use this setting to turn Session pooling on/off (true/false)-->
    <add key="MapInfo.Engine.Session.Pooled" value="true" />

    <!--Use this setting to save Session state automatically
    (HttpSessionState) or manually (Manual)-->
    <add key="MapInfo.Engine.Session.State" value="Manual" />

    <!--Use this setting to preload a workspace on Session creation-->
    <add key="MapInfo.Engine.Session.Workspace" value="c:\Program
    Files\MapInfo\MapXtreme\7.x.x\Samples\Data\World.mws" />
```

设置说明如下：

- **MapInfo.Engine.Session.Pooled:** 由于此键的值为 "true"，因此，MapXtreme Session 对象由 COM+ 服务提供对象池。当应新的客户端请求来检索新的 MapXtreme Session 时，Session 对于刚创建的新对象可能是“干净”的；或者如果已使用它来处理先前的请求，则它可能是“不干净”的对象。由于此对象具有不确定状态，在使用前此应用程序将在每个请求中按步骤来将 MapXtreme Session 回复到已知状态。
- **MapInfo.Engine.Session.State:** 由于此键的值为 "Manual"，应用程序负责显示保存和恢复 Session 状态。手动设置有助于提高性能，因为应用程序的StateManager 类是以智能方式编写，说明了它应保存和恢复哪些对象。StateManager 会选择性地保存和恢复所需的最小数量的对象；这就是为什么它是最有效的状态管理选项的原因。（请参阅第 113 页中的 实现StateManager）。
- **MapInfo.Engine.Session.Workspace:** 此设置让您能指定分号分隔的 .mws 工作空间文件列表，它将在 MapXtreme Session 创建时预装。要指定多个工作空间，请使用分号分隔文件名。

继续深入 Web.config 文件，您将找到以下设置：

```
<sessionState mode="StateServer"
stateConnectionString="tcpip=127.0.0.1:42424" sqlConnectionString="data
source=127.0.0.1;user id=sa;password=" cookieless="false" timeout="20" />
```

它会指示想要保存状态的方式；在这种情况下，会保存到 StateServer 中，可以在今后检索其中的信息。您还可以将它设置为 SQLServer，其状态将保存到数据库中。

实现 StateManager

主题图示例应用程序会使用手动状态管理，这意味着它需要实现 **StateManager** 类才能保存和恢复适当的 **MapXtreme Session** 更改。

该类从抽象基类 **MapInfo.WebControls.StateManager** 继承并且必须实现以下方法：

```
Public Overrides Sub RestoreState()  
Public Overrides Sub SaveState()
```



如果使用“模板”来在 Visual Studio 中创建新 Web 站点，所产生的项目包括对 **StateManager** 的实现，称为 **AppStateManager**。

AppStateManager 类的确切内容在每个应用程序之间都不同。应用程序提供让用户定制地图的选项越多，您就需要增加越多代码到 **AppStateManager** 中用于保存和恢复这些定制项。这就是为什么主题图示例应用程序随附的 **AppStateManager** 包含的代码比您从 **MapXtreme Web** 站点模板创建新项目时获得的 **AppStateManager** 包含代码更多的缘故。

例如，主题图示例会处理针对主题图层、主题表、属性表和组图层进行的定制。而与此相对，**web** 模板则有代码来处理对图层和选择的定制。

RestoreState 和 **SaveState** 方法的使用如下所示：

1. 客户端每提交一次请求，**Page_Load** 方法就会调用 **RestoreState**。每次在客户端使用地图工具时，也会调用 **RestoreState**。**RestoreState** 方法可确保 **MapXtreme Session** 对象（从处于未知的、可能“不干净”的状态的对象池中检索得到）会恢复为已知状态，可恢复到用户的状态（如果存在），或者恢复到应用程序的默认状态。

例如，如果用户上次请求地图时用户地图在法国部分放大，那么 **RestoreState** 方法将确保从对象池中检索到的地图会返回到“放大显示法国”状态。这样，用户的会话就可以从上次离开位置继续。

以下 VB 代码示例取自主题图示例应用程序 **WebForm1.aspx.vb**。

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As _  
    System.EventArgs) Handles MyBase.Load  
    ' The first time in  
    If Session.IsNewSession Then  
  
        '*****  
        ' You need to follow below lines in your own application.  
        '*****  
        Dim stateManager As New AppStateManager  
  
        ' tell the state manager which map alias you want to use.  
        ' You could also add your own key/value pairs, the value should be  
        ' serializable.  
  
        stateManager.ParamsDictionary.Item(AppStateManager.ActiveMapAliasKey)_  
            = Me.MapControl1.MapAlias  
        ' Put state manager into HttpSession, so we could get it later on from  
        ' different class and requests.
```

```
MapInfo.WebControls.StateManager.PutStateManagerInSession_
    (stateManager)
Me.InitState()
End If

MapInfo.WebControls.StateManager.GetStateManagerFromSession.RestoreState(
)

    PrepareData()
End Sub
```

2. 执行 `Page_Load` 方法后，应用程序会应用任何恰当的业务逻辑，根据客户端请求的性质而定。例如，如果是通过用户单击放大工具来产生请求，则业务逻辑会调整地图的中心和缩放级别。
3. 在请求处理循环结束时，`Page_Unload` 方法会调用 `SaveState`。在客户端上使用地图工具后，也会调用 `SaveState`。
`SaveState` 方法会保存地图最新状态，这样当用户提交另一项请求时，请求就能够再次调用 `RestoreState`。对 `SaveState` 的每次调用都是在预期在下次收到请求时，对 `RestoreState` 可能进行后继调用的情况下执行。

```
Private Sub Page_UnLoad(ByVal sender As Object, ByVal e As _
EventArgs) Handles MyBase.Unload

    MapInfo.WebControls.StateManager.GetStateManagerFromSession()_
        .SaveState()
End Sub
```

按照正常顺序对 **MapXtreme** 对象进行序列化

在主题图示例随附的 `AppStateManager` 类中，`SaveState` 方法会调用 `ManualSerializer.SaveMapXtremeObjectIntoHttpSession` 方法数次。对象按以下顺序保存（序列化）：

```
ManualSerializer.SaveMapXtremeObjectIntoHttpSession(MapInfo.Engine.Session.Current.Catalog(SampleConstants.EWorldAlias), "mdb_table")

ManualSerializer.SaveMapXtremeObjectIntoHttpSession(MapInfo.Engine.Session.Current.Catalog(SampleConstants.ThemeTableAlias), "theme_table")

ManualSerializer.SaveMapXtremeObjectIntoHttpSession(map.Layers(SampleConstants.ThemeLayerAlias), "theme_layer")

ManualSerializer.SaveMapXtremeObjectIntoHttpSession(map.Layers(SampleConstants.GroupLayerAlias), "group_layer")
```

因为某些对象取决于其它对象，所以按正确顺序调用 `SaveMapXtremeObjectIntoHttpSession` 非常重要。

在本例中，我们首先要保存表（在 Catalog 中引用）。接下来会保存图层（在地图的图层集合中引用）。这是正确的顺序，因为大多数图层类型取决于表。

例如，每个 FeatureLayer 都与表关联。必须打开表才能实例化（或反序列化）使用该表的 FeatureLayer。因此，此 AppStateManager 类会首先保存表，然后保存图层。

同样，在 RestoreState 方法中，会先存储数据对象，随后调用以恢复图层：

```
ManualSerializer.RestoreMapXtremeObjectFromHttpSession("mdb_table")
ManualSerializer.RestoreMapXtremeObjectFromHttpSession("theme_table")
ManualSerializer.RestoreMapXtremeObjectFromHttpSession("theme_layer")
ManualSerializer.RestoreMapXtremeObjectFromHttpSession("group_layer")
```

自动序列化 MapXtreme 对象

在 SaveState 方法中，我们会标识要使用特定引用保存的对象，例如地图中的特定图层。

```
ManualSerializer.SaveMapXtremeObjectIntoHttpSession(map.Layers(SampleConstants.ThemeLayerAlias), "theme_layer")
```

但是，RestoreState 方法并不会引用 Catalog 或 map.Layers 集合。在 RestoreState 方法中，我们在 ManualSerializer 类上调用方法，但是我们不会对该调用的结果进行任何操作，并且没有对 map.Layers 集合的引用：它们会自动反序列化到 Catalog 或图层集合中的恰当位置，或在有需要的地方进行反序列化。

```
ManualSerializer.RestoreMapXtremeObjectFromHttpSession("theme_layer")
```



使用 ManualSerializer 类执行对象的保存和恢复。不要使用以下语法将 MapXtreme 对象直接存储到 HttpSessionState 中：

```
HttpContext.Current.Session.Item("myTable") = _
    MapInfo.Engine.Session.Current.Catalog("myTable")
```

处理初始请求

应用程序状态是服务器端的状态管理机制，用于存储对所有用户会话为公共或全局性的信息。应用程序状态在 Session.IsNewSession 为真时初始化。应用程序状态上处理初始请求时的重要元素。

对于 MapXtreme，应用程序会保持背景地图的初始状态，这样任何用户的初始请求都会在此默认状态下接收地图。

MapXtreme 主题图示例会演示处理初始请求所需的步骤。

应用程序必须处理两种不同类型的“首次”请求：1) 第一个用户访问 web 应用程序的首次请求，以及 2) 所有后继用户的首次请求。

任何用于第一次使用应用程序时，会从对象池检索到一个干净的 MapXtreme Session，其背景处于初始状态。初始状态存储在应用程序状态中，以模板的方式使用。该操作在 SaveDefaultState 方法中发生。

所有后继用户调用 `RestoreDefaultState` 的首次请求，它会检索来自应用程序状态的地图的初始状态。

以下为与初始请求相关的步骤。

1. 在 `Page_Load` 方法中，检查当前会话是否是新的。如果是新的，则应实例化 `MyStateManager` 并将 `MapAlias` 增加到状态管理器的 `ParamsDictionary` 属性。这是使用手动状态管理时所必须的。
2. 接下来在 `Page_Load` 方法中，由于是新的会话，我们会通过调用 `InitState` 方法来初始化应用程序数据。在 `InitState` 方法中，我们会准备地图的初始状态。
3. 接下来在 `Page_Load` 方法中，我们会调用 `RestoreState`，以检查这是否对此应用程序的首次请求。

如果这是对应用程序的首次请求，我们会调用 `SaveDefaultState` 方法来将地图上的各种属性设置为其初始状态，此外我们还会调用 `ManualSerializer` 方法，以便将此初始地图状态以字节数组的方式存储到 `HttpApplicationState` 中。

反之，如果这是用户的首次请求，但并非对应用程序的首次请求，我们会调用 `RestoreDefaultState`，将属性重置为其初始状态。

处理后继请求

在处理完用户的首次请求后，用户可能会提交其它请求，例如单击地图以放大。这时将调用 `RestoreState`（使用 `Page_Load` 方法或从地图工具中选择），将跳过首次请求逻辑。将执行恢复用户状态的代码。

在用户会话结束时，或在使用地图工具后，将调用 `SaveState` 方法来保存会话的状态。

i 如果使用 `Server.Transfer` 或 `endResponse` 标记为真的 `Response.Redirect`，则不会调用 `Page_UnLoad` 方法，原因是那些方法将会忽略该事件处理程序。在使用 `Server.Transfer` 或 `endResponse` 标记为真的 `Response.Redirect` 来解决该问题前，可以显式地调用 `MapInfo.WebControls.StateManager.GetStateManagerFromSession().SaveState()`。

关于 MapXtreme Session 的深入讨论

MapXtreme Session 是所有 MapXtreme 应用程序的起点。其管理 MapXtreme 应用程序所需资源的初始化。Session 还提供对其它重要对象的访问，如 `Data.Catalog`、`MapFactory`、`CoordSysFactory`、`Selections` 及其它。

要访问 MapXtreme Session，请调用 `MapInfo.Engine.Session.Current()` 方法。进程中的每个线程均有一个 MapXtreme Session 对象。每个线程只能有一个 MapXtreme Session 且该会话不可以在多个线程之间共享。

Web 应用程序可为每个用户提供一个 MapXtreme Session 对象，或以对象池的形式为多个用户提供多个 MapXtreme Session。以下部分更详细地描述了 COM+ 对象池选项。

配置 Microsoft COM+ 对象池

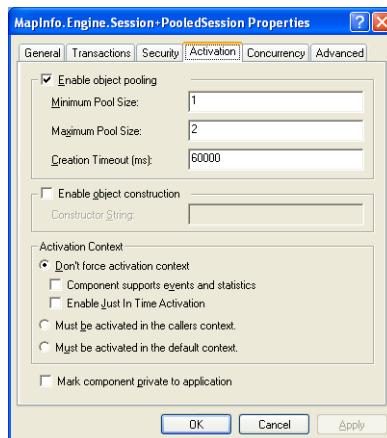
MapXtreme Session 对象用系统上的 COM+ 服务注册。此系统处理任何已注册池对象的配置和激活。运行时安装程序将会智能地注册和创建 MapInfo Session 对象默认设置。默认情况下，MapXtreme Session 对象用两个池对象配置，并且有 60 秒的超时。

配置池大小

MapXtreme Session 可以使用系统配置方法对适当部分进行配置。这些设置在“控制面板”的系统对话框中和应用程序的 .NET 配置文件中都有。Web 应用程序在使用对象池 MapXtreme Session 时必须管理这些设置。

一个重要的设置是 COM+ 系统所创建的池对象数目，用于服务运行的应用程序。此设置可使用控制面板 > 管理工具 > “组件服务”来访问。浏览组件服务 > 计算机 > 我的电脑 > COM+ 应用程序 > MapInfo.CoreEngine > 组件。在此处您将会找到 MapInfo.Engine Session 对象。右键单击该图标并选择“属性”。

“属性”对话框可用于设置各种不同的属性，包括每个应用程序的池对象数目和会话创建的激活超时。池对象数目的正确设置直接影响应用程序的性能。要正确设置池对象的数目，需要了解应用程序及它是如何访问数据的。



背景地图对性能的影响

会话可以直接影响应用程序性能的另一方面，是应用程序 Web.config 文件中定义的启动工作空间。此工作空间定义在创建会话对象时将加载哪些地图、图层和表。在 COM+ 池线程中及 web 应用程序空间之外会发生此种情况。

应用程序启动时，会向应用程序池请求一个 **MapXtreme Session** 对象。COM+ 池随后会创建在控制面板“组件服务”对话框中指定数目的池对象。该 **MapXtreme Session** 对象读取 Web.config 文件，在应用程序从池中得到实例之前加载指定的工作空间。因此，如果加载工作空间花费的时间比“组件服务”对话框中指定的超时时间还要长，将会接收到 COM+ 激活异常。因此非常重要的一点就是要了解工作空间中要加载的内容，以及完全加载需要多长时间。

如果指定“最小池大小”为 2，则 COM+ 池将为池对象在首次请求时创建 **MapXtreme Session** 的两个副本。当 COM+ 池发现超时等待池对象时，它将创建更多对象（最多达到最大设置）。如果工作空间非常复杂，需要花费大量时间来加载，则在应用程序激活状态下，后续对象的激活可能会造成超时错误。Web 应用程序池可以通过设置来控制对象池应用程序的再次激活，包括回收对象池类的设置。在某个时刻，可能会回收应用程序和池对象，从而加载其他工作空间。

必须判断哪些地图、表和图层是重要的，要作为应用程序的背景，并通过 Web.config 文件的工作空间加载。不过大的工作空间需要时间来加载，需要注意其产生的影响。要测试加载时间，只需简单地加载服务器上 **Workspace Manager** 中的工作空间，来确定实际的加载时间和调节相应的超时设置。

初始工作空间在加载不随用户而更改的地图、图层和表时非常有用，但是必须注意加载复杂工作空间的影响。

详细阅读

MapXtreme Session 的可配置性很高，设计为可以与 Microsoft COM+ 和 web 技术一起使用。要了解如何最好地使用此会话，需要了解 Microsoft web 传输体系结构。有多种途径可以获得更多有关上述主题的信息。以下是一些有关 web 体系结构信息的链接：

COM+ 对象池 (COM+ pooling)	COM+ 对象池概念
IIS 7.0	IIS 7.0：在 IIS 7.0 中管理 Web 应用程序。
IIS 8.0	Web 服务 (IIS) 概述

桌面应用程序、控件、对话框和工具

本章提供规划桌面应用程序方面的信息，还会概述 **MapXtreme** 中众多的桌面控件、对话框和工具。

在本章中：

◆ 规划桌面应用程序.....	120
◆ MapInfo.Windows.Controls 命名空间	121
◆ 要在桌面应用程序中使用的关键控件	122
◆ MapInfo.Windows.Dialogs 命名空间	128
◆ 定制控件和对话框.....	133
◆ MapInfo.Tools 命名空间概述.....	134
◆ MapXtreme 桌面工具 API	135
◆ 定制工具.....	139
◆ 工具事件.....	140
◆ 使用 Select 工具编辑 FeatureGeometry	141

规划桌面应用程序

MapXtreme 是一种基于 .NET 的对象模型，向任何基于 .NET 的开发工作提供地图功能。其中包括许多使用 .NET 框架开发应用程序的新兴技术。

用户通过桌面应用程序可与很多工具进行交互。虽然 Web 应用程序变得越来越复杂，但旨在用于桌面的应用程序仍有优势。例如，桌面应用程序有“对齐节点”功能，可用于将点图元准确地放在十字路口上。悬停鼠标指针时桌面应用程序还可以显示 InfoTips。

如果具有现有的 WinForm 或其他基于 .NET 的应用程序框架，则可以简单地将 MapXtreme 类集成到应用程序中。**MapXtreme** 对象模型是一个完全符合 .NET 的对象模型，因此您可以将 .NET 框架的功能与其它对象模型交互。以应用程序框架定义的模式简单地开发应用程序，并引入 **MapXtreme** 对象。

桌面应用程序的最佳实践

MapXtreme MaForm 应用程序模板旨在进行快速原型开发，因为仅可用其引用 **MapXtreme** 程序集。使用该模板作为快速演示或概念证明的起点。

WinForm 和 .NET 框架可为基于单文档界面 (SDI) 和多文档界面 (MDI) 的应用程序提供基本支持。WinForm 不提供可与 Microsoft 基础类 (MFC) 相似的应用程序框架。WinForm 没有文档视图、数据交换及其它基于 MFC 的用户界面的概念。

但是，Microsoft 确实提供了 Composite UI Application Block (CAB)，以此鼓励大家采用所推荐的模式和实践。请注意，.NET 2.0 Framework 和更高版本中才有 Composite UI Application Block。Microsoft 开发人员网站 (msdn2.microsoft.com/en-us/library/aa480450.aspx) 上有更多信息。

.NET 开发模型一个重要的方面是业务逻辑与表示的分离。**MapXtreme** 在其控件中使用此范例，使用户可以打开一个对话框修改对象的副本。以此方法，对话框不直接编辑运行时对象，可以在其它进程或机器中创建新的对象，并将序列化后的版本传送回去用于您的应用程序。

MapXtreme 和 COM

.NET 并不意味着 COM 的结束。许多开发厂家对 COM 对象模型做了很大的努力，投入很大不能够轻易放弃。那么开发具有 COM 对象的应用程序的最佳方法是什么呢？

最可能的情况是，新的 .NET 代码将必须与现有的 COM 代码可互操作（而不是相对立的情况），因此我们将集中讨论此种情况。.NET 客户端通过运行时可调用包装程序 (RCW) 访问 COM 服务器。RCW 将 COM 对象打包，并将其与 .NET 普通语言运行时 (CLR) 环境进行调节，使 COM 对象显示到 .NET 客户端，就象它是本地 .NET 对象，并使 .NET 客户端显示到 COM 对象，就象它是一个标准的 COM 客户端。

例如，您可以直接从 Visual Studio 将引用创建到这些 COM 对象，并简单地与它们进行交互，就象对 .NET 类一样。这样就可以在与旧 COM 对象进行通信时创建 .NET 应用程序，并将结果传递给 MapXtreme .NET 对象。将正在开发的 .NET 应用程序作为应用程序框架的协调段。这样就可以使用 .NET 框架中最新的技术，并保留在旧的程序库中现有的投入。因为 COM 对象中的所有数据已经转换到 .NET 环境，因此可以将这些数据传递到其它受管理的类，而不会产生域的问题。

示例应用程序和项目模板

MapXtreme 集成到 Visual Studio 中，并附带多种示例应用程序和项目模板。示例应用程序从不同方面展示如何在 .NET 框架和 Windows 窗体应用程序中使用对象模型。这些模板是应用程序开发的基本起点。

这些项目模板是 Visual Studio 的一部分，可以用于创建启动项目，并演示通用的应用程序框架。在 Visual Studio 中，通过选择 **MapForm** 应用程序模板创建新项目。这会创建一个基本的单文档界面 (SDI) 应用程序。有关基本的多文档界面 (MDI) 应用程序，以及如何将 MapXtreme 最紧密地集成到此模型中，请参见桌面应用程序样本。

.NET Windows 窗体开发平台是其方法和功能的基础。我们建议使用可以提供更完整的 UI 开发平台的第三方工具。

MapXtreme 附带的 **Desktop** 示例应用程序设计用于以最小开销显示特定任务。使用这些作为学习工具，而不是作为产品应用程序的起点，因为这些并不是设计为应用程序开发实例。

本章的其余部分将说明可用于构建桌面应用程序的各种控件、对话框和工具。

MapInfo.Windows.Controls 命名空间

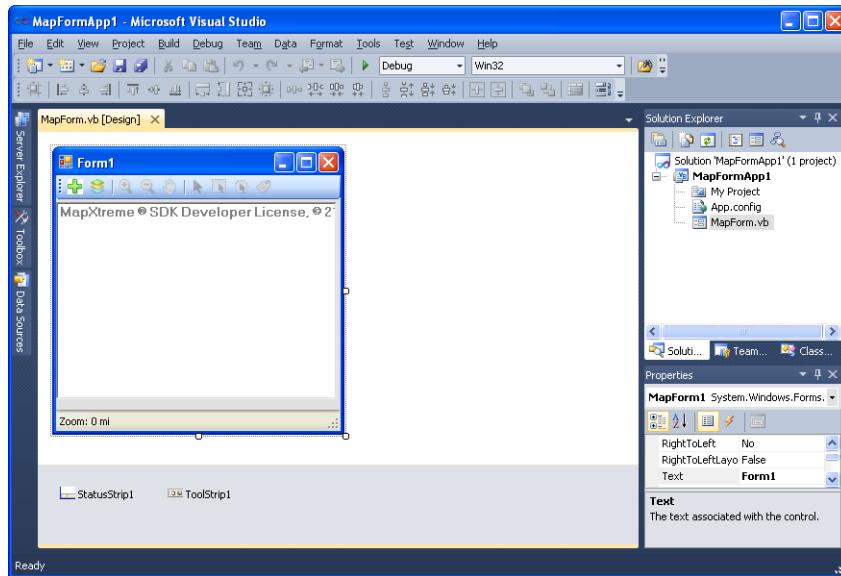
MapInfo.Windows.Controls 命名空间包括实现桌面应用程序中控件的类。使用 Windows 控件十分简单，可以将需要的控件放到窗体上并且可以使用各种属性对它们进行配置以满足您的规范。您将发现在此命名空间中的控件类似于 MapInfo.WebControls 命名空间中的控件。但是，每个命名空间中的控件不能彼此替换。此命名空间中的控件特定用在桌面应用程序中。有关基于 Web 的应用程序的控件的信息，请参阅第 5 章：**Web 应用程序、控件和工具**。



某些控件不能在设计阶段显示。将它们拖放到窗体时，这种控件仅显示为矩形。这些控件在运行时会正确显示。

MapXtreme 包括的桌面控件可分为两类：标准控件和地图控件。标准控件包括：按钮、视图工具、标注工具和各种各样的方框等。它们在很多情况下与 System.Windows.Forms 命名空间中的类十分类似，或者它们就是从 System.Windows.Forms 命名空间中的类继承而来的。

地图控件特定于 Pitney Bowes Software 的地图绘制实现。其中包括设置或修改对象样式、标注、图层、坐标系、主题以及其它相关功能方面的控件。



Visual Studio 工具箱和设计器，其中显示的是一些可用的 *MapXtreme* 控件。

最复杂的控件是“图层控件”，可通过模式对话框或直接显示在窗体中的控件使用该控件。“图层控件”有一个可以进行大量自定义的 API；有关示例，请参见示例应用程序“LayerControlDemo”。

要在桌面应用程序中使用的关键控件

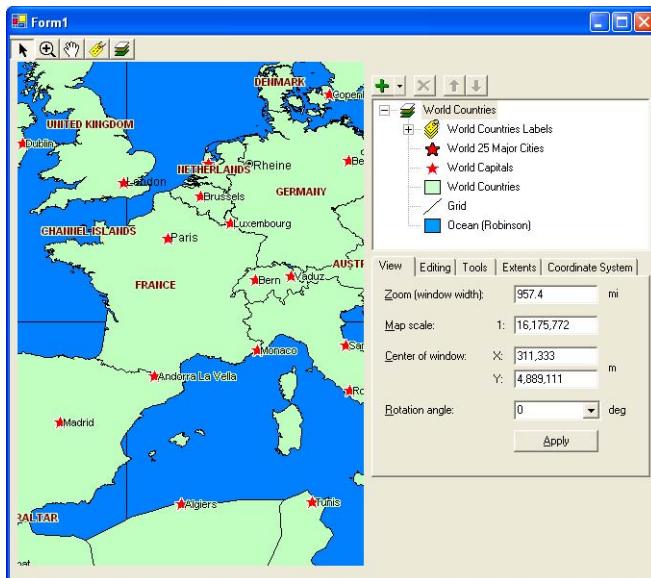
虽然包括在 *MapXtreme* 中的所有控件都非常有用，但有几个控件比其它控件在大多数地图绘制应用程序中更为重要。以下章节将讨论这些关键控件。其中的每个控件都假定项目中包括了适当的引用。（如果已在 Visual Studio 中通过 *MapXtreme MapForm* 模板创建项目，则会自动添加这些引用。）

-
- ①** 如果未从我们的模板之一创建项目，则请确保将引用添加到相应组件（例如：*MapInfo.CoreEngine*、*MapInfo.Windows*、*MapInfo.Windows.Dialogs* 和 *MapInfo.Windows.Framework*）。另请参阅 **MapXtreme 合并模块**。

MapControl

每个显示地图的应用程序都需要 MapControl。要将 MapControl 添加到窗体上，只需将它从 Visual Studio 的工具箱中拖放到窗体上即可。MapControl 位于窗体之后，可对其执行多种操作，让用户可以更好地使用地图。

在窗体上选择 MapControl 时可通过“属性”窗口执行对 MapControl 的修改。在设计模式中右键单击 MapControl 即可显示包含以下选项的上下文菜单：加载地图、清除地图、图层控件和创建主题图。加载地图打开一个标准文件选取器，开发人员可从中选择要预加载到 MapControl 中的地图。清除地图在将 Map Control 添加到窗体或设计过程中的某个位置时清除任何已在控件中的地图。图层控制启动允许定制地图的“图层控件”对话框（请参阅[图层控制](#)）。创建主题图启动可在地图上创建主题的 CreateThemeWizard（请参阅 [CreateThemeWizard](#)）。



显示 MapControl、图层控件和 MapToolBar 的窗体

MapControl 还具有多个与已加载的地图相关的属性，可以预设置已加载的地图，为特定应用程序进一步定制在运行时显示的地图。可以修改的属性包括：缩放级别、坐标系、地图中心和运行时期间三个鼠标按键的设置。可将鼠标按键设置为 Zoom、Pan、Draw geometries、Select 或其它任何几个地图工具。请参阅 [MapInfo.Tools 命名空间概述](#)。

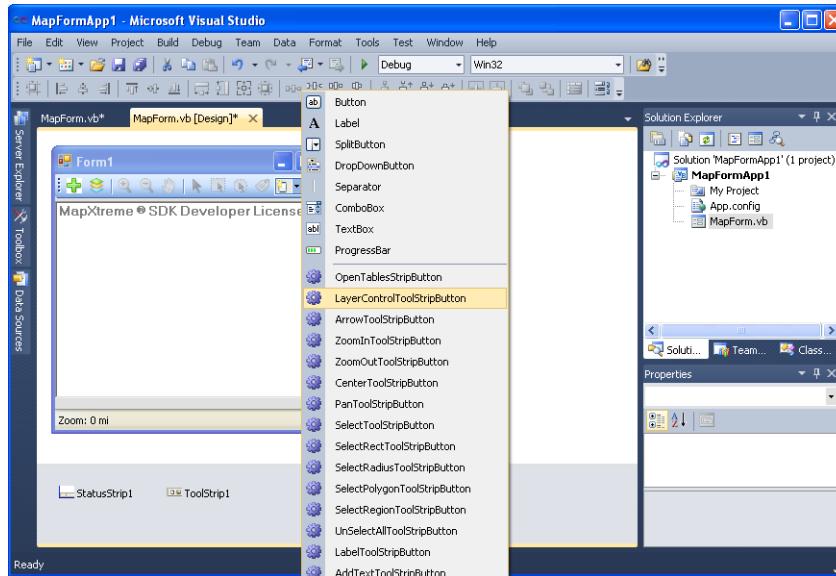
还可以设置设计阶段工具以在 Visual Studio 中工作时进一步操控地图。设计阶段功能包括 Zoom In、Zoom Out、Select、Pan、Center 和默认的箭头。上下文菜单中的“清除地图”可从 MapControl 的地图中删除所有图层并关闭相应的表（如果未由另一 MapControl 使用）。

MapToolStripButtons

MapXtreme 在其 Windows 窗体模板和 Visual Studio 设计器工具箱中提供了立即可用的地图工具。通过把 ToolStripButton 添加到 .NET ToolStrip 将工具添加到窗口。

在按钮上设置每个工具的属性。不再需要负责管理当前工具的 MapToolBar。若选择，则保留 MapToolBar 的向后兼容性并可在未来使用。在 Visual Studio 工具箱中仍提供 MapToolBar。

22 ToolStripButtons 包括始终可在 MapToolBar 中使用的相同的绘图活动，包括：导航、选择集、添加图元、标注、打开表和显示图层控制。



只可将两个 ToolStripButtons 添加到 ToolStrip 或 StatusStrip。用于显示当前地图视图缩放级别和比例的工具只能添加到 StatusStrip。

要添加新的 ToolStripButton，请在 Visual Studio 中从“菜单”和“工具栏”下的工具箱里选择 ToolStrip。出现在窗口上的分割按钮会显示一个小小的向下箭头。单击该箭头显示列表支持的 ToolStripButtons，然后选择所需的选项。一旦添加到 MapControl，该工具就会立即与其关联在一起。

ToolStripButtons 的 API 包含在 `MapInfo.Windows.Controls` 命名空间中。所有源自抽象基类的 ToolStripButtons 都是 `MapInfo.Windows.Controls.MapToolStripButtonBase`，而这又是继承 .Net `System.Windows.Controls.ToolStripButton` 类。

有关如何通过编程方式使用 ToolStripButtons 的信息，请参阅 MapForm 模板中的代码并访问“开发人员参考”。有关每个工具的行为信息，请参阅 `MapInfo.Tools` 命名空间中的单个 MapTool 类。

MapToolBar

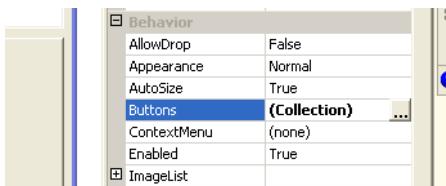
从 7.0 版本开始，MapForm 模板中已用 ToolStrip 来替代 MapToolBar。请参阅 [MapToolStripButtons](#)。在 Visual Studio 工具箱中仍可以使用 MapToolBar。本节概述了如何把 MapToolBar 添加到 MapForm。

MapToolBar 将多个地图工具控件（例如，放大和打开表）组合为单个控件。默认工具集合包括：OpenTable、LayerControl、Select、ZoomIn、ZoomOut、Pan 和 Center。

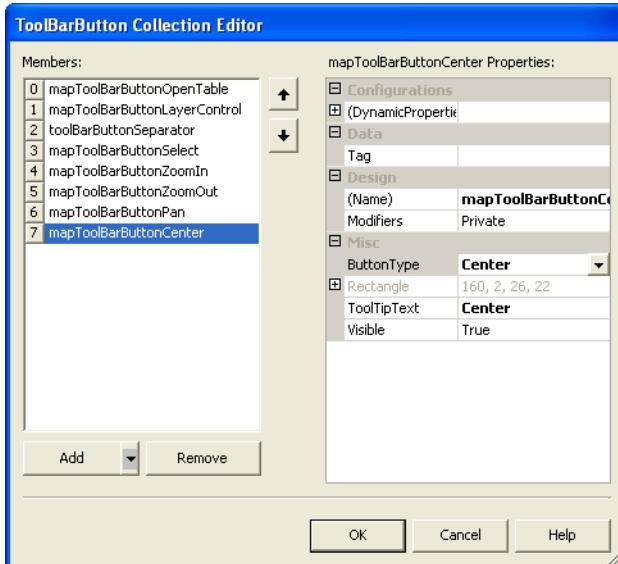
要将 MapToolBar 添加到 MapForm，请从 Visual Studio 工具箱中将其拖出。（MapForm 模板中不再提供 MapToolBar。）

要将工具按钮添加到 MapToolBar：

1. 在“属性”窗口中通过单击 Buttons 属性旁边的省略号 (...), 打开 ToolBarButton 集合编辑器。



将显示“ToolBarButton 集合编辑器”。



2. 单击添加按钮。

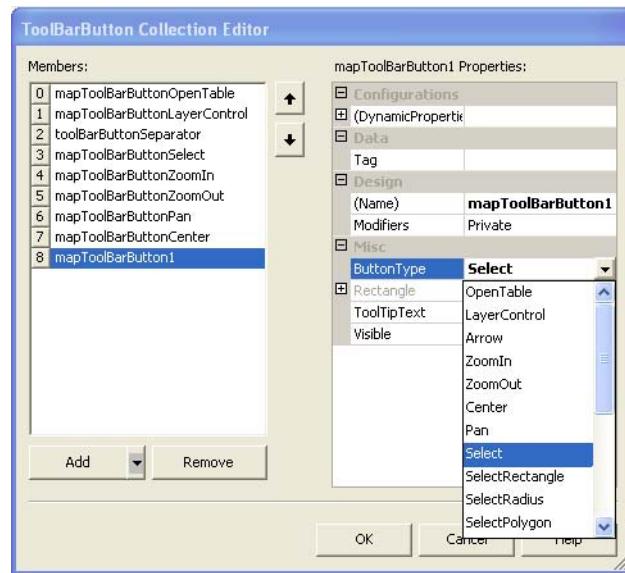
将在列表中最后一个按钮的后面创建新的 MapToolBarButton。



如果想要添加定制的工具，则需要单击“添加”按钮旁的箭头并选择 ToolBarButton。需要为新按钮编写定制的处理程序。

3. 命名新按钮。
4. 从右侧下拉菜单选择 **ButtonType**。

例如，如果添加 **Select** 工具按钮，则从列表中选择 **Select**。



5. 单击确定。

现在创建了 **MapToolBarButton** 并将其添加到按钮栏。新添加的工具在默认方式下工作。

将自定义按钮添加到 **ToolBar**

该程序假定已创建自定义工具并将其添加到 **MapControl.Tools** 集合。添加自定义按钮时，要将按钮的 **Toolid** 指定为自定义工具的名称，还要将自定义位图添加到 **MapToolBar** 的 **ImageList** 中。

有关如何创建自定义桌面工具的详细信息，请参阅[定制工具](#)。

有关 **MapXtreme** 包含的自定义位图符号的详细信息，请参阅[附录 F：定制符号](#)。

要将自定义按钮添加到工具栏：

1. 将 **MapToolBar** 添加到窗体。
2. 在 **Visual Studio** 属性窗口中，突出显示 **Buttons** 属性并按下 ... 按钮，调用 **ToolBarButton** 集合编辑器对话框。
3. 单击添加按钮，添加新的 **MapToolBarButton**。
4. 将按钮的 **ButtonType** 属性设置为 **CustomTool**（下拉列表中最后一项）。注意，按钮将显示为空白，因为暂时没有与该自定义按钮关联的图像。
5. 将按钮的 **Toolid** 属性设置为自定义工具的名称，该工具已添加到 **MapControl** 的工具集合。如果 **Toolid** 值与集合中任意工具都不匹配，则在用户单击该按钮时抛出运行时异常。

6. 单击确定，关闭 ToolBarButton 集合编辑器对话框。
7. 调用 InitializeComponent 之后，在窗体的构造函数中将要添加自定义位图的节点添加到 MapToolBar 的 ImageList。自定义位图与应用程序的关联方式由程序员决定。一种选择是在设计阶段将 ImageList 添加到窗体，用自定义图像填充，然后在运行时写入代码以便将其图像传输到工具栏的 ImageList。代码将如下所示：

```
// Add custom tool button's bitmap to toolbar's image list
foreach (Image image in this.imageList1.Images) {
    this.mapToolBar1.ImageList.Images.Add(image);
}

// Associate the bitmap with the custom tool's button (last image
in the list)
this.mapToolBarButtonBlueSelect.ImageIndex =
this.mapToolBar1.ImageList.Images.Count - 1;
```

图层控制

“图层控件”对话框用于显示构成当前地图的所有图层和图层属性的状态。这些属性包括：可视、可编辑、可选择和自动标注。每个复选框列上面的图标表示这些属性类型。当将光标移到属性图标上面时，图标上面将显示“工具提示”以帮助您熟悉每个图标。用复选框可轻松更改一个或多个图层的属性。

“图层控件”还具有可用于更改 Display 和 Label 设置；修改任何已创建的主题地图以及重新排序、添加或删除图层的选项。

可以通过在“图层控件图层”列表中拖放图层来重新排序 LayerControl 中的图层。

① 将图层放在“标注图层”上时会将新的标注集添加到“标注图层”。例如，如果尝试将图层正好移动到“标注图层”上方的位置，就会偶然发生这种情况。

提示：如果想要将图层放在“标注图层”之外，则在放下图层之前按下 Shift 键。按下并松开 Shift 键时，光标会发生更改以指示要将图层放在目标图层的上面还是目标图层的内部。

“图层控制”将“图层控件”对话框的所有功能放在一个窗体上。必须向 Form_Load() 方法中添加一行代码，以便将“图层控件”链接到 MapControl 的地图。

```
layerControl1.Map = mapControl1.Map;
```

要更好地了解“图层控件”的复杂性和功能，请运行名为 Workspace 管理器的 MapXtreme 实用程序。该工具包括“图层控件”的运用示例。请参阅第 24 章：Workspace 管理器。可通过 MapInfo\MapXtreme\7.x.x 下的“程序”菜单访问 Workspace Manager。

定制上下文菜单

可为“图层控件”创建自定义上下文菜单项，这些项目在用户右键单击图层树中的节点时显示。使用 ContextMenuTargetObject 属性返回用户右键单击的对象。

提供的代码示例说明了如何定义 **LayerControlEnhancer** 类，该类允许用户将自定义项目添加到“图层控件”的上下文菜单中。可在位于

..\\MapInfo\\MapXtreme\\7.x.x\\Samples\\Desktop\\Features\\LayerControl directory 目录下的 **LayerControl** 示例应用程序中找到该示例。

快捷键编程

可以创建用于访问 **LayerControl** 工具栏上全部内容的快捷键。**PerformDown**、**PerformUp** 和 **PerformRemove** 方法提供了对“下移”、“上移”和“删除”按钮的编程式访问。

AddMenuMnemonic 属性提供了显示“添加”菜单的键的编程式访问。**AddMenuMnemonic** 属性的语法为：

```
public System.Windows.Forms.Keys AddMenuMnemonic {get; set;}
```

MapInfo.Windows.Dialogs 命名空间

MapInfo.Windows.Dialogs 命名空间包括创建具有特定功能的对话框的类。与控件相反，对话框在设计阶段不是可视的，并且只能在代码中创建和配置。可以使用 **MapInfo.Windows.Dialogs** 命名空间中的类创建您自己定制的对话框，然后通过调用 **System.Windows.Forms.Form.ShowDialog()** 方法来利用它们。

常用对话框是通过 **MapXtreme** 对象模型的公共 API 构建的。其中没有隐藏的内部字段或私有项目。可使用这些对话框或按需对其进行自定义，也可从头自行编写。可能需要减少常用对话框中的控件，以禁止用户更改某些行为。

要在应用程序中使用特定对话框，请按下面的办法将 **MapInfo.Windows.Dialogs** 命名空间添加到代码：

使用 **MapInfo.Windows.Dialogs**

```
private void DoLayerControl()
{
    LayerControlDlg layerControl = new LayerControlDlg();
    layerControl.Map = mapControl1.Map;
    layerControl.ShowDialog(this);
}
```

上面的代码在调用 **DoLayerControl()** 方法时显示 **LayerControl** 对话框。

常用对话框

MapXtreme 包含许多常用对话框和控件，用于处理操纵 **MapXtreme** 对象的任务。**Workspace** 管理器中使用的所有对话框和控件都可在您的应用程序中使用。这些对话框只是低级控件的容器。这些控件设计用于在特殊情况下与多数 **MapXtreme** 对象一起使用。

常用对话框使用包含 **MapXtreme** 的控件创建基本的 UI 组件。这些控件置于对话框中，用于创建特定的 UI 组件。同样地，您可以使用基本控件设计自己的对话框。要创建对话框，只需创建对话框类，然后将控件拖放到表面。

也可以通过可视化继承来自定义常用对话框。可以从我们的对话框中派生类，然后通过重载方法和属性自定义某些行为。其它对话框（如“图层控件”）都可定制，因此可以更改默认行为，以便隐藏不想让用户访问的控件、更改图标的外观或删除选项卡。

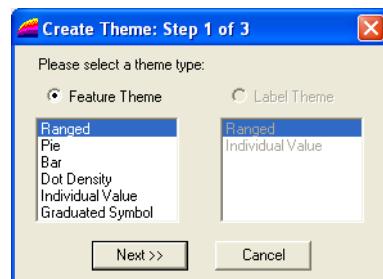
基本上可以控制 **MapXtreme** 包含的所有 UI 组件。这样就可以创建自己的自定义界面。使用上述任何方法都可以创建自定义桌面应用程序，该应用程序只公开所需内容，而不是 **MapXtreme** 定义的所有内容。

CreateThemeWizard

CreateThemeWizard 类是一种可添加到应用程序的简单类。该类可以启动向导（一组对话框）以指导用户完成创建新主题的过程。创建主题有以下三个基本步骤：1) 选择主题类型；2) 选择要用作主题的表和列；以及 3) 修改主题属性（样式、范围数量等）。向导将所有这些设置集成到连续的对话框，以使 **MapXtreme** 应用程序的最终用户可以很简单地创建主题。

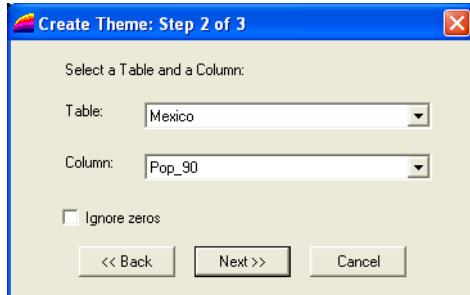
使用 CreateThemeWizard

根据用户指定的选择，**CreateThemeWizard** 显示几种不同的对话框。通过第一个对话框可以选择主题类型。



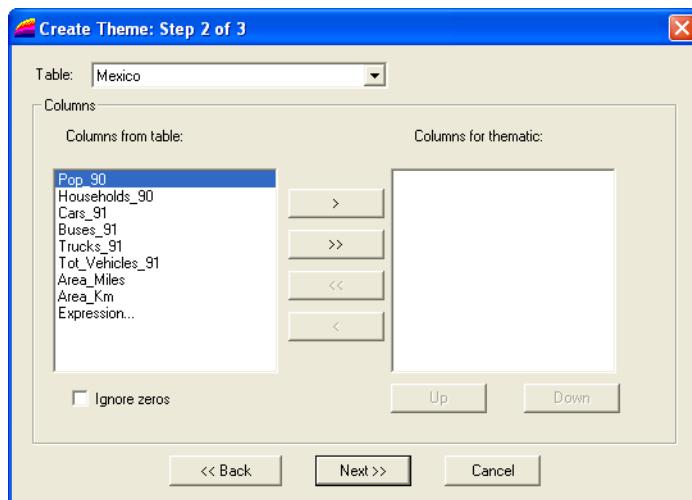
“创建主题图：步骤 1 (共 3 步)”对话框

如果用户选择单变量主题类型（“范围”、“点密度”、“单值”或“分级符号”），则显示的第二个对话框允许选择表和单一列。



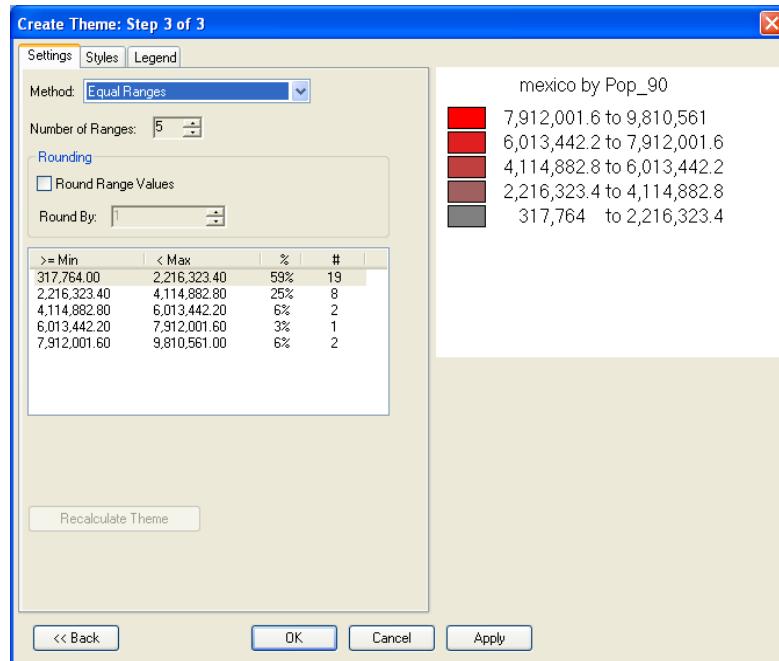
“创建主题图：步骤 2（共 3 步）”对话框（单一列）

如果用户选择多变量主题（饼图或条形图），则显示的第二个对话框允许选择从中创建主题的多列数据。



“创建主题图：步骤 2（共 3 步）”对话框（多列）

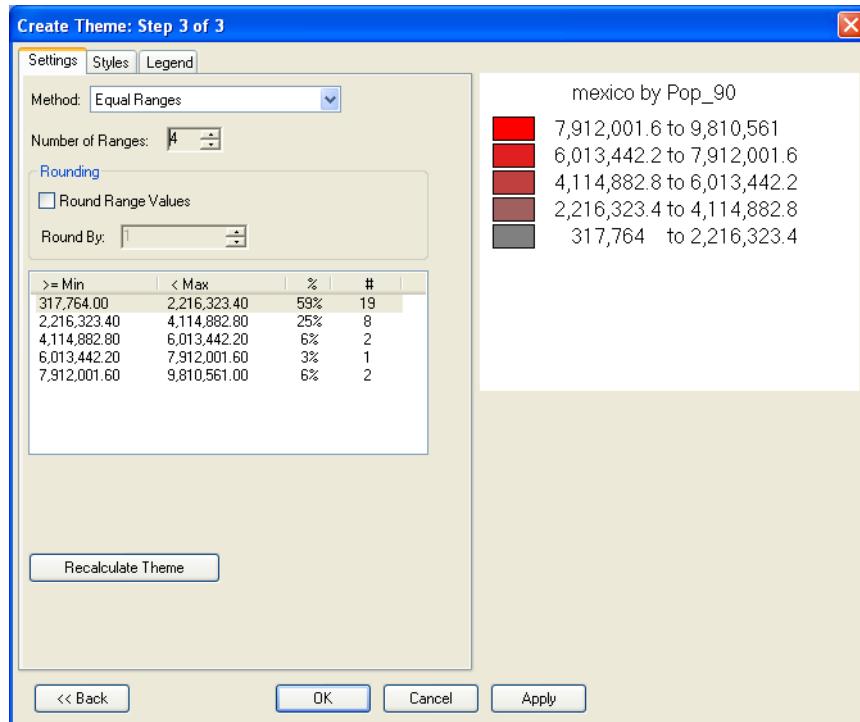
以下对话框取决于步骤 1 中选择的主题类型。下图显示的是范围主题特定的“步骤 3”对话框。



“创建主题图：步骤 3（共 3 步）”对话框（范围主题）

对话框显示默认的主题分发方法：`EqualRangeSize`。此分发方法在向导用户界面中的对应名称是“范围相等”。

用户在“设置”选项卡中进行更改（例如更改范围数量）时，将会激活“重新计算主题图”按钮。用户必须单击重新计算主题图、应用、确定按钮，或者在新选项卡上才能执行重新计算。重新计算并不会自动发生。下图显示了修改范围数量的“步骤 3”对话框。此时，重新计算主题图按钮处于可用状态。



更改设置后的“创建主题图：步骤 3（共 3 步）”对话框（范围主题）

通过程序化的方式检查“设置”选项卡的值是否有变化以及在执行以下操作后是否需要重新计算主题图容器：

- 切换到“样式”或“图例”选项卡
- 单击“应用”按钮
- 单击“确定”（接受）按钮

根据应用程序中的控件，在设计阶段和运行时，可从多个不同地方访问 CreateThemeWizard。设计时，可从 MapControl 属性窗口底部的菜单中访问 CreateThemeWizard；还可以通过右键单击某个图层，然后选择添加主题，从图层控件中对其进行访问。

使用 CreateThemeWizard 进行开发

要通过编程方式添加 CreateThemeWizard，请执行以下示例代码。

```
using MapInfo.Windows.Dialogs;
```

```
CreateThemeWizard themeWizard = new CreateThemeWizard(mapcontrol1.Map,  
this);
```

方法

创建向导后，可从 `CreateThematicWizard` 类调用 `CreateTheme()` 方法来创建主题图。可添加可选字符串参数以设置主题别名。在 `ObjectTheme` (`Bar`、`Pie` 和 `GraduatedSymbol`) 情况下，可以使用别名访问图层集合中的主题，或者在 `FeatureStyleModifier` (`Ranged`、`RangedLabel`、`IndividualValue`、`IndividualValueLabel` 和 `Dot Density`) 情况下，可以使用别名访问修饰符集合。可以访问主题以进行修改或删除。

```
ITheme theme = themeWizard.CreateTheme("theme1");
```

如果已经了解要作为主题列出的表或要用于标注主题的标注源，则可以使用方法 `CreateFeatureTheme()` (用于表) 或 `CreateLabelTheme()` (用于标注)。

属性

以下是一些属性，对获取维护有关主题创建的信息非常有用。`WizardResult` 包括允许检查用户如何退出向导的枚举 `WizardStepResult`。这些选项分别为 `WizardStepResult.Done` 和 `WizardStepResult.Cancel`。这些选项可用于更新应用程序中的相应控件或菜单。

以下示例说明了如何使用。

```
if (createThemeWizard.WizardResult == WizardStepResult.Done
{
    // Update the menus
    mnuRemoveTheme.Enabled = true;
    mnuModifyTheme.Enabled = true;
}
```

`SelectedLabelSource`、`SelectedLayer` 和 `SelectedThemeType` 是 `CreateThemeWizard` 类上可用于找出主题已应用了哪些图层或标注源的属性。这在想要修改主题或需要访问对象时非常有用。

要修改主题，首先要确定主题的类型，然后启动适当类型的修改对话框。用于修改主题的对话框类包括 `ModifyBarThemeDlg`、`ModifyDotDensityThemeDlg`、`ModifyGradSymbolThemeDlg`、`ModifyIndValueThemeDlg`、`ModifyPieThemeDlg` 和 `ModifyRangedThemeDlg`。

`MapXtreme` 包含 `ThemeDialogs` 示例应用程序，可以分析其中的实现并根据需要对它进行自定义。请参阅 `..\MapInfo\MapXtreme\7.x.x\Samples\Desktop\Features\ThemeDialogs` (在 `MapXtreme` 安装下)。

定制控件和对话框

`MapXtreme` 在 `MapInfo.Windows.Dialogs` 命名空间中随附了多种对话框类。其中的每个对话框类都具有特定于每个对话框类的定制功能。要修改特定对话框，只需将值指定给特定于每个对话框的不同属性。因为将值指定给属性的情况非常多，所以无法列出全部的定制设置。请参阅联机参考以获取关于每个对话框的详细信息。

自定义对话框的示例如下：`LineStyleDlg` 类创建一个“线样式”对话框。此对话框可以使用稀疏样式模式（打开时未选择任何内容）创建或使用某些选项是可视的但未激活的模式（灰色）创建。

由设置属性值提供的定制局限于要修改的特定对话对象。通过在设计阶段为新窗体添加一些控件，可基于已建立对话框的设计创建新的对话框。例如，要创建修改后的 **LineStyle** 对话框，可以将 **LineStyle** 控件放在新窗体上，然后为那个窗体添加其它控件，并使它们对已修改的 **LineStyle Dialog** 框生效。

MapInfo.Tools 命名空间概述

MapTool 对象模型

MapTool 是用作与地图交互的对象。这些工具实现您需要地图具有的基本行为。有基于视图的工具（缩放和平移），以及在特定图层上创建几何体的工具，还有根据某个地图区域生成选择集的工具。每种工具都是可自定义的，做法是使用事件来捕获预处理和后处理事件。

与工具相关联的方法的属性在工具上，而不是在图层上。例如，图层的可选择性是在工具上指定，而不是在图层上指定。以此方法，可以使用一个工具在某图层上选择对象，而使用另一个工具从另一图层上选择。这提供了更多在工具中使用的自由，而且您还可以仿效图层上的属性。“图层控制”在地图的整个工具集中操作，这时选择集或其它属性通过其 UI 更改。

此设计还允许您创建插入特定图层的工具。例如，城市规划者可能想要应用程序将探井仅插入水图层中，而将树插入蔬菜图层中。您可以创建一个自定义工具，选择时仅在适当的图层插入指定的符号或几何体。对常用工具也起作用，因此绘制多边形将始终在指定的图层上进行。还有，常用“图层控制”操纵工具集，使其象是插入在单独的图层上。

MapInfo.Tools 命名空间包含了所有允许为 **MapXtreme** 应用程序创建基本和可定制工具的类。

可以使用事件代码自定义这些常用工具的新实例，或编写从常用工具派生的您自己的工具类，但这样会覆盖特定的方法。请参阅第 139 页中的 **定制工具**。

可将工具指定给特定鼠标按键（鼠标左键、鼠标右键或鼠标中键）。请为适当的鼠标按钮属性赋以下字符串工具名：“Arrow”、“ZoomIn”、“ZoomOut”、“Center”、“Pan”、“SelectPoint”、“SelectRect”、“SelectRadius”、“SelectPolygon”、“SelectRegion”、“AddPoint”、“AddLine”、“AddPolyline”、“AddPolygon”、“AddRectangle”、“AddCircle”、“AddEllipse”、“AddText” 和 “Label”。

例如，下面的代码会将鼠标左键设置为 **ZoomIn** 工具：

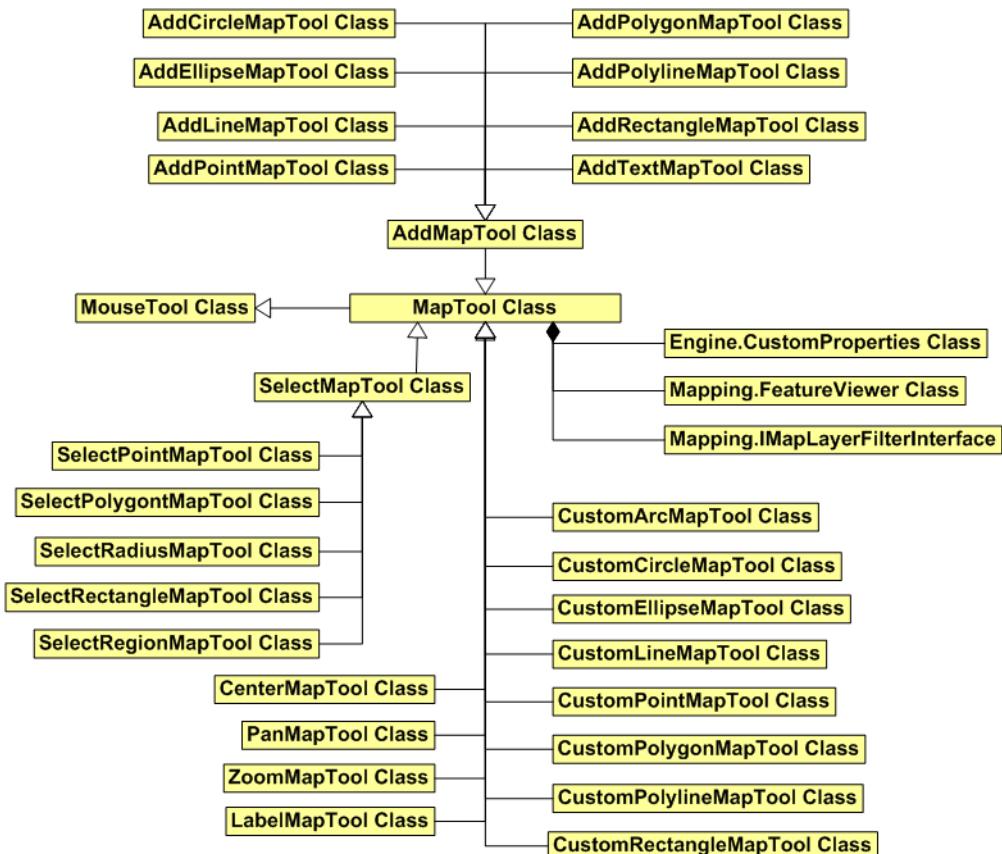
```
mapControl1.Tools.LeftButtonTool = "ZoomIn"
```

要获得通过编程方式使用桌面工具的示例，请参阅 **MapXtreme** 安装目录

..\\MapInfo\\MapXtreme\\7.x.x\\Samples\\Desktop\\Features\\DesktopTools 下的示例应用程序。

MapXtreme 桌面工具 API

下面的图表显示了 MapTool 类的 UML 表示。



工具基本上都是从 MapTool 类派生的。Select 和 Add 工具分别从 SelectMapTool 和 AddMapTool 类派生。

此名称空间中包含的工具可分为四类：View 工具、Select 工具、Add 工具和 Custom 工具。

在 Add 中，Custom 和 Select 工具类别是用于绘制或选择以下这些几何体对象的工具：Ellipse、Arc、Circle、Rectangle、Polygon、Point、Line 和 Polyline。

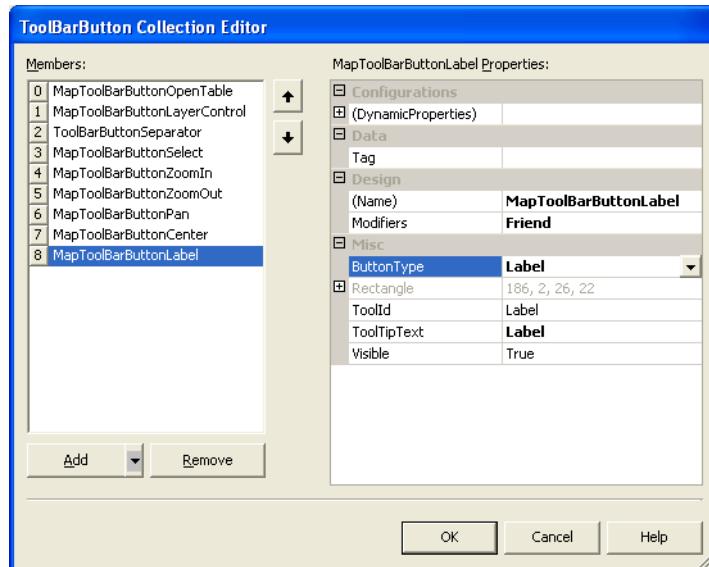
所有工具都具有与它们实现三个接口相关联的属性：IMapToolProperties、ISelectMapToolProperties 和 IAddMapToolProperties。

View 工具

View 工具（ZoomMapTool、CenterMapTool 和 PanMapTool）将地图的视图直接更改为正在使用的工具。这些工具是默认工具集合的一部分。若使用 MapXtreme 模板构建桌面应用程序，则这些工具将出现在默认工具栏上。

LabelMapTool 可以在地图上标注图元。该工具未包括在默认工具集合内。MapForm 模板上。要将该工具添加到 MapControl，首先从 Visual Studio 工具箱中添加一个 ToolStrip，然后右键单击上下文菜单项 LabelToolStripButton。

要进行添加。单击 MapToolBar 的 Buttons 属性下面的“集合...”按钮。



Select 工具

Select 工具系列（SelectMapTool、SelectPointMapTool、SelectPolygonMapTool、SelectRadiusMapTool、SelectRectMapTool 和 SelectRegionMapTool）选择位于由这些工具定形的几何对象之内的点和区域。例如，SelectRectMapTool 会选择位于矩形内的对象。SelectMapTool 是派生其它 select 工具的抽象基类。



MapXtreme 不支持对象与折线的交集，因此无 select polyline 工具。

SelectPointMapTool 可用于旋转对象。

SelectNode 节点模式允许将独立点从对象中移动、添加或删除。有关示例，请参阅第 141 页中的 使用 **Select** 工具编辑 FeatureGeometry。

使用 `SelectRectMapTool`、`SelectRadiusMapTool` 和 `SelectPolygonMapTool`，动态 `Selection` 显示当鼠标移动时何种对象可以在选择集内。

`Selection` 地图工具具有设置各种选项的属性，包括哪些图层是可选的。这些默认的工具属性可在 `MapTools` 集合上设置，该集合位于 `MapInfo.Windows.Controls.MapControl.Tools` 中。每个工具也可以设置自己的重载属性并指定使用 `MapTools` 集合上的默认值或自己的重载值。

要设置可选图层的默认列表，请创建 `IMapLayerFilter`，指定哪些图层是可选的。然后在 `MapTools` 上为 `SelectableLayerFilter` 设置默认属性。

例如，要设置在所有矢量图层上使用的可选图层的默认列表：

```
// select from all vector layers
IMapLayerFilter selectableLayerFilter =
MapLayerFilterFactory.FilterAnd(MapLayerFilterFactory.FilterVisibleLayers
(true), MapLayerFilterFactory.FilterByLayerType(LayerType.Normal));

mapControl1.Tools.SelectMapToolProperties.SelectableLayerFilter =
selectableLayerFilter;
```

有关 `Select` 工具属性的更多信息，请参阅 `MapXtreme` 开发人员参考中的 `MapInfo.Tools.MapTool.ISelectMapToolProperties`。

Add 工具

`Add` 工具系列（`AddArcMapTool`、`AddCircleMapTool`、`AddEllipseMapTool`、`AddLineMapTool`、`AddPointMapTool`、`AddPolygonMapTool`、`AddPolylineMapTool` 和 `AddRectangleMapTool`）全部基于 `AddMapTool` 类。`Add` 工具可以在地图上绘制特定的几何对象。要将对象增加到地图，那么地图中必须具有需要激活的插入图层，创建对象后，对象会驻留在该图层中。特定对象的大小和形状取决于所使用的工具，并可在工具使用过程中通过修饰符键（`Shift` 和 `Ctrl`）进行约束。

有关 `Add` 工具属性的更多信息，请参阅 `MapXtreme` 开发人员参考帮助中的 `MapInfo.Tools.MapTool.IAddMapToolProperties`。

Custom 工具

`Custom` 工具（`CustomArcMapTool`、`CustomCircleMapTool`、`CustomEllipseMapTool`、`CustomLineMapTool`、`CustomPointMapTool`、`CustomPolygonMapTool`、`CustomPolylineMapTool` 和 `CustomRectangleMapTool`）是非常基本的工具，它们仅用于激发事件。可以使用这些类设计特定工具的定制行为。例如，每次使用这些工具时，都可以用 `CustomEllipseMapTool` 绘制具有红色轮廓的绿色 `Ellipse`。使用各种工具事件（请参阅第 140 页中的 工具事件）指定对这些工具所生成的特定事件作出响应的特定行为。

自定义工具未包括在默认的工具集合内。若要将自定义工具分配给鼠标按键，则可能要提供标识该工具的任意字符串。

有关自定义工具的详细信息请参阅第 139 页中的 定制工具。

Shape 工具

MapXtreme 提供了一组工具，用于在地图上绘制几何图元。这些工具中的大多数都通过“单击和拖放”的方式使用：在地图上单击并拖放到另一位置时，会显示一个橡皮圈图像，它显示正在绘制的对象的当前大小和形状。这些工具都使用 Esc 键取消当前操作（如果适用的话）。

以下是 MapXtreme 支持的 shape 工具类型。

Line 工具

该组中的工具包括 AddLineMapTool 和 CustomLineMapTool。这些工具绘制 Line。通过单击并从起点拖动到终点来激活该工具。松开鼠标按键将创建 Line。如果在拖动过程中按下 Shift 键，则 Line 的角度将被约束为 45 度的倍数。如果在拖动过程中按下 Ctrl 键，则 Line 的长度和高度加倍。在松开鼠标按键之前按下 Esc 键，将取消该操作。

Polyline 工具

该组中的工具包括 AddPolylineMapTool 和 CustomPolylineMapTool。该组中的工具绘制具有多个段的 Line。通过单击一个点然后再单击后续点可以激活该工具。完成单击点之后，再次单击最后一个点，则完成 Polyline 绘制。如果在拖动过程中按下 Shift 键，则 Line 的角度将被约束为 45 度的倍数。Ctrl 键不会影响该工具。在完成线段之前按下 Esc 键，将取消该操作。

Ellipse 和 Arc 工具

该组中的工具包括 AddEllipseMapTool、CustomEllipseMapTool 和 CustomArcMapTool。Ellipse 工具会绘制 Ellipse。CustomArcMapTool 仅会绘制圆弧；但不会将圆弧插入图层。

通过单击并将边缘一端上的点拖动到边缘另一端上的点来激活 Ellipse 工具。松开鼠标按键将创建 Ellipse。如果在拖动过程中按下 Shift 键，则 Ellipse 被约束在半径为常量的圆中。如果在拖动过程中按下 Ctrl 键，则使用中心内的起点绘制 Ellipse。在松开鼠标按键之前按下 Esc 键，将取消该操作。请注意 Ellipse 轴始终与坐标系对齐。

MapXtreme 提供了轻量级的 CustomArcMapTool，该工具在地图上绘制参考圆弧，但不会提供 LegacyArc FeatureGeometry 或将其插入图层中。要创建带有上述行为的 arc 工具，必须创建您自己的实现方法。例如，若想要 CustomArcMapTool 与 AddMapTool 具有相同的操作，则创建从 CustomArcMapTool 派生的新 arc 工具类并实现 IAddMapToolProperties 接口。

Circle 工具

该组中的工具包括 AddCircleMapTool 和 CustomCircleMapTool。Circle 工具绘制圆形。这与按下 Shift 键时使用 Ellipse 工具具有同样的功能。通过单击并从 Circle 中央的中心点拖动到边缘上的点来激活该工具。拖动时，用户可看到 Circle 的大小在该中心点周围进行扩展。松开鼠标按键将创建 Circle。如果在拖动过程中按下 Ctrl 键，则以两个边缘点绘制 Circle。在松开鼠标按键之前按下 Esc 键，将取消该操作。

Rectangle 工具

该组中的工具包括 AddRectangleMapTool 和 CustomRectangleMapTool。这些工具绘制矩形。通过单击并从一个顶点拖动到相对的顶点来激活该工具。松开鼠标按键将创建 Rectangle。注意，绘制的对象始终轴对齐。如果在拖动过程中按下 Shift 键，则 Rectangle 被约束为正方形。如果在拖动过程中按下 Ctrl 键，则使用中心内的起点绘制 Rectangle。在松开鼠标按键之前按下 Esc 键，将取消该操作。

使用 InfoTip

InfoTip 是指将鼠标指针悬停在地图图元上方时出现的小文本框。地图与 MapInfo.Tools.MapTools 的实例相关联，后者包含可在该地图上使用的所有工具（MapInfo.Tools.MapTool 示例）。

MapInfo.Tools.MapTools 类有三个控制 InfoTip 外观的属性。它们是：

- InfoTipsEnabled – 获取/设置在鼠标闲置时是否显示 InfoTip。
- InfoTipTimerDelay – 获取/设置在显示 InfoTip 之前必须等待的时间（单位为毫秒）。默认为 500。
- InfoTipDisplayDelay – 获取/设置 InfoTip 显示的持续时间（单位为毫秒）。默认值为 0，这意味着 InfoTip 将一直显示到鼠标移动为止。

注意：只要鼠标开始移动，就不会显示 InfoTip。只有当鼠标指针位于带有标注的图元上方并且闲置时间达到 InfoTipTimerDelay 指定的最小间隔时间时，才会显示 InfoTip。当鼠标移动或鼠标闲置时间超过 InfoTipTimerDelay 指定的时间（假定该时间大于 0）时，InfoTip 将会消失。

通过以下方法您可以禁止显示 InfoTip：

```
mapControl1.Tools.InfoTipsEnabled = false;
```

通过以下方法您可以设置让 InfoTip 延迟 1 秒显示：

```
mapControl1.Tools.InfoTipTimerDelay = 1000;
```

通过以下方法，您可以将鼠标闲置时 InfoTip 显示的持续时间设置为 2.5 秒：

```
mapControl1.Tools.InfoTipDisplayDelay = 2500;
```

有关在 API 中使用 InfoTip 的更多信息，请参阅《MapXtreme 开发人员参考》中的 MapInfo.Tools.MapTools 类。

定制工具

可通过两种方式为桌面应用程序自定义工具：通过创建现有工具的子类或使用 CustomTools 组中的某个工具来自定义工具。在 MapXtreme 中提供的任何 Tool 类都可通过创建子类自定义，而子类是从需要的特定 Tool 类派生的。下面的示例就是从 AddLineTool 类派生新的工具类。用户可能想将该工具更改为特定行为（例如，绘制直线时，始终发出蜂鸣声）、特定外观（例如，始终以红色绘制直线）或者特定功能（例如，始终将绘制直线的角度约束为添加 90 度）。

另一个示例是基于同一个工具创建两个不同的工具。可以使用 **AddPointTool** 划分两个不同工具的子类，这两个工具使用不同的符号以指示地图上两种类型的点。将其中一个工具指定给鼠标左键，将另一个工具指定给鼠标右键。

以下代码示例说明了 **AddPolygonTool** 的定制。

VB 示例：

```
Dim insertionlayerfilter As IMapLayerFilter
Dim style As MapInfo.Styles.CompositeStyle
Dim addmaptoolproperties As MapInfo.Tools.AddMapToolProperties
Dim maptool As MapInfo.Tools.MapTool

insertionlayerfilter = _
    MapLayerFilterFactory.FilterByLayerType(LayerType.Normal)

style = New MapInfo.Styles.CompositeStyle

addmaptoolproperties = New _
    MapInfo.Tools.AddMapToolProperties(MapLayerFilterFactory.Filter_ _
        ForTools(MapControl1.Map, insertionlayerfilter, _
            MapLayerFilterFactory.FilterVisibleLayers(True), _
            "CustomPolygonAddMapToolProperties", Nothing), style)

maptool = New MapInfo.Tools.AddMapTool(MapControl1.Viewer, _
    MapControl1.Handle.ToInt32(), MapControl1.Tools, New _
        MapInfo.Tools.MouseToolProperties(Cursors.Default, _
            Cursors.Default.Cursors.Default), _
        MapControl1.Tools.MapToolProperties, addmaptoolproperties)
```

此外，**MapXtreme** 提供了桌面工具示例应用程序，在该应用程序中可以按需分析和实现桌面工具。

请参阅 **MapXtreme** 安装目录下的

..\\MapInfo\\MapXtreme\\7.x.x\\Samples\\Desktop\\Features\\DesktopTools\\cs。

工具事件

MapInfo.Tools 命名空间支持含有信息的工具事件，还有用于取消工具操作的功能。**Select** 工具事件用于列出所选或取消选中的项目。可在使用特定工具的每个阶段激发事件。在代码中，可以在工具使用周期中的某些时刻设置陷阱。在地图绘制应用程序中定制工具时，将代码添加至特定事件就提供了极大的灵活性。

命名空间中的事件如下所示：

FeatureAddingEventArgs	在添加工具将要绘制对象时激发该事件。
FeatureAddedEventArgs	在添加工具已将对象添加到表和地图时，激发该事件。
FeatureSelectingEventArgs	在选择工具将要更改选择集时激发该事件。

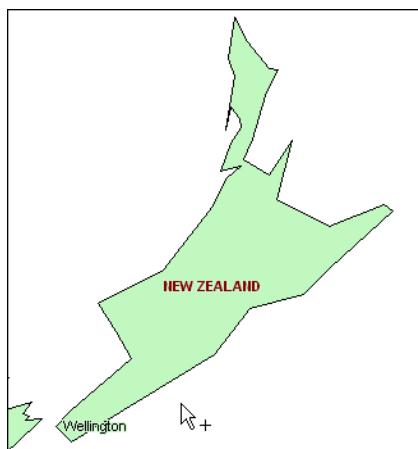
FeatureSelectedEventArgs	在选择工具更改选择集时激发该事件。
FeatureChangingEventArgs	在选择工具将要更改图元时激发该事件。使用该事件进行检查以了解将要进行的更改是否有效。
FeatureChangedEventArgs	在选择工具更改图元之后激发该事件。
NodeChangedEventArgs	在选择工具改变选择集中的节点时激发该事件。
NodeChangingEventArgs	在选择工具将要更改选定图元的节点时激发该事件。使用该事件进行检查以了解将要进行的更改是否有效。
ToolActivatedEventArgs	在激活鼠标工具时激发该事件。
ToolActivatingEventArgs	将要激活鼠标工具时激发该事件。
ToolEndingEventArgs	在鼠标工具将要结束时激发该事件。此时可以开始一个新操作。
ToolUsedEventArgs	该事件在鼠标工具正在使用时激发。可以使用该事件为鼠标单击连续点的开始，中间和结束处设置标记。

使用 Select 工具编辑 FeatureGeometry

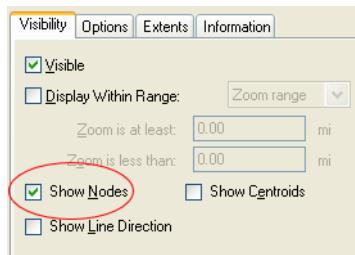
以下部分说明如何通过重新定形节点并将节点增加到 FeatureGeometry 来编辑图元。它说明了 MapControl、LayerControl 和 Select 工具的使用。另请参阅第 123 页中的 [MapControl](#) 和第 127 页中的 [图层控制](#)。

重新定形图元

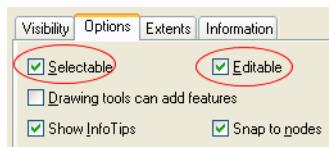
1. 将地图加载到 MapControl 并放大想要修改的 Feature。



2. 在 LayerControl 中选择对象的图层。
3. 在 LayerControl 的“可见性”选项卡上，选择“显示节点”复选框。



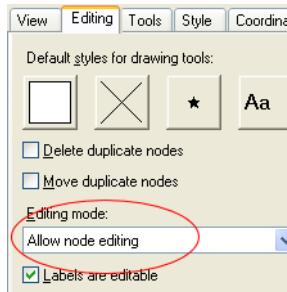
4. 在“选项”选项卡上，选择“可选”和“可编辑”复选框。



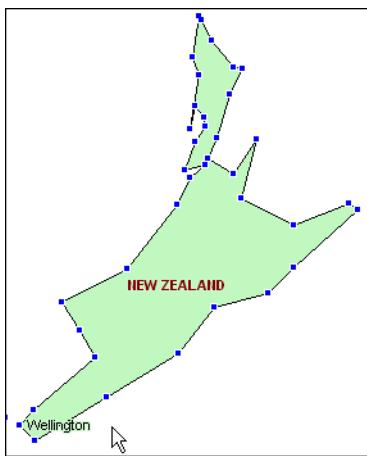
5. 在图层控制树视图中选择地图的根节点。



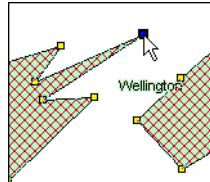
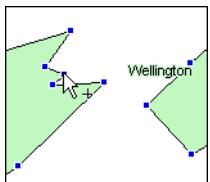
6. 在“编辑”选项卡上，从“编辑”模式组合框选择“允许编辑节点”。



- 单击确定接受更改。此时地图显示将会更改并显示该对象的节点。



- 使用 **Select Item** 工具，选择想要修改的多边形，然后单击并拖动节点以更改其位置。可用 **Shift** 键选择某范围内的节点，也可用 **Ctrl** 键切换节点的选择状态。

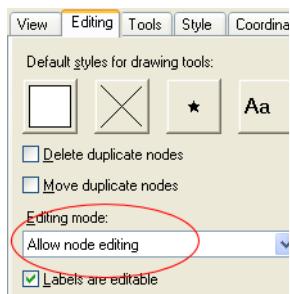


增加节点

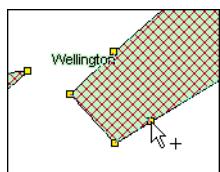
要将节点增加到 Feature Geometry：

- 再次打开 **LayerControl** 并选择地图的根节点。

- 在“编辑”选项卡上，从“编辑”模式组合框选择“允许增加节点”。



- 单击确定接受更改。
- 在地图上，选择想要修改的多边形。
- 使用 Select Item 工具，按住 Ctrl 键并单击想要增加节点的多边形的边缘。此时将会显示新节点。



通过编程方式重新定形和增加节点

本部分说明了如何通过编程方式重新定形和增加节点。提供有 C# 和 VB 代码。

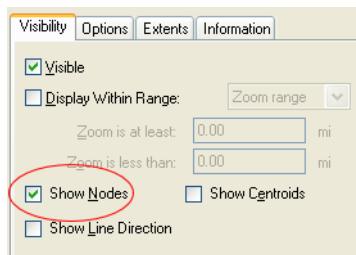
- 将名为 "btnEditNodeTool" 的按钮增加到 Visual Studio 设计器的主窗体中。
- 双击该按钮以在代码页中打开该按钮的句柄。增加适当的代码示例：

VB 示例：

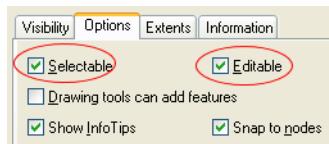
```
Private Sub btnEditNodeTool_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    MapControl1.Tools.LeftButtonTool = "Select"
    MapControl1.Tools.SelectMapToolProperties.EditMode =
        MapInfo.ToolsEditMode.Nodes
End Sub
```

- 运行应用程序。
- 在 LayerControl 中选择对象的图层。

5. 在 LayerControl 的“可见性”选项卡上，选择“显示节点”复选框。



6. 在“选项”选项卡上，选择“可选”和“可编辑”复选框。



7. 单击确定接受更改。

8. 现在，单击已增加到窗体的新按钮。此时，光标将会更改为 Select 箭头。
9. 选择要修改的对象，然后单击并拖动节点（与上一示例步骤 1 中所述）。

使用数据

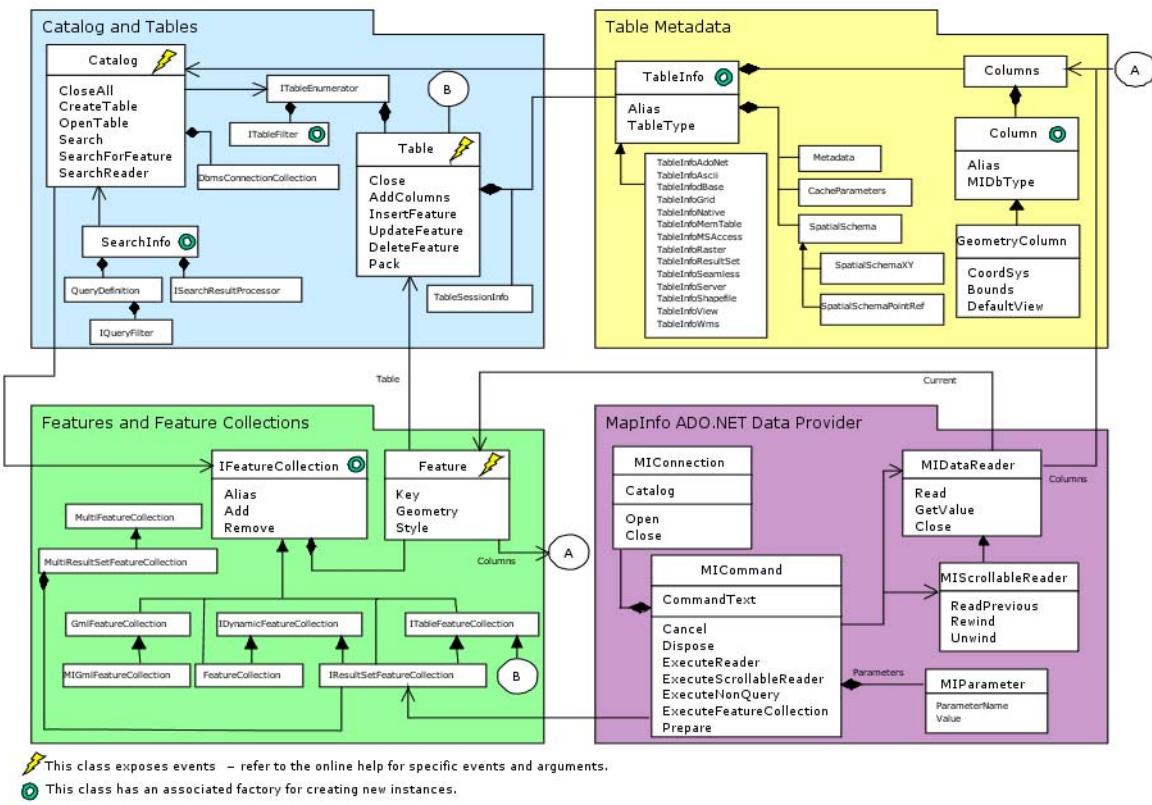
MapInfo.Data 命名空间包含了提供多种方式访问 MapXtreme 应用程序数据的类和接口。

在本章中：

◆ MapInfo.Data 命名空间概述	147
◆ 目录和表	148
◆ 支持的表类型	153
◆ 处理目录和表	155
◆ 表元数据 (TableInfo)	158
◆ MapInfo ADO.NET 数据提供方	174
◆ Features 和 Feature 集合	178
◆ 分析数据	184
◆ 提升数据访问性能	187

MapInfo.Data 命名空间概述

MapInfo.Data 命名空间包含了提供多种方式访问 MapXtreme 应用程序数据的类和接口。在该命名空间内可以使用具有 MapInfo SQL 语言的 MapInfo ADO.NET 数据提供方，用于数据库和表的标准查询。Feature 对象模型是另一种访问数据的方法，该方法使用对象来代替 SQL。目录是数据访问的起点，包含了以各种方式管理表（打开、关闭、创建）和搜索数据的方法。



本章是按上述 MapXtreme 数据访问模型图编排的，包括以下主题：

- 目录和表
- 支持的表类型
- 表元数据 (TableInfo)
- MapInfo ADO.NET 数据提供方
- Features 和 Feature 集合

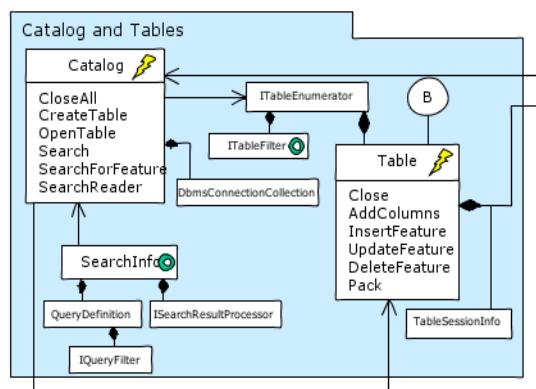
数据访问对任何 MapXtreme 应用程序均至关重要，涵盖各种主题。在以上列示的主题中，有些其他重要主题涉及到不可忽视的信息。

紧接本章之后的是与数据访问相关的两个附加章：[第 10 章：创建表达式](#)和[第 11 章：从 DBMS 访问数据](#)。

目录和表

目录是 **MapXtreme** 数据访问模型的管理器。表是 **MapXtreme** 的基本单元。表用于保存想要在应用程序中显示和分析的数据。作为管理器的目录保存有目前在会话中打开的表的列表。表也可从目录打开、创建和关闭。

几乎所有的 **MapXtreme** 数据访问操作均涉及到目录和表。



表

Table 类是所有数据访问的基本单元。**Table**、**Column** 和所有 **TAB** 文件元数据信息可以从 **MapInfo Table** 访问。表可以是可制图的（包含 **FeatureGeometry** 类型的列）或不可制图的。也可在不显示地图时打开和访问表。

表别名

打开表时，这些表可以分配名称（或别名），为引用该表而打开表时使用该名称（或别名）。例如，在 SQL 语句中可按其别名引用表。从 **TAB** 文件打开的表如果未指定别名，则分配默认的别名。默认的别名基于 **TAB** 文件的名称。该属性是可选的，且可设置为 Null。不过，最好分配别名。

列

Column 对象标识了表、图元或图元集合中一列的属性，并标识了列名称（别名）、数据类型、宽度（对于字符串和小数列）以及列的其它属性。

支持的数据类型包括：

Data Type	Description
Int	提供 32 位带符号整数。这里映射到 .NET Framework 数据类型 Int32。
SmallInt	提供 16 位带符号整数。这里映射到 .NET Framework 数据类型 Int16。
Double	提供从 -1.79E +308 到 1.79E +308 范围内的浮点数。这里映射到 Double。
dBaseDecimal	提供从内部视为与 Double 相同的浮点数。dBaseDecimal 保持在表中时具有固定的精度和比例。从名称上就可以看出，这是从 dBase 文件格式派生的旧式数据类型。这里映射到 Double。
Boolean	提供布尔值。这里映射到 Boolean。
String	提供变长，用空终止的 UNICODE 字符串值。这里映射到 String。
Date *†	提供日期值。Date 类型在 MapInfo.Data 命名空间中作为结构实现。
DateTime *‡	提供组合的日期和时间值。DateTime 类型映射至 System.DateTime。
Time	提供时间值。支持 9.0 版和更高版本的 MapInfo Professional 表 (TAB 文件) 中的 Time 类型。Time 类型在 MapInfo.Data 命名空间中作为结构实现。
FeatureGeometry	提供 FeatureGeometry。
Binary	提供二进制数据的数组。这里映射到 Byte 值的 Array。
Key	提供表中的键。这是 Table 上 Key 伪列的数据类型。
CoordSys	提供坐标系。该类型存在的惟一目的是将坐标系对象绑定到函数的 MICommand，该函数需要坐标系的规范。
Style	提供了 Style 类的实例。请参阅 MapInfo.Styles.Style。这是存储在 Table 上样式列中 Style 对象的数据类型。
Raster	提供来自表的光栅列的 RasterInfo。这是存储在 Table 上光栅列中 RasterInfo 对象的数据类型。
Grid	来自表的网格列的 GridInfo。这是存储在 Table 上网格列中 GridInfo 对象的数据类型。
Wms	提供来自表的 Wms 列的 WmsClient。这是存储在 Table 上 Wms 列中 WmsClient 对象的数据类型。
TileServer	提供来自表的栅格列的 TileServerInfo。这是存储在 Table 上栅格列中 TileServerInfo 对象的数据类型。

- * 为确保与 MapXtreme 的早期版本向后兼容，`MapInfo.Data.MIDataReader.GetDateTime` 方法支持 `DateTime` 和 `Date` 类型。这两种情况下，均会返回 `System.DateTime` 值。
但 `MapInfo.Data.Column.DataType` 将反映实际的数据类型，即 `Date` 或 `DateTime`。
- † `MapInfo` SQL 函数不支持 `Time` 和 `DateTime` 类型。但调用 / 使用 `DateTime` 类型的 `MapInfo` SQL 函数将会返回 `DateTime` 值的日期部分。有关详细信息，请参阅《`MapInfo` SQL 参考》。
- ‡ `MapInfo` SQL 函数不支持 `Time` 和 `DateTime` 类型。但调用 / 使用 `DateTime` 类型的 `MapInfo` SQL 函数将会返回 `DateTime` 值的日期部分。有关详细信息，请参阅《`MapInfo` SQL 参考》。

时间和日期时间数据源支持

`MapXtreme` 可以从支持的数据源和数据提供方读取 `Date`、`DateTime` 和 `Time` 数据（并且在适用时将其存回）。不同的数据源可能对日期/时间具有不同的类型定义，这可能与 `MapXtreme` 类型精确匹配或不匹配。

支持以下数据源的新数据类型：

- `Mem` 表
- `Native` 表（`TAB` 文件）
- `ADO.NET`
- `Oracle`（通过 `OCI`）
- `MS SQL Server`（通过 `ODBC`）

不支持 `ASCII`、`dBase` 和 `Microsoft Access` 数据源。

远程数据库中的 `Date` 和 `DateTime` 支持

远程数据库可能无法支持 `MapXtreme` 所支持的全部数据类型。下表列出了本地 `TAB` 文件和每个支持的数据库中支持的日期和时间数据类型。

MapXtreme	本地 TAB 文件	ADO.NET	Oracle (OCI)	MS SQL Server
Date	Date		Date	
Time	Time			
DateTime	DateTime	DateTime	DateTime	DateTime

远程数据库中返回列类型和值的变化

新数据类型的加入使远程数据库的返回列类型和值出现了一些变化。下表分别列出了 `MapXtreme` 6.7.x 和 `MapXtreme` 6.8 远程数据库支持类型的返回列类型和值。

服务器 / 数据类型	MapXtreme 6.7.x 返回列类型 / 值	MapXtreme 6.8.0 * 返回列类型 / 值
SQL Server/DateTime [†]	Date/System.DateTime	DateTime/System.DateTime
Oracle/TimeStamp	Date/System.DateTime	DateTime/System.DateTime
Oracle/Date	Date/System.DateTime	Date/MapInfo.Data.Date

* 适用于 6.8.0 或更高版本。

† SQL Server 2005 及之前版本。

以下章节向您提供了 MI_Key、MI_Geometry 和 MI_Style 列的信息。

MI_Key

所有的表都有一个伪列，名称为 MI_Key，返回 Key 的实例。MI_Key 伪列的概念与 MapInfo Professional 和 MapBasic 中的 rowid 伪列类似。与 rowed 不同，该列不是数字列。Key 实例可以与字符串文字互相转换。

MI_Geometry

表、图元或图元集合中的 Geometry 列对象包含 FeatureGeometry 对象并指定如列的坐标系、其所包含的所有几何体对象的整个边界等属性。

大多数表类型的几何体列都被赋予了名称 "Obj"。为了兼容较早版本的 MapX 和 MapInfo Professional，别名 "Obj" 将解析到表中的第一个 GeometryColumn。此外，别名 "MI_Geometry" 也可以用于任何表，指代 "Obj" 所指的列。

MI_Style

具有 Geometry 列的表也包含具有名称 “MI_Style” 的列，或者如果找不到，则源于具有 MIDbType.Style 类型的第一列。该列用于保存 Geometry 对象的样式信息，如多边形的直线宽度和点的符号大小。该列无法独立更新。Style 和 Geometry 列必须同时更新。

在打开 MapInfo 本机格式 (.TAB) 的表时自动创建 MI_Style 列。对于所有其它表类型，该列必须专门创建。如果使用 MapInfo.Data.ColumnFactory.CreateStyleColumn，则创建具有 "MI_Style" 名称（别名）和 MIDbType.Style 数据类型的列。

使用 MISQL 在表中插入行时，请确保在插入语句中包括 MI_Style 列。请参阅以下代码示例：

```
Table tab = MapInfo.Engine.Session.Current.Catalog.GetTable("MapView");
TableInfo ti = TableInfoFactory.CreateTemp("Test",
((MapInfo.Data.GeometryColumn)tab.TableInfo.Columns["Obj"]).CoordSys);
Table tabTemp =
MapInfo.Engine.Session.Current.Catalog.CreateTable(ti);

MICnection conn = new MICnection();
conn.Open();
```

```
MICommand comm = conn.CreateCommand();
comm.CommandText = "Insert Into " + tabTemp.Alias +
    " (Obj, MI_Style) SELECT MI_Point(MI_X(Obj), MI_Y(Obj), '" +
    ((MapInfo.Data.GeometryColumn)tab.TableInfo.Columns["Obj"]).Coord
    Sys.SrsString + "'), MI_Style" + " FROM " + tab.Alias + " WHERE msaname
    = 'Minneapolis-St.Paul, MN-WI' AND Not Obj = Null";
MessageBox.Show(comm.CommandText);
int numChanged = comm.ExecuteNonQuery();

mapControl1.Map.Layers.Add(new FeatureLayer(tabTemp));
mapControl1.Map.SetView(mapControl1.Map.Layers["Test"] as FeatureLayer);
```

Catalog

Catalog 实际上是 **MapXtreme** 数据访问模型的管理器。目录保存有目前在 **MapXtreme Session** 中打开的表的列表。表也可从目录打开、创建和关闭。**Catalog** 可以视为在其中保存所有打开的表的单一数据库，与其实际数据源无关。

每个 **MapXtreme Session** 只管理一个目录。

Catalog 在初始化时不包含任何表。表打开时，别名（或名称）被分配到表或由调用方提供表（或名称）。别名用来标识查询和其它操作中的表。

表可以是可制图的（包含一个空间组件）或不可制图的（仅包含数据列）。**MapXtreme Catalog** 可以打开两种类型，并在查询或连接中使用。

目录提供了工具来创建新建表的定义，以及通过当前打开的表进行枚举。目录还包含了搜索方法，可用于访问打开的表中的数据。

目录有一个 **SQL** 引擎，可用于选择、插入、更新和删除表中的表和数据。**SQL** 引擎可用于连接在目录中定义的任何表（例如，从本地连接到 **SQLServer**，或从 **SQLServer** 连接到 **Oracle**）。目录可代您处理从各种数据源进行的集成。当组织源自于不同数据源的数据时，这就成为一个强大的工具。

MapXtreme Catalog 通过 **MapInfo ADO.NET** 数据提供程序公开。访问表和结果集通过此接口控制。请参阅 **MapInfo ADO.NET** 数据提供方。

代码示例

以下示例说明了如何从 **MapXtreme Session** 对象访问目录，打开多个表并通过目录中仅可编辑的表之后的所有表进行枚举。

VB 示例：

```
Public Shared Sub MapInfo_Data_Catalog()
    ' Catalog is accessible off the Session object
    Dim catalog As Catalog = Session.Current.Catalog

    ' Open a bunch of tables
    Dim table As Table = catalog.OpenTable("States.tab")
    table.SessionInfo.ReadOnly = True ' Make states ReadOnly
```

```
table = catalog.OpenTable("world.tab")
table = catalog.OpenTable("worldcap.tab", "World Capitals")

' Enumerate the catalog directly - includes All tables
Dim t As Table
For Each t In catalog
    Console.Out.WriteLine("Table : {0}", t.Alias)
Next
Console.Out.WriteLine()

' Now enumerate through only tables that are editable (not ReadOnly)
Dim tEnum As ITableEnumerator =
catalog.EnumerateTables(TableFilterFactory.FilterEditableTables())
While tEnum.MoveNext()
    Console.Out.WriteLine("Table: {0}", tEnum.Current.Alias)
End While

Session.Current.Catalog.CloseAll()
End Sub
```

支持的表类型

能够在“数据驻留的位置”访问数据是 **MapXtreme** 的优点之一。这意味着我们尽力在处理广泛的数据格式。以下是在 **MapXtreme** 中支持的表类型：

MapInfo .TAB 格式	MapInfo 本地表格式。 该基于文件的表可以具有关联的 .MAP 文件，该文件包含关于 FeatureGeometry 和 Style 的信息。非空间数据存储于 DAT 文件中。TAB 在缓存时可以用作存储格式。请参阅 新建表 。
dBase	数据存储在 dBase 文件中。 该表可以具有关联的 .MAP 文件，该文件包含关于 FeatureGeometry 和 Style 的信息。非空间数据存储于 DBF 文件中。关联的 .IND 文件拥有一个或多个非空间属性值（字符串、数字和日期）的 B-Tree 索引
ASCII	数据存储在分隔的 CSV 或文本文件中。最大字符串长度为 255 个字符（最多包括两个引号）。ASCII 表仅可插入。 该表可以具有关联的 .MAP 文件，该文件包含关于 FeatureGeometry 和 Style 的信息。非空间数据存储于 .CSV 或 .TXT 文件中。

MS Access	<p>Microsoft Access 数据库表。</p> <p>该基于文件的表位于 Microsoft Access .MDB 数据库，可以具有关联的 .MAP 文件，该文件包含关于 FeatureGeometry 和 Style 的信息。非空间数据存储于 Access 文件中。</p>
Shapefile	<p>ESRI Shapefile 表。</p> <p>这些表是只读的，且支持三维几何体 (X, Y, Z, M)。非空间属性数据以 .DBF 文件格式存储。FeatureGeometry 值以 ESRI .shp 文件格式存储。MapXtreme 不具备访问空间索引的权限。可作为临时或永久的 .MAP 文件支持缓存。可与 MapInfo Professional 共享永久的缓存。它由 TableInfoShapefile 类上的 PersistentCache 属性控制。</p>
MemTable	<p>非空间属性数据的内存中存储。</p> <p>FeatureGeometry 数据和索引存储于磁盘中。这些表是临时的 - 关闭时所有的数据均将丢失。MemTables 是可序列化的。数据可能是永久化的工作空间（仅指数据；键可能在重新加载时被更改）。该表类型在缓存时可用作存储格式。没有与 MemTable 对等的 .TAB 文件。请参阅创建临时 MemTable。</p>
RDBMS Server	<p>空间表存储于远程数据库管理系统（如 SQL Server 或 Oracle）中。</p> <p>该表是由本地 SQL SELECT 语句定义的。MapXtreme 执行查询解析和修改。默认情况下会启用缓存。支持的协议（工具包）包括：OCI (Oracle Spatial) 和 ODBC (SQL Server、SpatialWare 和 XY)。请参阅第 11 章：从 DBMS 访问数据。</p> <p>请参阅安装要求 支持的 RDBMS 列表。</p>
ADO.NET	<p>非空间数据的表基于 ADO.NET DataTable 或 IDbCommand。</p> <p>该表类型通过提供程序特有的实现支持许多不同的数据提供程序。如果没有受 MapXtreme 支持的数据提供程序，就可选择 ADO.NET。ADO.NET 旨在为 Connected (IDBCommand) 和 Disconnected (DataTable) ADO.NET 这两种模型提供支持。IDBTables 是只读的。可强制应用缓存（隐式键）。DataTables 是可编辑的，支持运行时序列化。请参阅使用 ADO.NET 数据提供方。</p>
Raster	<p>包含光栅图像的表。</p> <p>这通常为其他空间表类型提供了基本地图。这些表仅带有单一记录和固定的列唯一 (RasterInfo、MI_Geometry 和 MI_Style)。可使用空间谓词（例如，“within”）将这些表与矢量表连接起来。请参阅第 17 章：运用光栅和网格。</p>

Grid	包含网格图像的表。 该表类型为其他空间表类型提供了基本地图。这些表仅带有单一记录和固定的列架构 (<code>GridInfo</code> 、 <code>MI_Geometry</code> 和 <code>MI_Style</code>)。可使用空间谓词（例如，“ <code>within</code> ”）将这些表与矢量表连接起来。 <code>GridRead</code> 类可提供对网格单元值的访问权限。 <code>MapInfo.Raster.GridCreatorFromFeatures</code> 类用内插器创建网格。请参阅 第 17 章：运用光栅和网格 。
WMS	包含源自于 Web 地图服务 (WMS) 的图像的表。 该表类型为其他空间表类型提供了基本地图。这些表仅带有单一记录和固定的列架构 (<code>GridInfo</code> 、 <code>MI_Geometry</code> 和 <code>MI_Style</code>)。可使用空间谓词（例如，“ <code>within</code> ”）将这些表与矢量表连接起来。如同动态光栅一样，可通过 <code>MapInfo.Wms.WmsClient</code> 访问 WMS 表。请参阅 第 23 章：Web 地图服务 。
Seamless	该类型表用相邻地理组合两个或多个基表。该表显示为单个地图图层。 无缝表特别适用于诸如绘制地图之类的空间查询，在绘制地图时可使用无缝表实现对相应组件表的最佳查询。组成无缝表的组件表可能是矢量或光栅。他们的架构必须全部相同。而且要是只读的。不能直接修改基本组件表。通过排序和合计运算，可检查每个组件表中的每次记录（如果使用的是矢量表，可能会对性能产生重大的影响。）
View	基于 <code>MapInfo SQL Select</code> 语句的视图（非受 <code>Server</code> 表支持的本地 SQL）。 请参阅 视图表 。
ResultSet	该类型表包含搜索结果。 <code>ResultSet</code> 专用于 <code>ResultSetFeatureCollections</code> 。 请参阅 结果集 。
TileServer	包含 <code>TileServer</code> 图像的表。这通常为其他空间表类型提供了基本地图。这些表仅带有单一记录和固定的列架构 (<code>TileServerInfo</code> 、 <code>MI_Geometry</code> 和 <code>MI_Style</code>)。

处理目录和表

本节介绍一些基本的表操作，包括：

- [定位打开的表](#)
- [关闭表](#)
- [打包表](#)
- [侦听表和目录事件](#)

另请参阅 `MapXtreme` 开发人员参考中的 `MapInfo.Data.Table` 类。

定位打开的表

要定位打开的表，必须枚举目录。该操作可以通过使用以下章节中的方法实现。

Catalog.GetTable

MapInfo.Data.Catalog.GetTable 方法返回由 TableAlias 参数引用的 Table 对象。这必须是已经打开的表。如果未找到这种表（或表已随后关闭）则方法返回空。

Catalog.Item (Indexer)

MapInfo.Data.Catalog.Item 属性可用作通过其 Alias 定位 Table 的索引。这在功能上与使用 Catalog.GetTable 方法等价，不过该索引生成的代码可读性更好。Alias 必须指定已打开的表。

VB 示例：

```
Public Shared Sub MapInfo_Data_Catalog2()

    Dim tbl As Table
    For Each tbl In Session.Current.Catalog
        System.Console.WriteLine("Table: " + tbl.Alias)
    Next

End Sub
```

TableEnumerators

表枚举符可以通过各种重载的 EnumerateTables 方法来获取。表枚举符可以与筛选器一同创建。筛选器确定了枚举中实际包括了哪些表，而枚举符只是简单地提供了枚举的机制。用户可以创建自己的表筛选器用在 TableEnumerator 中。也可以通过实现 ITableEnumerator 接口来创建自己的表枚举符。

VB 示例：

```
Public Shared Sub MapInfo_Data_Catalog3(ByVal catalog As Catalog)
    Dim te As ITableEnumerator = _
catalog.EnumerateTables(TableFilterFactory.FilterEditableTables())

    While te.MoveNext()
        Dim tbl As Table = te.Current

    End While
End Sub
```

关闭表

有三种方法可以关闭表。MapInfo.Data.Catalog.CloseAll 关闭所有打开的表，而 Catalog.CloseTable 关闭某一个打开的表。Table 类也有 Close 方法。

打报表

`MapInfo.Data.Table.Pack` 方法从先前标记为要删除的表中删除记录。打报表时，激发表的 `TablePacked` 事件。事件的参数指示表的键是否因打包而更改（删除已删除的记录时发生）。如果 `PackType` 包括 `RemoveDeletedRecords` 且如果表的中间实际有已删除的记录，则仅更改键。如果只有已删除的记录位于表的末尾，则没有任何键发生更改。事件并不表明键已更改。

i 由于 `ResultSet` 表中保存了键集合，而对派生这些键的表进行打包操作极容易影响这些键。如果键发生更改，那么 `ResultSet` 不再有效。

`PackType` 枚举提供了以下选项。

- `PackGeometry` - 指示几何体对象被打包。打包对象操作会尝试删除尽可能多的未使用空间。完全被打包的 `RTree`（用于从空间访问几何体对象的空间索引）由于引起许多不必要的读取会降低性能。要在磁盘空间和处理速度之间取得平衡，打包几何体对象可以继续在 `RTree` 中留下一些未使用的空间。另请注意：打包的 `RTree` 会导致在插入和更新操作时出现轻微的性能降低，因为 `RTree` 更有可能需要扩展。
- `RebuildGeometry` - 重建几何体对象删除了从一系列插入、更新和（或）删除操作生成的未使用空间。与打包几何体对象不同，该选项有意在 `RTree` 索引中留下未使用的空间以改善以后插入和更新操作的性能。
- `PackIndex` - 非空间的索引作为 `B*trees` 维护。这些结构不总是具有填存的内部或分支节点。默认情况下，有意允许在索引的位置放置插入和更新操作，而无需索引的大量调整。由于插入、更新或删除操作的出现，未使用的空间会剧增。打包索引操作完全打包了每个内部和分支节点（除可能的“最后”节点）。该选项尽可能减少了索引所用的磁盘空间，还改善了索引的读取性能。在完全被打包的索引上插入和更新操作的性能会降低。
- `RebuildIndex` - 重建索引不会像 `PackIndex` 选项一样完全打包内部和分支节点。相反，重建索引创建具有未使用空间的索引，该未使用空间有意放置在索引中用于在磁盘空间、读取性能和修改性能之间取得平衡。多次修改操作后，索引会包含大量的未使用空间。该选项重新获得未使用的空间。
- `RemoveDeletedRecords` - 包括 `MapInfo Native` 和 `dBase` 数据源在内的某些数据源当它们被删除时并不物理删除记录。要物理删除已删除的记录，必须指定此选项打报表。记录号通常用作这些数据源类型的记录键。从表删除已删除的记录会使得键成为无效，因为这些键会因打包而更改。
- `CompactDb` - 如果表的数据源是 `Microsoft Access`（`Access` 的 `TableType`），则通过使用 `Pack` 方法并指定该选项，也可以压缩包含表数据的 `MDB` 文件。
- `All` - 这是与 `PackGeometry` | `PackIndex` | `RemoveDeletedRecords` 等价的简洁选项。

侦听表和目录事件

`Table` 公开了应用程序可以订阅的多个事件。它们是：

- `RowInsertedEvent` - 将新行添加到表时发生。
- `RowUpdatedEvent` - 更新表中现有的行时发生。
- `RowDeletedEvent` - 删除表中的行时发生。
- `TablePackedEvent` - 打报表时发生。

- TableCloseRequestEvent - 已经要求关闭表时发生。
- TableIsClosingEvent - 关闭表时发生。
- TableClosedEvent - 关闭表时发生。

Catalog 也公开以下事件。

- TableOpenedEvent - 打开表时发生。
- TableCreatedEvent - 创建新表时发生。
- TableIsClosingEvent - 关闭表时发生。

表元数据 (TableInfo)

MapInfo.Data 命名空间中的 TableInfo 类是包含关于现有表的信息和元数据的抽象基类，包括：

- 列 - 编号、名称、数据类型等。
- 表别名及数据源的描述和路径名。
- 客户端元数据（TAB 文件中 begin_metadata/end_metadata 标记之间的信息）。

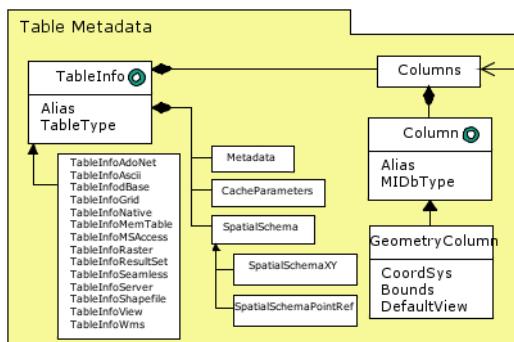
TableInfo 用于打开表并新建表。也可用于检索关于打开的表的信息。

源自于 TableInfo 的类包括提供程序特有的元数据。每一个受 MapXtreme 支持的表类型均有一个 TableInfo 实现。请参阅[数据源](#)。

TableInfo 实例可能是手动构建的，或从 .TAB 文件定义（未打开表）中选择的，如以下所示。

```
TableInfo.CreateFromFile(...)
```

TableInfo 包含的属性可用于启用 Table Services，包括通过空间架构缓存表并使其成为可制图的。
请参阅[运用缓存和使表成为可制图](#)。



MapXtreme 提供 M 和 Z 值的表列元数据支持。该功能在希望了解特殊数据提供方的几何体是否能支持 3D 和测量值而无需计算其单个几何体时很有用。

可从表的 **TableInfo** 属性中访问表的元数据。可从表元数据访问 **GeometryColumn** 查询该表是否支持 M 或 Z 值，以及在范围已知的情况下该表的范围值是多少。有关 M 和 Z 值支持的更多信息，请参阅 [M 和 Z 值支持](#)。

检查 TAB 文件元数据

TAB 文件元数据可以访问和编辑。可从 **Table** 获得 **TableInfo** 类来获得与表结构有关的信息。

以下代码说明了如何获得打开的表的元数据。该代码还说明了几何体列如何用于确定表的坐标系和边界。有关返回 M 和 Z 值的代码示例，请参阅《开发人员参考》中的 **MapInfo.Data.TableInfo**。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfo2()
    ' Get the metadata for an open table
    Dim ti As TableInfo = Session.Current.Catalog("states").TableInfo

    ' Print out some information to the console
    Console.Out.WriteLine("Table Alias={0}, Datasource={1}, _
        Description={2}, Type={3}", _
        ti.Alias, ti.DataSourceName, ti.Description, ti.TableType)

    ' Print out some information about each column
    Dim col As Column
    For Each col In ti.Columns
        Console.Out.WriteLine("Column {0} Type={1} Width={2}", _
            col.Alias, col.DataType, col.Width)
        ' If the column is a geometry column, print csys and bounds.
        If col.DataType = MIDbType.FeatureGeometry Then
            Dim geocol As GeometryColumn = col
            Dim csys As MapInfo.GeomeTry.CoordSys = geocol.CoordSys
            Console.Out.WriteLine("CSys : {0}", csys.MapBasicString)
            Dim dr As MapInfo.GeomeTry.DRect = geocol.Bounds
            Console.Out.WriteLine("Bounds=({0},{1}),({2},{3})", dr.x1, _
                dr.y1, dr.x2, dr.y2)
        End If
    Next

End Sub
```

新建表

以下章节说明了如何创建永久本地表、临时本地表和临时 **MemTable**。

创建新的永久本地表

MapInfo 本地表的 **MapInfo.Data.Table.TableInfo** 属性返回 **TableInfoNative** 的实例。本地表为 **MapInfo** .TAB 文件。该类可用于访问特定于本地表类型的属性。可以创建该类的新实例，并使用该实例构造新表。另请参阅[数据源](#)。

请注意 **ColumnFactory** 类的使用。提供该类的目的是帮助您了解对于不同的数据类型哪些参数是必须的。例如，几何体列需要坐标系。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfoNative()
    Dim ti As TableInfoNative = New TableInfoNative("NewTable")
    ti.TablePath = "c:\data\Capitals.TAB"
    ti.Columns.Add(ColumnFactory.CreateIndexedStringColumn("Capital", _
        25))
    ti.Columns.Add(ColumnFactory.CreateStringColumn("Country", 30))
    ti.Columns.Add(ColumnFactory.CreateDoubleColumn("Pop_Grw_Rt"))

    ' Make the table mappable
    ti.Columns.Add(ColumnFactory.CreateStyleColumn())
    Dim Robinson As CoordSys = _
        Session.Current.CoordSysFactory.CreateFromPrjString("12, _
        62, 7, 0")

    ti.Columns.Add(ColumnFactory.CreateFeatureGeometryColumn(Robinson))
    ' Note we do not need to (nor should we) add a column of type Key.
    ' Every table automatically contains a column named "MI_Key".
    Dim table As Table = Session.Current.Catalog.CreateTable(ti)
End Sub
```

创建临时本地表

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfo3(ByVal conn As MIConnection)
    Dim ti As TableInfoNative = New TableInfoNative("NewTable")
    ti.Temporary = True
    Dim col As Column

    col = New Column
    col.Alias = "FString30"
    col.DataType = MIDbType.String
    col.Indexed = True
    col.Width = 30
    ti.Columns.Add(col)

    col = New Column
    col.Alias = "FInt32"
    col.DataType = MIDbType.Int
    col.Indexed = True
    ti.Columns.Add(col)

    col = New Column
    col.Alias = "FInt16"
    col.DataType = MIDbType.SmallInt
    col.Indexed = True
    ti.Columns.Add(col)
```

```
col = New Column
col.Alias = "FDouble"
col.DataType = MIDbType.Double
ti.Columns.Add(col)

col = New Column
col.Alias = "FDateTime"
col.DataType = MIDbType.Date
ti.Columns.Add(col)

col = New Column
col.Alias = "FBoolean"
col.DataType = MIDbType.Boolean
ti.Columns.Add(col)
' Note we do not need to (nor should we) add a column of type Key.
' Every table automatically contains a column named "MI_Key".
Dim miTable As Table = conn.Catalog.CreateTable(ti)
End Sub
```

创建临时 MemTable

内存表的 MapInfo.Data.Table.TableInfo 属性返回 TableInfoMemTable 的实例。该类可用于访问特定于内存表类型的属性。可以创建该类的新实例，并使用该实例构造新表。

可将诸如 XML 或 GML 之类格式的数据从 Web 服务加载到 Catalog 中并以此方式对其进行使用。可通过 MapXtreme API 将其转换为 MultiPolygon、LineString、Point 或其它 Geometry。MapXtreme 然后将 Geometry 转换为 FeatureCollection，并依次将其保存到 memTable 或本地 TAB 格式。

如果想要使数据在 MapXtreme 中可用但不必用于地图显示，则可以使用此方法。

MapXtreme 支持将 Z 和 M 值读写至 MemTables。MultiCurves 上的 M 值可执行线性参考操作和动态分割。请参阅[第 21 章：线性参考](#)。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfoMemTable()
    Dim ti As TableInfoMemTable = New TableInfoMemTable("NewTable")
    ' Note: The TablePath property does not apply - it can be set but it _
    ' is meaningless.

    ti.Columns.Add(ColumnFactory.CreateIndexedStringColumn("Capital", _
        25))
    ti.Columns.Add(ColumnFactory.CreateStringColumn("Country", 30))
    ti.Columns.Add(ColumnFactory.CreateDoubleColumn("Pop_Grw_Rt"))

    ' Make the table mappable
    ti.Columns.Add(ColumnFactory.CreateStyleColumn())
    Dim Robinson As CoordSys =
        Session.Current.CoordSysFactory.CreateFromPrjString("12, 62, _
        7, 0")
    ti.Columns.Add(ColumnFactory.CreateFeatureGeometryColumn(Robinson))
```

```
' Note we do not need to (nor should we) add a column of type Key.  
' Every table automatically contains a column named "MI_Key".  
    Dim table As Table = Session.Current.Catalog.CreateTable(ti)  
End Sub
```

将表达式列增加到表

使用 `MapInfo.Data.Table.AddColumn` 方法向表中添加表达方式列。这种形式的 `AddColumns` 根据由函数、运算符、重复值和表上其它列组成的表达式，使用 `Columns` 对象创建临时列。`columns` 参数中 `Column` 的所有实例必须具有指定的表达式字符串。

i `TableAddColumns` 不支持以下表类型：`Server`、`View`、`Seamless`、`AdoNet`、`ResultSet` 或 `Drilldown`。`MapXtreme` 检查表并在遇到上述表类型时抛出异常。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableAddColumns(ByVal miTable As Table)  
    Dim NewCols As Columns = New Columns  
    NewCols.Add(New Column("PopDensity1990", "Pop_1990 / _  
        MI_Area(Obj, 'sq mi', 'Spherical')"))  
    NewCols.Add(New Column("PopDensity2000", "Pop_2000 / _  
        MI_Area(Obj, 'sq mi', 'Spherical')"))  
    miTable.AddColumn(NewCols)  
End Sub
```

表达式字符串 “`Pop_1990 / MI_Area(Obj, 'sq mi', 'Spherical')`” 表示放置在临时列中的派生信息。其表示“每个人口记录除以平方英里为单位的面积得出人口密度”。表达式中的 SQL 函数 `MI_Area()` 将从记录的几何体派生面积。

使用 `AddColumns` 方法可改进桌面应用程序的性能，只需执行一次连接即可，以后访问时不再需要进行连接（如在视图中）。

详细信息和代码示例，请参阅开发人员参考帮助系统中的 `MapInfo.Data.Table.AddColumn` 类。

有关创建表达式的详细信息请参阅[第 10 章：创建表达式](#)。

数据源

下表列出了 `MapXtreme` 支持的数据源。某特定的数据提供程序（源自于 `MapInfo.Data.TableInfo` 的 `TableInfo` 类）可访问每种类型的数据源。有关每种数据类型的简要概述，请参阅[支持的表类型](#)。

数据源	类
Native (<code>MapInfo.TAB</code>)	<code>TableInfoNative</code>
dBase	<code>TableInfodBase</code>

数据源	类
MS Access	TableInfoMSAccess
ASCII	TableInfoAscii
RSBMS Server	TableInfoServer
ESRI Shapefile	TableInfoShapefile
Seamless	TableInfoSeamless
Raster	TableInfoRaster
Grid	TableInfoGrid
WMS	TableInfoWMS
ADONET	TableInfoAdoNet
MemTable	TableInfoMemTable
View	TableInfoView
ResultSet	TableInfoResultSet
TileServer	TableInfoTileServer

选取正确的数据源

选取正确的数据源可大大提升应用程序的性能。在某些情况下您没有选择，例如本地 **MapInfo** 文件 (.TAB)，而在其他情况下可能会有多种选择。大多数情况下，均可将受支持的数据提供程序用于数据源中。在无法通过任一种这些方式访问数据的情况下，则可使用 **MapInfo ADO.NET** 数据提供程序。该数据提供程序接口与 Catalog 用于数据检索的相同。

每个数据源均有某些性能特征。本地表提供最佳的访问和地图绘制时间。数据存储在本地系统中，为当前操作而优化。其他基于文件的表类型性能很好，这取决于当前硬件和文件大小。

访问数据的方法

MapXtreme 提供了多种方法将数据加载到 Catalog 中：

- 直接访问数据源
- 通过 ADO.NET 数据提供程序 (TableInfoAdoNet) 进行访问
- 来自第三方 web 服务的 XML/GML

访问数据的最好方法是使用数据驻留位置特有的任一个 TableInfo 类直接将其打开。

使用第二种方法 (TableInfoAdoNet) 访问不被内部支持但有 ADO.NET 提供程序的数据。

第三种方法使开发人员可以将数据集成到可能与返回 XML 或 GML 的 HTTP 服务交互的 Catalog。

直接访问数据源

MapXtreme 为访问存储在诸如 SQL Server 和 Oracle 之类基于文件的表格式和 RDBMS 服务器中的数据提供了本地支持。对于基于文件的访问，则可提供相应的 TableInfo 实例（TableInfoNative、TableInfodBBase、TableInfoMSAssess 和 TableInfoAscii）中的路径和文件名。

对于对存储在 RDBMS 服务器上的数据进行的直接访问，可使用 TableInfoServer 类来定义要在远程表上执行的连接串和 SQL 语句。MapXtreme 内部使用 ODBC 或 OCI 访问远程数据库。

TableInfoServer 将使用远程服务器上定义的任何空间特征打开服务器的连接，查询表的元数据，并创建适当的表定义。这往往是使用远程数据的最好执行方法。在内部，MapXtreme 仅可以访问执行当前操作的数据。地图绘制过程中，MapXtreme 会构造一个查询，此查询仅返回几何列而不返回数据列。这会最小化网络通讯量。如果高速缓存已启用，那么只有第一次访问才会出现这个问题，因为所有后续请求将来自该缓存。请参阅[第 11 章：从 DBMS 访问数据](#)。

通过 ADO.NET 数据提供程序进行访问

第二种数据访问方法是使用 ADO.NET 数据提供程序。这需要数据检索的 ADO.NET 类定义。只有不可制图的表才可作为 AdoNet 表提供。不可制图的表不包含关于数据的几何体信息。但是，通过将 SpatialSchema 应用到表定义中，可将从 ADO.NET 提供程序检索的表变成可制图的。用此种方法，无论何时用户请求数据，MapXtreme DataAccess 引擎均将调用 ADO.NET 数据提供程序。这会是一种较慢的访问数据的方法。但是，当与高速缓存共同使用时就会很好地执行。请参阅[使用 ADO.NET 数据提供方](#)。

第三方 Web 服务中的数据

MapXtreme 可以将 Web 服务 XML 或 GML 输出集成到 Catalog 中以便用于 MapXtreme 桌面或 web 应用程序中。可通过 MapXtreme API 将数据加载进 Catalog 中，并将其转换为 MultiPolygon、LineString、Point 或其他 Geometry。MapXtreme 然后将 Geometry 转换为 FeatureCollection，并依次将其保存到 memTable 或本地 TAB 格式。

如果想要使数据在 MapXtreme 中可用但不必用于地图显示，则可以使用此方法。

Data Readers、MemTables 和 Result Sets

访问数据返回数据读取器或结果集的方法。数据读取器允许以顺序方式访问，不存储数据的副本。除非在缓存数据源的情况下，否则均从数据源检索数据。结果集是键的集合。这些键可以用于访问回原始表，并不创建该数据的副本。

MemTable 也可以用于将数据从各种源中存储到表中。此种类型的表以内存数组与临时磁盘存储方式存储数据。添加数据时，MemTable 制作一个数据的副本，且没有键或指针指向原始表。这些对于地图和容器的临时图层来讲是非常有用的，可以返回进程值如地理编码或路由结果。MemTable 访问和地图渲染性能与本地表相同。

当需要访问定义的一组行和需要从源获取数据时，结果集会是一个非常有用的工具。如果源数据在会话过程中可以更改，如果数据源支持同步访问，则此方法可用于查看结果。因为 **MemTable** 是数据的副本，它们是数据行的静态集，将不反应用于原始数据源的更改。

使用 ADO.NET 数据提供方

无法通过特定 **TableInfo** 数据源对其进行访问的数据，可使用 **TableInfoAdoNet**。ADO.NET 表可以是以下两种形式之一：**DataTable**（保存在内存中的单一表中的行集合，允许读写访问）；或 **IDbCommand**（在生成只读数据的数据源处执行的 SQL 语句，动态数据）。

访问 **DataTable** 中的数据

使用 **DataTable** 时，**Catalog** 实际保存到对 **DataTable** 的引用上，其中 **DataTable** 提供到对 **Catalog.OpenTable** 的调用（使用 **TableInfoAdoNet** 类）。通过发布 **Insert**、**Update** 和（或）**Delete** 命令，使用 **MapInfo ADO.NET Data Provider** 可以编辑 **DataTables**。应用程序也可以继续直接访问 **DataTable**。不过要注意，**Catalog** 具有到表的引用时表的结构不应该更改。还需注意，**MapInfo Data Provider** 之外的其它数据发生更改时（例如不使用 **MICommand** 发布 **Insert**、**Update** 或 **Delete** 命令时）不会激发插入、更新或删除表的事件。

DataTable 几乎包含了 **Catalog** 定义表的全部信息。不过对于字符串列，**Catalog** 需要将一个长度分配到该字段。构造临时索引、用于合计的临时表等时将使用该长度。对于这种类型的操作，一定要保证获得的字符串长度正确。 **DataColumn** 具有 **MaxLength** 属性，该属性设置用于指示列可以保存的最大字符串长度。如果不设置，则该值默认为 -1，这种情况下使用的长度值为 254。检查 **MaxLength** 属性之前，“目录”查看 **DataColumn** 是否具有在其 **ExtendedProperties** 集合中定义的属性，名称为 “**StringWidth**”。如果找到该属性，则将该属性的值用作列的宽度。

以下示例说明了如何创建 **MapInfo Table**，该表的数据存储在 **DataTable** 中。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfoAdoNet(ByVal connection As _
    MIConnection)
    ' Create a new DataTable.
    Dim dt As DataTable = New DataTable("CityData")
    Dim dc As DataColumn
    dc = dt.Columns.Add("City", Type.GetType("string"))
    dc.MaxLength = 30
    dc = dt.Columns.Add("Country", Type.GetType("string"))
    dc.MaxLength = 30
    dc = dt.Columns.Add("Continent", Type.GetType("string"))
    dc.MaxLength = 30
    dc = dt.Columns.Add("Population", Type.GetType("string"))

    ' Populate the DataTable...
    dt.Rows.Add(New Object() {"Madrid", "Spain", "Europe", 1500000})
    dt.Rows.Add(New Object() {"Stockholm", "Sweden", "Europe", 985000})

    ' Now open a MapInfo Table which accesses this DataTable
```

```
Dim ti As TableInfoAdoNet = New TableInfoAdoNet("Cities")
ti.ReadOnly = False
ti.DataTable = dt
Dim table As Table = connection.Catalog.OpenTable(ti)
End Sub
```

保存和恢复 ADO.NET 对象

通过执行下述步骤，可在 MapXtreme 中完成 ADO.NET 表保存和恢复。由于对恢复基本 System.Data.DataTable 的限制，ADO.NET 表的显式序列化/反序列化不受支持。ADO.NET 表所属的 Catalog 是通过使用 MapInfo.Data.TableInfoAdoNet 对象创建的。

序列化/反序列化基于 ADO.NET 的 MapInfo 表的正确方法可在 SaveState 方法中找到。序列化引用了 ADO.NET 表（即 ViewTables 或 joins）的所有表，然后关闭 ADO.NET 表。在 RestoreState 方法中，用同样的名称重新创建 ADO.NET MapInfo 表，然后反序列化任何附属的 MapInfo 表。顺序相当重要，因为在创建 ADO.NET 表之前必须恢复任何其他表。

在执行手动状态管理的 MapXtreme Web 应用程序中，请遵循以下步骤来保存和恢复客户端请求之间的 ADO.NET 表。

 以下步骤特别参考了创建自 DataTable 的 ADO.NET 表。

1. MapInfo.WebControls.StateManager.SaveState 方法：
 - a. 将 ADO.NET DataTable 或 DataSet 置于 HttpSession 实例。
 - b. 使用 ManualSerializer.SaveMapXtremeObjectIntoHttpSession 保存任何从属于 ADO.NET 表的 MapInfo.Data.TableInfoView 或 MapInfo.Data.TableInfoResultSet 表。
 - c. 关闭 Catalog 中的 ADO.NET 表。
2. MapInfo.WebControls.StateManager.RestoreState 方法：
 - a. 根据从 HttpSession 实例检索的 DataTable 新建 TableInfoAdoNet 对象。
 - b. 根据与原始 ADO.NET 表具有相同名称的 TableInfoAdoNet 对象在目录中打开新的 ADO.NET 表。
 - c. 恢复任何基于使用 ManualSerializer.RestoreMapXtremeObjectFromHttpSession 的 ADO.NET 表的 MapInfo.Data.TableInfoView 或 MapInfo.Data.TableInfoResultSet 表。

记住：保存和恢复的顺序对所有的表和它们的依赖关系的适当创建非常重要。有关序列化的信息请参阅第 6 章：了解状态管理。

使用 IDbCommand 访问数据

第二种形式的 ADO.NET 表基于 ADO.NET 中连接的对象类型：以这种方式构造的 MapInfo Tables 是只读的。通过传递到已配置的 Catalog 和 IDbCommand 对象，创建这些类型的表来返回构成表的所有数据。如果初始化时创建表（通过调用 Catalog.OpenTable），则在 IDbCommand 上调用 ExecuteReader。生成的数据读取器用于确定列及其数据类型。所有后续光标请求（不是检索特定记录的光标 - 名为键获取）也调用 ExecuteReader 来获取满足光标的 data。注意，这种操作效率很低。实际情况中最好使用其它的表类型之一来访问数据。

因为基于命令形式的 ADO.NET 表设计用于一般接口，无需任何与这些接口的特殊实现有关的特定信息，所以该表也不假定 IDbCommand.CommandText 是任何形式的标准 SQL。实际上，它根本不可能为 SQL。该表类型不访问、分析或修改 CommandText。这意味着该表类型不具有相应的机制来了解结果中的哪些列设计唯一、非空的键值。对于这种类型的表，需要告知表哪些列构成键。这可以通过将 KeyType 指定为 Explicit 并设置 KeyColumns 属性来实现。

MapInfo Data Provider 内有很多操作需要通过键（也称为键获取）检索特定的记录。Select 语句带有格式为 MI_Key = '5' 的 where 子句，这是一个需要检索记录的简单示例，该记录的 MI_Key 列可以通过字符串文字 '5' 表示。映射选择集、标注和在 MIScrollableReader 中滚动（这种情况下读取器可以通过键值的列表滚动）时常用键检索。MapInfo 表取决于通过键值有效获取记录的能力。就是因为基于命令形式的 ADO.NET 表不读取、分析或修改定义表（“Sequential” IDbCommand）的 IDbCommand 对象的 CommandText，所以它不能修改 IDbCommand 对象来获取特定的记录。因此，要达到该目的必须提供第二个 IDbCommand 对象。“FetchByKey” IDbCommand 对象必须满足以下要求：

- 在该命令对象上调用 ExecuteReader 时，生成的数据读取器必须具有与连续命令对象相同的列，且顺序也相同。
- FetchByKeyCommand 必须包含 Parameters 集合，且为键的每个成员包含一个参数。例如，如果 TableInfo.KeyColumns 指定了由 "city" 和 "state" 列组成的键，则 FetchByKeyCommand 必须包含两个参数对象。为第一个参数对象分配的值表示在 TableInfo.KeyColumns 集合中指定的第一列（例如，值 "city"），为第二个参数对象分配的值表示在 TableInfo.KeyColumns 集合中指定的第二列（例如，值 "state"）等等。在 FetchByKeyCommand 上调用 ExecuteReader 时，读取器必须返回表示指定键的记录。

以下示例说明了如何创建 MapInfo Table，该表通过连接命令对象的 ADO.NET 访问数据。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfoAdoNet2(ByVal connection _
    As MICConnection)
    Dim ti As TableInfoAdoNet = New TableInfoAdoNet("EuropeanCities")
    Dim _conn As OleDbConnection = New _
        OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data _
            Source=C:\Data\EuropeCities.mdb")
    Dim selectQuery As String = "SELECT City, Country, Continent, _
        Pop_1994 FROM EuropeCities"
    Dim _OleDbCommand As OleDbCommand = New OleDbCommand(selectQuery)
    _OleDbCommand.Connection = _conn

    selectQuery = selectQuery + " where City = @City AND _
        Country = @Country"
    Dim _OleDbKeyCommand As OleDbCommand = New _
        OleDbCommand(selectQuery)
    _OleDbKeyCommand.Parameters.Add("@City", OleDbType.Char)
    _OleDbKeyCommand.Parameters.Add("@Country", OleDbType.Char)
    _OleDbKeyCommand.Connection = _conn

    ' The MapInfo Table will Open/Close the connection as necessary.
    ' If this is expensive the application could open the connection
```

```
' before opening the table and closing the connection after the
' table is closed.
ti.SequentialCommand = _OleDbCommand
ti.FetchByKeyCommand = _OleDbKeyCommand

' Tell the table which column(s) constitute a key - for this table
' it is a compound key consisting of values from the City and County
' columns.
Dim sc As StringCollection = New StringCollection
sc.Add("City")
sc.Add("Country")
ti.KeyColumns = sc
ti.KeyType = KeyType.Explicit

' Ask the Catalog to open the table.
Dim tbl As Table = connection.Catalog.OpenTable(ti)

' Now the MICommand object may be used to select data from the table
' (by the name EuropeanCities since that is the alias we assigned to
' it). The data in this table may be joined with any other table and
' it may be used as source data in a call to AddColumns to populate
' temporary columns with data from this table.
End Sub
```

数据绑定 (Data Binding)

数据绑定是将数据从数据源带到 MapXtreme 中的过程。使用 TableInfoAdoNet 打开用作表的 ADO.NET DataTable，即可将外部（ADO.NET 和其他旧的数据源）数据绑定到 MapInfo.Data.Table 中。表然后可与另一个表连接、与自身连接或使用 Table.AddColumn 将列绑定到第二个表。

要将表与其自身连接，请参照以下示例：

```
Select ... From T as A, T as B Where A.X = B.Y
```

如果应用程序具有存储在 DataTable 中的数据，或具有可以通过 ADO.NET 数据提供方访问的数据，则这些数据可以提交到 Catalog 并视为 MapInfo 表。如果数据不能通过其它表类型之一来访问，就需要使用这种功能。

例如，如果数据存储在 dBase 文件（Microsoft Access 表）中，或可以通过 ODBC 或 Oracle 的 OCI 接口对其进行访问，则建议使用这些 TableInfo 类型来访问该等数据。如果数据不能通过其中一个表类型来访问，但可以加载到 DataTable 中或通过某些 ADO.NET Data Provider（实现 Command、Parameter 和 DataReader 对象类型）可以访问，则这些数据仍然可以通过 Catalog 来访问。

应用程序可能需要数据可以像 MapInfo 本地表一样方便可用，以便可以执行查询，从而将数据与其他 MapInfo 表数据连接。对于 Catalog 可能也需如此，以便在对 Table.AddColumn 的调用中可以将 Catalog 用作源数据。

代码示例：如何从 Oracle 表连接数据

```
public Shared Sub MapInfo_Data_TableAddColumns5(ByVal map As Map)
    Dim Connection As MapInfo.Data.MIConnection = New _
        MapInfo.Data.MIConnection
    Connection.Open()

    'Add the USA table to the map
    map.Load(New _
        MapInfo.Mapping.MapTableLoader("C:\\maps\\usa.TAB"))

    Dim lyr As MapInfo.Mapping.FeatureLayer = map.Layers("usa")

    ' Open the table from Oracle

    Dim ti As TableInfoServer = New TableInfoServer("StateCapXY", _
        "SRVR=tempest;UID=tn;PWD=tn", "Select * from usa_caps", _
        MapInfo.Data.ServerToolkit.Oci)
    Dim StateCapXY As Table = Connection.Catalog.OpenTable(ti)

    ' Add the Oracle columns to the USA table
    Dim states As Table = Connection.Catalog.GetTable("usa")
    states.AddColumn(Nothing, MapInfo.Data.BindType.Static, _
        StateCapXY, "state", MapInfo.Data.Operator.Equal, "state")

    'Create a ranged theme on the USA layer using a field
    'from the Oracle table
    Dim thm As MapInfo.Mapping.Thematics.RangedTheme = New _
        MapInfo.Mapping.Thematics.RangedTheme(lyr, _
        "pop_1990", "popusa", 4, _
        MapInfo.Mapping.Thematics.DistributionMethod.EqualCount_
        PerRange)
    lyr.Modifiers.Insert(0, thm)

    'Create a legend to appear on the map
    Dim legend As MapInfo.Mapping.Legends.Legend = _
        map.Legends.CreateLegend(New Size(5, 5))
    legend.Border = True
    Dim frame As MapInfo.Mapping.Legends.ThemeLegendFrame = _
        MapInfo.Mapping.Legends.LegendFrameFactory.CreateTheme_
        LegendFrame(_"Pop", "pop", thm)
    legend.Frames.Append(frame)
    frame.Title = "pop"
    map.Adornments.Append(legend)
End Sub
```

视图表

视图是按指定的选择语句将一个或多个表中的信息互为关联的一种方法。Catalog 可用于根据任何表定义创建视图。视图表的特征如下：

- 数据未复制。
- 访问视图时始终要访问其基表。

- 视图是带有名称（别名）的 SQL Select 语句。
- 查询可能是 joins（组成复合键）。
- 视图中的成员条件是活动的。
- 异常：合计缓存数据的视图。数据更改的事件会触发重新计算。
- 可在工作空间中进行序列化和永久化。

有关详细信息和代码示例，请参阅开发人员参考帮助系统中的 `MapInfo.Data.TableInfoView` 类。

结果集

`ResultSets` 与使用 MapInfo SQL 选择语句定义的且具有相关名称（别名）的视图表相似。但是，`ResultSets` 具有固定的记录成员关系，该成员关系基于在创建结果集时对 `where` 子句（如有）的评估。对 `ResultSet` 中数据进行的任何访问始终反映了源表中的数据。然而，对源数据的更改将不会引起 `ResultSet` 根据原始 `where` 子句添加/删除记录。`ResultSets` 内部管理一组键。

`ResultSets` 通常是轻量级和临时的。结果集的一些特征是：

- 数据未复制。
- 访问结果集始终将访问其基表。
- `ResultSet` 是已排序的键列表，列定义和名称的集合。
- `ResultSet` 中的成员关系是固定的。
- 异常：合计缓存数据的 `ResultSets`。数据更改的事件会触发重新计算。
- 可在工作空间中序列化但不可永久化。
- `ResultSets` 易于受“删除”和“打包”操作影响。

详细信息请参阅开发人员参考帮助系统中的 `MapInfo.Data.TableInfoResultSet` 类。

源行

源行表示 `Table.AddColumn` 所涉及的表记录之间的匹配。在表中增加临时列时，数据源的多个记录可以合计，对目的地表（也称为 bind 表）中的每个记录都计算一个值。`MapInfo.Data.SourceRows` 类是 `SourceRows` 的集合，标识了合计在一起的源自于数据源的记录。

仅当 `BindType` 属性为 `DynamicCopy` 时 `SourceRows` 才存在，这表明对源数据作出的更改会自动传播到临时列。



`Table.AddColumn` 不支持以下表类型：`Server`、`View`、`Seamless`、`AdoNet`、`ResultSet` 或 `Drilldown`。

另请参阅[将表达式列增加到表](#)。

GeoDictionary

GeoDictionary 维护可以将地图实体匹配到某些信息的信息。GeoDictionaries 类是 GeoDictionary 对象的集合。MapInfo.Data.GeoDictionary 命名空间通过成为 GeoDictionary 文件的可编程表示为数据自动绑定提供支持。GeoDictionary 文件包含与表有关的信息（仅 TAB 文件）。GeoDictionary 用于自动确定应该将应用程序绑定到的表。GeoDictionary 保留在文件中（通常为 GeoDict.DCT）并通过使用 GeoDictionaryManager 实用程序得以维护（请参阅[第 25 章：使用 GeoDictionary 管理器](#)）。

使用 GeoDictionary 的 AutoMatching

Data.GeoDictionary 命名空间中的 MatchResolver.AutoMatch 方法初始化 AutoMatching 进程。它不调用 AddColumns，即不执行绑定。执行自动绑定需要对 BindColumn 的后续调用或对 AutoMatchAndBind 的直接调用。

Automatching 可能会遇到模棱两可的情况。这些情况包括：

- 在用户数据中检测到多个源列
- 检测到匹配源列的多个表/图层
- 匹配的表/图层有多个 geosets/工作空间

在解决模棱两可的匹配过程中 GeoDictionary 是与 MatchResolver 对象通信。它提供了匹配算法。基类选择第一个或匹配百分比最高的项目。该类未封闭，客户端应用程序可以从中派生出其自己的类并重载其行为。

使表成为可制图

表可以是制图的（包含 GeometryColumn）或不可制图的（无空间属性数据）。可将可制图的表作为地图中图层添加到 MapXtreme 应用程序中。在为不可制图表（如客户数据）创建 GeometryColumn 时，不可制图的表也变得可以制图。MapXtreme 提供了用于完成这项功能的空间构架。

空间构架是服务，可以应用到表来增强表的空间能力。有两种类型的空间构架：XY 和 PointRef。具有表示为 X 和 Y 值（如经度和纬度）的属性列的不可制图表使用 SpatialSchemaXY，具有可用于引用可制图表中记录的属性列的表则使用 SpatialSchemaPointRef。

SpatialSchemaXY

SpatialSchemaXY 使用表中每个记录的 X 和 Y 值来构造点对象并将其存储在称为 MI_Geometry 的临时列中。该空间唯一可应用到除 Seamless、Views 和 ResultSet 之外任意数据源的表。

有了 GeometryColumn，表就可显示为地图图层并用于空间分析。

SpatialSchemaXY 的特征如下：

- Geometry 列可编辑。
- 自动编辑 Geometry 会更改 X 和 Y 值。
- 可以在表中定义每个点的样式。
- 可以将空间信息存储为 TAB 文件，并像打开任何其他表一样将其打开。

该空间唯一可以用于没有 MapCatalog 的传统服务器 XY 数据。（使用 MapCatalog 可以改善 RDBMS 的性能，因为有更多的工作在服务器上完成 请参阅 [MapInfo_MapCatalog](#)。）

MI_Geometry 是临时列，除非您使用 TableInfo.WriteToTab 方法明确写出 TAB 文件。打开表时将自动重新生成唯一。

VB 示例：

```
Public Shared Sub MapInfo_Data_SpatialSchemaXY()
    Dim ti As TableInfo = _
        TableInfo.CreateFromFile("c:\data\customers.TAB")
        ' a non-mappable table
    Dim xy As SpatialSchemaXY = New SpatialSchemaXY
    xy.XColumn = "Xcoord"
    xy.YColumn = "Ycoord"
    xy.NullPoint = "0.0, 0.0"
        ' Any customer at 0,0 means we don't know their location.
    xy.StyleType = StyleType.None
    xy.CoordSys = _
        Session.Current.CoordSysFactory.CreateLongLat(DatumID.WGS84)
    ti.SpatialSchema = xy
        ' Now set the spatial schema information before
        ' opening the table.
    Dim table As Table = Session.Current.Catalog.OpenTable(ti)
End Sub

Public Shared Sub MapInfo_Data_TableInfoNative2(ByVal ti As _
TableInfoNative)
    ti.WriteTabFile()
End Sub
```

SpatialSchemaPointRef

该空间唯一可使用表的数据值创建 Point 几何体对象，只需将该值与可制图表中的等值相匹配。

例如，如果客户表包含带有邮政编码的地址，则可能将客户记录绑定至邮政编号参考表中的空间点。

PSpatialSchemaPointRef 实际上是两个表之间的连接，一个包含数据，另一个包含连接列和对象列。连接列包含的值与不可制图的表中数据列的值相同，例如，邮编编码。应用 SpatialSchemaPointRef 后，就会使表包含适用于原先为非空间的记录的空间几何体列。该几何体列的特征如下：

- 该数据表可与几何体表中的多个记录匹配。发生此情况时，相似行会合并到 MultiPoint 几何体中。
- 该几何体是其他表中几何体的质心。

SpatialSchemaPointRef 的特征如下：

- 临时 Geometry 列为只读。
- 对引用表的值作出的任何编辑均会改变数据表中的 Geometry 值。
- 可将 SpatialSchemaPointRef 应用于除 Seamless、View 和 ResultSet 之外的任何数据源。
- 可以在表中定义每个点的样式。
- 可以将表信息存储为 TAB 文件，并象打开任何其他表一样将其打开。

有关详细信息和代码示例，请参阅开发人员参考帮助系统中的 `MapInfo.Data.SpatialSchemaPointRef` 类。

VB 示例：

```
Public Shared Sub MapInfo_Data_SpatialSchemaPointRef(ByVal _
    map As _Map)

    ' a non-mappable table
    Dim ti As TableInfo = _
        TableInfo.CreateFromFile("c:\data\customers.TAB")
    Dim pr As SpatialSchemaPointRef = New SpatialSchemaPointRef
    pr.CoordSys = map.GetDisplayCoordSys()
    pr.StyleType = StyleType.None
    pr.RefTable = "us_zips"

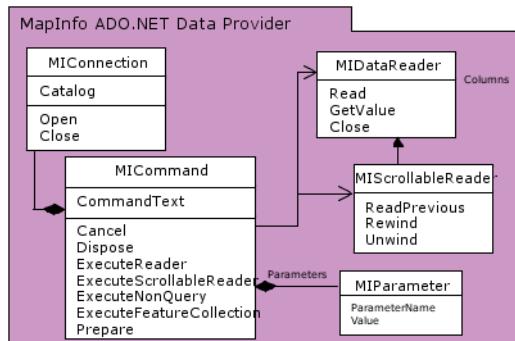
    ' the column in RefTable which will match the MatchColumn in my data
    pr.RefColumn = "zipcode"

    ' a column in the Customer table
    pr.MatchColumn = "zip"
    pr.RefTableLocation = "c:\data\us_zips.tab"

    ' Now set the spatial schema information before opening the table.
    ti.SpatialSchema = pr
    Dim table As Table = Session.Current.Catalog.OpenTable(ti)
End Sub
```

MapInfo ADO.NET 数据提供方

MapXtreme 提供了发布 SQL 命令的机制，该命令使用 ADO.NET 从表返回记录集。MapInfo ADO.NET 数据提供程序是以此方式访问 .NET 应用程序中数据的一种机制。有关使用 Catalog 上 Feature 类和 SearchInfo 方法的选项，请参阅 [Features](#) 和 [Feature 集合](#)。



以下章节介绍了通过 MapInfo ADO.NET 数据提供方访问数据的主要接口和类。

- [MIConnection](#)
- [MICommand](#)
- [MIDataReader](#)
- [MapInfo SQL](#)

MIConnection

MIConnection 表示到 Catalog 的连接。连接提供了发布 SQL 命令和获取结果的起点。虽然大多数数据提供程序连接都允许用户对现有的表（或唯一对象）立即开始发布查询或其他命令，但 MapInfo Data Provider 最初并没有表可用。表要在打开或创建之后才能访问。打开后，可以将名称（别名）与表关联，该名称在查询引擎中解析标识符时使用。

在 MapInfo Data Provider 中，连接是不进入连接池的，创建新的连接也不需要连接字符串。

MapInfo.Engine.Session 类创建并初始化 Catalog，可通过 Session.Current.Catalog 属性来访问该 Catalog。MIConnection.Open 方法用 Session.Current.Catalog 属性来获取对 Catalog 的引用，MIConnection.Close 方法将对 Catalog 的内部引用设置为空。

VB 示例：

```
Public Shared Sub MapInfo_Data_MIConnection()
    Dim connection As MIConnection = New MIConnection
    Dim command As MICommand = connection.CreateCommand()
    command.CommandText = "Select * From States Where Pop > 1000000"

    connection.Open()
    Dim reader As MIDataReader = command.ExecuteReader()
```

```
Dim i As Integer, n As Integer = reader.FieldCount
For i = 0 To n - 1 Step i + 1
    Console.Out.WriteLine("{0}\t", reader.GetName(i))
Next
Console.Out.WriteLine()
While reader.Read()
    For i = 0 To n - 1 Step i + 1
        Dim o As Object = reader.GetValue(i)
        If o Is DBNull.Value Then
            Console.WriteLine("null\t")
        Else
            Console.WriteLine("{0}\t", o.ToString())
        End If
    Next
    Console.Out.WriteLine()
End While
reader.Close()
command.Dispose()
connection.Close()
End Sub
```

MICommand

MICommand 提供了必要的接口来在 MapInfo 数据提供方上执行 SQL 命令。MICommand 创建 MIDataReader 和 MIScrollableReader 实例，分别通过 ExecuteReader 和 ExecuteScrollableReader 方法来获取数据。

支持的命令

MICommand 识别的命令为：

选择

```
SELECT < select_list >
    FROM { < table_source > } [ ,...n ]
    [ WHERE < search_condition > ]
    [ GROUP BY expression [ ,...n ] ]
    [ ORDER BY {expression | column_position [ ASC | DESC ] } [ ,...n ] ]

< select_list > ::=*
{
    *
    | { table_name | table_alias }.*
    | { expression } [ [ AS ] column_alias ]
} [ ,...n ]

< table_source > ::=
    table_name [ [ AS ] table_alias ]
```

Insert

```
INSERT [INTO] { table_name } [ ( column_list ) ]
```

```
{ VALUES ({expression | NULL} [, ...n]) | query_specification
```

Update

```
UPDATE { table_name }
    SET {{ column_name } = { expression | NULL } [, ...n]
    [WHERE < search_condition > ]
```

Delete

```
DELETE [FROM] { table_name } [ WHERE < search_condition > ]
```

```
< search_condition > ::=  
{ [ NOT ] < predicate > | ( < search_condition > ) }  
[ { AND | OR } [ NOT ] { < predicate > |  
( < search_condition > ) } [, ...n] ]
```

```
< predicate > ::=  
{  
    expression [ { = | < > | != | > | >= | < | <= } expression ]  
    | string_expression [ NOT ] LIKE string_expression [ ESCAPE  
        'escape_character' ]  
    | expression [ NOT ] BETWEEN expression AND expression  
    | expression IS [ NOT ] NULL  
}
```

expression

是列名、伪列、列的别名、常量、函数，或是用运算符连接的以上项目的任意组合。列名和伪列可以是以表名或表别名后接点（“.”）字符为前缀。

group_by_expression

是选择列表的列或选择列表表达式的匹配副本的参考，别名 - 基于 1 的数字，指示列的位置，或 *n* 是表示列的数字的 *coln* 位置。

order_by_expression

是选择列表的列或选择列表表达式的匹配副本的参考，别名 - 基于 1 的数字，指示列的位置，或 *n* 是表示列的数字的 *coln* 位置。

有关表达式，使用表达式的情况以及创建表达式的方式，请参阅 [Features 和 Feature 集合](#)。

ExecuteFeatureCollection

MICommand 类中的 ExecuteFeatureCollection 方法是 MapInfo ADO.NET 数据提供方和 Feature 对象模型之间的桥梁。该方法在数据源连接上执行命令文本（SQL 语句）并生成 IResultSetFeatureCollection。有关 Feature 模型的讨论请参阅 [Features 和 Feature 集合](#)。

MIDataReader

对从执行 SQL Select 语句返回的数据，MIDataReader 提供了只前进和只读的访问。要创建 MIDataReader，必须调用 MICommand 对象的 ExecuteReader 方法，而不是直接使用构造函数。

MapInfo Data Provider 允许在同一个连接中使用多个 `MIDataReader` 实例。但是，如果被访问的 Table 驻留于 Microsoft SQL Server 数据库，则一次只能打开一个 `MIDataReader`。

`IsClosed` 和 `RecordsAffected` 是 `MIDataReader` 关闭后可以调用的唯一属性。尽管 `MIDataReader` 存在时都可访问 `RecordsAffected` 属性，但为了保证返回准确的值始终应在返回 `RecordsAffected` 值前调用 `Close`。

使用 `MIDataReader` 时必须显式调用 `Close` 方法。

通过 `IEnumerator` 或 `IFeatureEnumerator` 接口来访问 `DataReader` 时，如果 `MoveNext()` 返回 `false` 则会自动调用 `Close()`。`DataReader` 中只能使用一个枚举符。

i 为了得到最佳性能，`MIDataReader` 会试图避免创建不必要的对象或复制不必要的数据。这样，对 `GetValue` 等方法的多重调用可能会返回对同一对象的引用。修改 `GetValue` 等方法所返回对象的基础值时要小心。

`MIDataReader` 提供了从 MapInfo 数据提供程序读取行的前向数据流的方法。该光标类型是访问一系列行的最佳执行方法，因为几乎没有安装和开销。

可滚动的数据读取器

`MIScrollableReader` 从 `MIDataReader` 派生并提供前向和反向读取。`MIScrollableReader` 可用的某些选项包括：

- `ReadPrevious`
- `Rewind`
- `Unwind`
- `ReadTop`
- `ReadBottom`
- `AtTop / AtBottom`

i 创建 `MIScrollableReader` 比创建 `MIDataReader` 更消耗资源。这是最昂贵的光标。因为需要安装和必要的额外资源来跟踪记录顺序以允许滚动。仅当需要滚动查看记录集时才使用该光标。

MapInfo SQL

MapInfo SQL 语言允许您在 `MapXtreme` 应用程序中添加功能强大的分析处理。`MapXtreme` 通过用于访问数据的 MapInfo ADO.NET 数据提供方（特别是 `MICommand` 对象）向用户公开了 SQL 处理。此外，表达式也用于标注、主题、图例、`AddColumns`、图元搜索和选择集处理。

MapInfo SQL 根据 SQL-3 规范标准化。例如，您会发现：

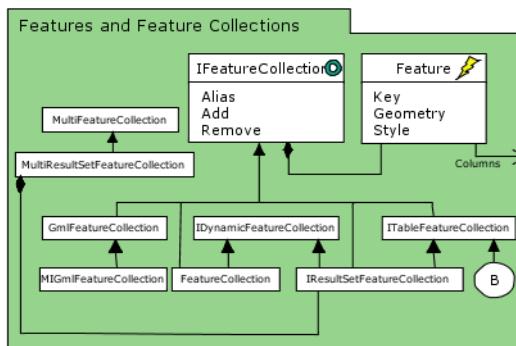
- 字符串常数包括在单引号内
- 标识符可以包括在双引号内

- Select 与 Selection 没有关系

包括 MapInfo SQL 语言代码示例的完整引用可在 MapInfo SQL Reference（可从 Visual Studio 帮助系统直接查看）中找到。

Features 和 Feature 集合

MapXtreme 中的 Feature 类对象模型提供了不基于 SQL 的方法来访问和操控数据。本节涵盖了 Feature 类和 IFeatureCollection 接口。处理图元时的主要任务是可以使用查询定义对象搜索图元。



Feature

图元由它们的几何体、样式、数据源、键和属性描述。通常，图元为表中的行。图元的几何体是 FeatureGeometry 对象。FeatureGeometries 可以包含给定的区域 (MultiPolygon)，位置 (Points、MultiPoints) 和距离 (MultiCurves、LegacyArcs)。从 FeatureGeometry 派生和用于地图图元的其它 Geometry 类是 FeatureGeometryCollection 和 LegacyText。（矩形、有圆角的矩形和椭圆对象也从 FeatureGeometry 派生，但主要用于外观显示。）

计算机化地图其中一项主要用途就是获取关于图元的信息。在 MapXtreme 中，可通过多种方法将图元返回到 FeatureCollections，例如，使用构架从空白创建，使用选择集工具或方法选择，或通过搜索这些满足特定标准集合的工具或方法的 Catalog 选择。

可以使用 Load 方法强制 Load。Feature 保存回表之前，对 Feature 所做的更改不反映在底层表（如果有的话）中。使用 Update 方法、UpdateFeature 或 InsertFeature 可以完成该操作。可以在使用 DiscardEdits 方法保存 Feature 对象前删除对其执行的任何编辑。

Feature 具有描述 Feature 属性的构架。Columns 属性描述该构架。

从表检索图元

Table 是 Feature 集合的一种类型。因此，表内的 Features 可以直接枚举。例如：

VB 示例：

```
Dim ftr As Feature
For Each ftr In table
    ...

```

表的默认图元枚举符内部使用具有以下命令的 **MIDataReader**：

```
command.CommandText = "Select MI_Key, * From \" + table.Alias + "\";
```

要检索表中图元的子集，请使用某一种 **Catalog.Search** 方法或 **MICommand.ExecuteFeatureCollection** 方法。

修改表中的图元

要修改表中的图元，请使用以下方法之一。

- **Feature.Update**
- **Table.UpdateFeature**
- **Table.InsertFeature**

Feature 集合

Feature 集合是一组 **Feature** 对象。集合中的所有 **Features** 共享同一 **Schema**（列）。**Feature** 集合包含了某个构架，该构架是所有其成员图元实例的构架。某些 **Feature** 集合拥有它们的 **Features**，而其它 **Feature** 集合维护对 **Features** 的引用。

搜索图元

Pitney Bowes Software 地图绘制应用程序最普通的任务之一就是搜索满足某些标准的图元。有了感兴趣的图元之后，就可以执行更多分析，例如主题绘制。在 **MapXtreme** 中，可通过多种方式搜索图元：使用工具、使用 **Catalog** 搜索方法或使用 SQL 和 **MapInfo ADO.NET Data Provider**。

以下代码示例说明了搜索同一内容的两种方式，该例中的城市为 **New York**。

```
// Using SQL
command.CommandText = "Select Obj From States Where state = 'NY' ;
FeatureGeometry nyGeom = command.ExecuteScalar() as FeatureGeometry;
command.CommandText =
    "SELECT * FROM Cities WHERE Obj within @newyork";
command.Parameters.Add("@newyork", nyGeom);
MIDataReader reader = command.ExecuteReader();
// or... to get a FeatureCollection
IFeatureCollection fc = command.ExecuteFeatureCollection();

// Using Features
Feature fNY = catalog.SearchForFeature("States",
    SearchInfoFactory.SearchWhere("state='NY'"));
SearchInfo si = SearchInfoFactory.SearchWithinFeature(fNY,
```

```
    ContainsFilter.ContainsType.Centroid);
IDynamicFeatureCollection dfc =
    catalog.Search("Cities", si) as IDynamicFeatureCollection;
Console.Out.WriteLine(
    "There are {0} cities whose centroid is within NewYork." -
    dfc.Count);
```

有关 SQL 搜索的完整讨论请参阅 [MapInfo ADO.NET 数据提供方](#)。以下章节着重介绍使用 Catalog 和 SearchInfo 进行搜索。

Catalog 搜索方法

Catalog 具有大量搜索方法成员函数。重载的 Search 方法可用于在一个或多个表上搜索。这些方法包括了不同的参数，使得每个搜索惟一。例如，基本的 Search (Table, SearchInfo) 搜索给定的表并返回 FeatureCollection。Search (ITableEnumerator, SearchInfo) 方法在多个表上搜索并返回 MultiResultSetFeatureCollection。

SearchForFeature 方法从结果返回第一个 Feature。SearchReader 方法与结果一起返回 MIDataReader 光标。

代码示例：SearchForFeature

以下示例说明如何使用 Catalog.SearchForFeature 和 Catalog.SearchWithinGeometry。它查找 uscty_1k 表中位于佛罗里达州的所有城市。假定 "usa" 和 "uscty_1k" 表均打开，此外还有一张地图。

VB 示例：

```
Public Shared Sub MapInfo_Data_SearchInfo(ByVal catalog As Catalog)
    Dim fFlorida As Feature =
        catalog.SearchForFeature("usa", MapInfo.Data._.
        SearchInfoFactory.SearchWhere_("State='FL'"))
    Dim si As SearchInfo =
        MapInfo.Data.SearchInfoFactory.SearchWithinGeometry(fFlorida._.
        Geometry, ContainsType.Centroid)
    Dim fc As IResultSetFeatureCollection =
        MapInfo.Engine.Session.Current.Catalog.Search("uscty_1k", si)

    ' Set the map view to show search results
    MapInfo.Engine.Session.Current.MapFactory(0).SetView(fc.Envelope)
    ' Set the view of the first map.

    ' Add results to selection.
    MapInfo.Engine.Session.Current.Selections.DefaultSelection.Add(fc)
End Sub
```

SearchInfo 和 SearchInfoFactory

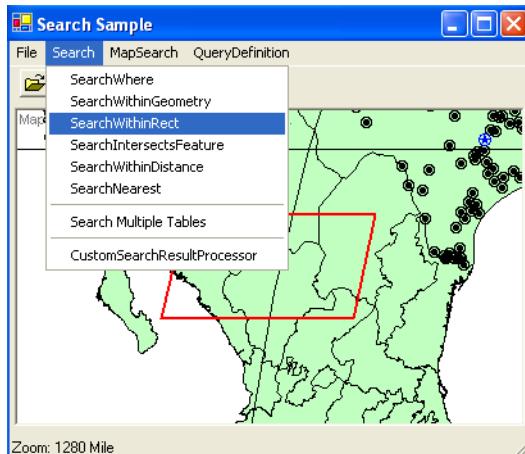
MapInfo.Data.SearchInfo 类定义搜索中所用的查询并处理搜索结果必需的任何后续处理。

SearchInfoFactory 创建 SearchInfo 对象。SearchInfoFactory 包含了大量搜索方法，允许使用对搜索位置的空间引用或使用在屏幕上绘制的几何体进行搜索。

下表说明了 SearchInfoFactory 搜索方法。

SearchInfoFactory 方法	行为
SearchAll	返回所有行。
SearchNearest	返回带有表几何体的行，该表几何体最接近给定的搜索点。
SearchWhere	返回给定的 where 子句指定的行。
SearchWithinDistance	返回表几何体包含在搜索点、矩形或几何体的距离范围内的行。该方法使用 Geometry.Distance 方法确定对象是否在搜索区域内。先前的 SearchWithinDistance 对距离进行缓冲并在缓冲区内搜索，因此结果不够精确。
SearchWithinFeature	返回表几何体包含在搜索图元几何体内的行。
SearchWithinGeometry	返回表几何体包含在搜索几何体内的行。
SearchWithinRect	返回表几何体与给定矩形相交的行。
SearchIntersectsFeature	返回表几何体与搜索图元的几何体相交的行。
SearchIntersectsGeometry	返回表几何体与搜索几何体相交的行。
SearchWithinScreenRadius	在表几何体与屏幕圆圈相交的位置，创建返回行的 SearchInfo。
SearchWithinScreenRect	返回表几何体与给定屏幕矩形相交的行。

MapXtreme 包含了 Search 示例应用程序，可以运行和了解每种搜索类型。下图显示了带有 SearchInfoFactory 方法的 Search 菜单，该方法使用空间引用。Map Search 菜单具有根据绘制的屏幕几何体对象（圆或矩形）进行搜索的方法。Query Definition 菜单突出使用各种作用在 SQL 语句上的筛选器。示例位于 ..\MapInfo\MapXtreme\7.x\x\Samples\Desktop\Features\Search 文件夹中。



代码示例

本节包括了属于 `SearchInfoFactory` 方法的多个代码示例。

SearchNearest

使用 `SearchNearest` 方法后，代码模拟 `Select Point` 工具行为，通过鼠标单击选择最顶层的项目，并将它们添加到选择集。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_SearchInfoFactory(ByVal _
    mapControl1 As MapControl)
    ' Get a point from mouse click. Hard coded value use in sample.
    Dim pt As System.Drawing.Point = New System.Drawing.Point(100, 100)
    ' Assumes there is a MapControl with a map in it.
    Dim map As Map = mapControl1.Map
    Dim session As ISession = MapInfo.Engine.Session.Current
    Dim si As SearchInfo =
        MapInfo.Mapping.SearchInfoFactory.SearchNearest(map, _
            pt, 3) ' 3 pixel tolerance radius
    si.QueryDefinition.Columns = Nothing
    ' fetch all columns instead of just default
    ' Customize to stop at topmost layer where something is found
    CType(si.SearchResultProcessor, ClosestSearchResultProcessor)._
        Options = ClosestSearchOptions.StopAtFirstMatch
    ' Puts results of search directly into default selection
    ' Searches all tables in map in order from top to bottom.
    MapInfo.Engine.Session.Current.Catalog.Search(map.-
        Layers.GetTableEnumerator(), si, _
        session.Selections.DefaultSelection, _
        ResultSetCombineMode.Replace)
End Sub
```

SearchIntersectsFeature

VB 示例：

```
Public Shared Sub _
    MapInfo_Data_SearchInfoFactorySearchIntersectsGeomeTry(ByVal _
        map As Map)
    Dim ti As Table = _
        MapInfo.Engine.Session.Current.Catalog.GetTable("usa")
    Dim lParks As MapInfo.Mapping.FeatureLayer = _
        CType(map.Layers("USA"), MapInfo.Mapping.FeatureLayer)
    Dim g As MapInfo.Geometry.FeatureGeometry = New _
        MapInfo.Geometry.Point(map.GetDisplayCoordSys(), -98, 34)
    Dim si As SearchInfo = _
        MapInfo.Data.SearchInfoFactory.SearchIntersects_
        Geometry(g, MapInfo.Data.IntersectType.Geometry)
    Dim fc As IResultSetFeatureCollection = _
        Session.Current.Catalog.Search("usa", si)
    map.SetView (fc.Envelope)
End Sub
```

SearchWithinScreenRadius

该示例说明了如何使用屏幕绘制的圆来搜索图元。

C# 示例：

```
// find nearest city to center of map
private void menuItemSearchNearest_Click(object sender, System.EventArgs e)
{
    try
    {
        Cursor.Current = Cursors.WaitCursor;
        // to compare to SearchWithinScreenRadius, we are
        // calculating the search distance the same way it does
        System.Drawing.Rectangle rect=mapControl1.Bounds;
        System.Drawing.Point pt = new System.Drawing.Point(rect.Left,
rect.Top);
        pt.X += rect.Width/2;
        pt.Y += rect.Height/2;

        DPoint dpt1 = new DPoint();
        // convert center point to map coords (could use map.Center)
        _map.DisplayTransform.FromDisplay(pt, out dpt1);
        Distance d = MapInfo.Mapping.SearchInfoFactory.ScreenToMapDistance
(_map, 3);

        SearchInfo si =MapInfo.Data.SearchInfoFactory.SearchNearest(dpt1,
_map.GetDisplayCoordSys(), d);
        IResultSetFeatureCollection fc = _catalog.Search("uscty_1k", si);
```

```
MapInfo.Geometry.Point p = new  
MapInfo.Geometry.Point(_map.GetDisplayCoordSys(), dpt1);  
    FeatureGeometry buffer = p.Buffer(d.Value, d.Unit, 20,  
DistanceType.Spherical);  
    ShowSearchGeometry(buffer);  
  
    SelectFeatureCollection(fc);  
}  
finally  
{  
    Cursor.Current = Cursors.Default;  
}  
}
```

代码示例：如何查找包含点的图元

您也可能希望通过相反类型的搜索查找包含点的图元。例如，在运输规划中，可能要了解客户所在区域。

VB 示例：

```
Dim g As MapInfo.Geometry.FeatureGeometry = New _  
    MapInfo.Data.SearchInfoFactory.SearchIntersectsGeometry(g, _  
    MapInfo.Data.IntersectType.Geometry)  
Dim irfc As MapInfo.Data.IResultSetFeatureCollection = _  
    MapInfo.Engine.Session.Current.Catalog.Search("british_ _  
    columbia", si)  
Me.MapControl1.Map.SetView(irfc.Envelope)
```

分析数据

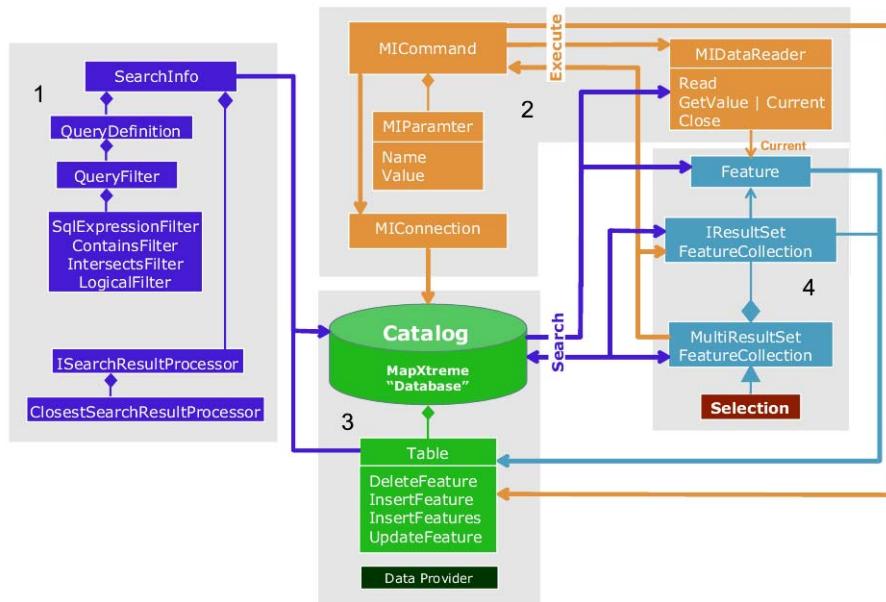
一旦数据可用于 Catalog，您就会根据自己的业务目标对其进行分析。Catalog 的 SQL 处理器可用于解析和合计数据。有两种选择：

- 基于 OGC 对象的查询接口
- 基于 ADO.NET SQL 的接口

下图显示了两者之间的关系。

第 1 组显示 OGC 查询接口。使用这些对象构造查询。该接口可以用于创建查询以过滤列和行，并添加空间和非空间条件。查询通过 Search 方法进行交互（而不是查询对象），然后返回数据读取器和结果集。如果更愿意使用面向对象编程技术而用 SQL 语法较少，则使用这些对象。请参阅 [SearchInfo](#) 和 [SearchInfoFactory](#)。

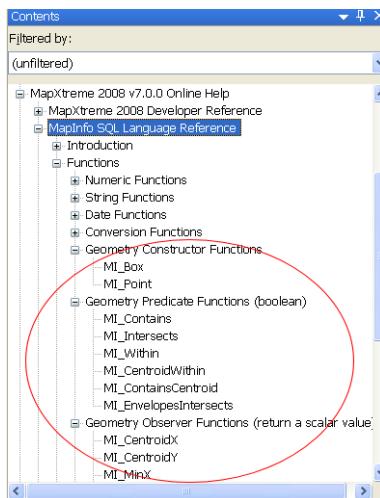
ADO.NET 接口（如第 2 组中显示）使用定义的 ADO.NET 模型，允许通过 MapInfo SQL 语言访问。ADO.NET 接口使用 SQL 语法与 Catalog 交互。在此实例中，需要生成 SQL 状态，并将其分配到 MySqlCommand 对象。这些对象使用 Execute 命令返回数据读取器或结果集。请参阅 [MapInfo ADO.NET 数据提供方](#)。



MapXtreme 数据模型

基于查询的 OGC 和基于命令的 ADO.NET 方法使用 Catalog（第 3 组）来组织数据源作为对象或 SQL 查询的响应。基于对象的查询 API 将生成 SQL，并将此传递给 Catalog 进行处理。在一些实例中，您可以手动生成更有效的 SQL，但是对象可以很好地进行定义并且接口限制交互的方式，因此最好使用 SQL。如果您愿意使用 SQL 语言，那么使用 ADO.NET 方法会更习惯。但是如果您的经验没有使用 SQL 的经验，那么基于查询的 OGC 对象也可以很好地工作。

MapInfo SQL 语法在 MapXtreme 附带的 SQL 参考中定义，安装 MapXtreme 后集成到 Visual Studio 中。该语言基于 SQL3，具有为空间分析而定义的特殊 MapInfo 运算符。这些运算符以 MI_ prefix 开头。



MapInfo SQL 语言参考的目录显示在 *Visual Studio* 中

Data Readers、MemTables 和 Result Sets

访问数据返回数据读取器或结果集的方法。数据读取器允许以顺序方式访问，不存储数据的副本。除非在缓存数据源的情况下，否则均从数据源检索数据。结果集是键的集合。这些键可以用于访问回原始表，并不创建该数据的副本。

MemTable 也可以用于将数据从各种源中存储到表中。此种类型的表以内存数组与临时磁盘存储方式存储数据。添加数据时，**MemTable** 制作一个数据的副本，且没有键或指针指向原始表。这些对于地图和容器的临时图层来讲是非常有用的，可以返回进程值如地理编码或路由结果。**MemTable** 访问和地图渲染性能与本地表相同。

当需要访问定义的一组行和需要从源获取数据时，结果集会是一个非常有用的工具。如果源数据在会话过程中可以更改，如果数据源支持同步访问，则此方法可用于查看结果。因为 **MemTable** 是数据的副本，它们是数据行的静态集，将不反应用于原始数据源的更改。

提升数据访问性能

性能始终是任何应用程序访问数据的一个重要方面。请考虑应用程序的设计和开发规划中的以下列表。

- 只请求需要的数据（尤其是从 **RDBMS**）。这就限制了通过连接发送的数据数量。
- 对于 **web** 应用程序，请将属性列信息纳入工作空间文件中，以便在 **MapInfo Session** 创建时间能预载所有必需的数据。有关示例请查看[主题图示例概述](#)。
- 仅当需要井然有序的列表时，才对表进行排序。此过程需要花费时间来读取整个表以建立顺序。如果列没有索引，排序还会更慢。
- 仅当需要随机访问表，才将其滚动。此外还生成索引来加速访问和记住顺序。数据读取器直接访问数据，而不需要读取附加数据。
- 使用一致坐标系进行 **Join** 和 **Search** 操作。从而不需要每一次访问时都转换几何体。
- 使用带索引的列来进行 **Join / Filter / Sort / Aggregate** 等操作。
- 实际检查几何体相交之前，使用 **CentroidWithin**、**ContainCentroid** 和 **EnvelopesIntersect**。这些测试非常快速，大多数情况下可以从列表中不费力地删除大量几何体。
- 当执行多查询和/或编辑时，使用 **BeginAccess/EndAccess**（特别是对基于文件的表）。
- 尝试避免在 **where** 子句中调用 **Area** 和 **Buffer**，因为每次创建一个新光标时必须执行此操作。
- 定义视图或结果集时，尝试避免在 **Select** 列表中调用 **Area** 和 **Buffer**，原因同上。
- 对于管理键的中间结果或操作使用结果集。这些是很轻量的操作，可以提供快速直接访问原始数据。

运用核心 MapXtreme 类

`MapInfo.Engine` 命名空间包含了所有直接与核心功能有关的接口和类，该核心功能驱动基于 `MapXtreme` 的所有应用程序。其中包括核心 `ISession` 接口，该类是所有 `MapXtreme` 应用程序的起点。该命名空间中的类包括 `Session`、`Selections` 和 `SearchPath`。命名空间中的其他类型支持 `Collections`、`Resources` 和 `CustomProperties` 的类、委托、结构体和枚举。

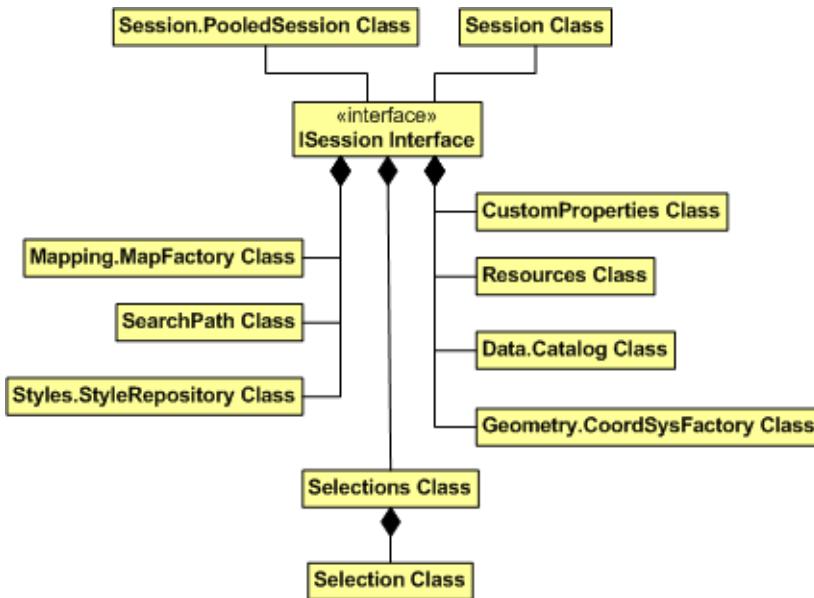
在本章中：

◆ Session 接口	189
◆ 串行化和永久化	191
◆ 打开和保存包含命名资源的工作空间	193
◆ Selection 类	194
◆ Selection 代码示例	196
◆ 事件参数	198
◆ 异常	198

Session 接口

ISession 接口是所有基于 MapXtreme 的应用程序的起点。它管理 MapXtreme 应用程序所需资源的初始化。

ISession 的实例保存 MapXtreme 对象模型的组件，如 DataAccess 引擎、MapFactory、CoordSysFactory，这样桌面或 web 应用程序才能够运行。下图对实现 ISession 接口的类进行了说明。



在 ASP.NET 应用程序中，每个客户端请求都具有其自身的 ISession 实例。该实例驻留在调用上下文中，在客户端请求的整个生命周期中都可用。

在单一线程的桌面应用程序上，只有一个实例。在多线程桌面应用程序中，每个线程都有一个实例。

`MapInfo.Engine.Session` 类提供了对 ISession 对象的访问。要获得当前的 ISession 实例，请访问 `MapInfo.Engine.Session.Current` 属性。

会话管理

会话管理是在设计应用程序时需要了解的关键主题。桌面应用程序的会话管理简单易懂（每个用户都具有其自身的 ISession 实例）时，而 Web 应用程序则需要考虑或许会有未知数目的用户使用您的应用程序。必须了解如何处理每个用户的状态，以便将正确的信息和可视化显示返回给正确的用户。

MapXtreme 提供了用于构建 web 应用程序的模板，帮助您正确管理状态。有关该主题的讨论请参阅 [第 6 章：了解状态管理](#)。显示有关状态管理、对象池、性能和数据访问的关键决定，有助于在开始编码前的项目设计阶段做出周密的决定。

使用 **Session.Dispose** 方法

`MapInfo.Engine.Session` 类包含两个预装的 `Dispose` 方法。您的选择将取决于要构建的应用程序类型。

Session.Dispose()

`Session.Dispose()` 会处置 `ISession` 实例，该实例可通过 `Session.Current` 属性访问。该方法仅用于多线程桌面应用程序。

不要将该方法用于 `web` 应用程序或单线程桌面应用程序。对于 `web` 应用程序，`ISession` 由 `WebSessionActivator` 管理。

对于单线程桌面应用程序，会在应用程序关闭或使用 `MapXtreme` 的 `AppDomain` 被卸载时自动调用 `Dispose`。

Session.Dispose(HttpContextState)

在使用默认会话状态设置的 `web` 应用程序中使用 `Session.Dispose(HttpContextState)`，因为此时 `ISession` 存储在内存中。不要在任何其它配置中调用该方法，因为在任何其它配置中 `ISession` 实例都不存储在内存中。

状态设置通过下列键表示在应用程序项目的 `web.config` 文件中：

```
<add key="MapInfo.Engine.Session.State" value="HttpSessionState" />
<sessionState mode="InProc" />
```

第一个设置是特定应用程序的设置，它控制用于保存和恢复 `MapInfo.Engine.ISession` 实例状态的机制。该实例可通过 `MapInfo.Engine.Session.Current` 属性访问。`HttpSessionState` 设置表示该会话通过 `ASP.NET` 会话状态进行保存和恢复。该状态通过当前的 `HttpContext` 公开，属于 `HttpSessionState` 类型。

第二个设置是 `ASP.NET` 设置，它控制 `HttpSessionState` 的保存和恢复方式。`InProc` 表示 `ASP.NET` 会话的状态被放置在内存中并且对于每个 `ASP.NET ISession` 实例都是唯一的。这是默认设置。

使用这些设置时，每个 `ASP.NET` 会话都有一个 `ISession` 实例，该实例在 `ASP.NET` 会话的整个生命周期都存储在 `HttpSessionState`。要在会话暂停或结束时正确处置 `ISession` 实例，则必须将下列语句添加到 `Global.asax` 源码文件下的 `Session_End` 方法中。

VB 示例：

```
MapInfo.Engine.Session.Dispose(Me.Session)

Protected Sub Session_End(ByVal sender As Object, ByVal e As EventArgs)
    MapInfo.Engine.Session.Dispose(Me.Session)
End Sub
```

执行该调用可以确保正确处置 `ISession` 实例且回收占用的内存。

I SessionEventHandlers

MapXtreme 提供 `MapInfo.Engine.ISessionEventHandler` 接口，用于加载扩展 MapXtreme 功能的自定义 DLL。使用该接口自动加载任意所需的应用程序扩展 DLL，比如：默认工作空间可使用的可扩展数据提供器、永久性提供器和样式。MapXtreme 启动时，将初始化这些程序集，然后执行需要的指令。

例如，在编译时，MapXtreme 提供的 `Spatialite` 样本实现会产生包含支持 `Spatialite` 数据提供器的程序集，以及会在 MapXtreme 会话初始化时加载数据提供器。

为了在启动时初始化 `SessionEventHandler` 程序集，这些程序集必须位于 `\Common Files\MapInfo\MapXtreme\7.x.x\SessionEventHandlers` 文件夹中。实际上，将加载该文件夹中任意带有 `.SessionEventHandler.DLL` 文件后缀的程序集。

`ISessionEventHandler` 接口提供的两种方法可调用加载任意所需的应用程序扩展代码，并且提供了附加的初始化功能。

```
void BeforeWorkspaceLoad(ISession session)
```

在加载任意默认工作空间前调用，可用于设置由默认工作空间使用的任意可扩展的数据提供器、`edp` 回调、`edp` 持久性提供器、命名连接、加载或创建样式、打开表、数据库连接等等。

```
void AfterWorkspaceLoad(ISession session)
```

在加载任意默认工作空间（以 `app.config` 或 `web.config` 为单位指定）后调用。此时可以操作应用程序需要的任意最终会话初始化。

当 MapXtreme 发现实现 `ISessionEventHandler` 接口的程序集中的类时，它会构造该类的实例并将其添加到内部集合。无论何时创建新的 MapXtreme 会话实例（可以是桌面应用程序的一次会话，也可以是 Web 应用程序的多个会话），可用已初始化的会话实例调用会话处理程序。

由于可以有多个已加载的 `SessionEvent` 处理程序且调用的顺序是未确定的，若正在写入处理程序，请不要对“什么已出现在会话中”做任何假设（比如：地图、表等等）。

使用该功能时要小心！在计算机上创建的任意 MapXtreme 会话将加载这些会话事件的处理器。当这些对话显示在服务器上时，添加显示对话窗口或其他用户界面组件的处理器可能导致 Web 应用程序挂起。

串行化和永久化

Map 绑定到 `ISession` 对象。无法从一个 `ISession` 获取 `Map` 对象，而在另一个 `ISession` 对象中使用 `Map` 对象。如要克隆整个 `ISession` 对象，可使用两个选项 — 通过使用 `Serialization` 或通过使用 `Persistence`。

序列化 (Serialization)

序列化是将对象转换为数据流的过程，从而将对象保留在服务器上。该过程是在 **MapXtreme web** 应用程序中维护对象的基本部分。如果对象没有维护，则服务器需要为每个 web 请求重建对象（例如地图）。对象被请求时，首先反序列化对象（或从数据流重建），然后修改对象。因为该算法不生成对象的副本（与其它串行化算法相同），所以反串行化的对象仅创建一次。

串行化通过嵌入 Microsoft.NET Framework 中的格式化程序来执行。Framework 中包括两个不同的格式化程序，一个是用于二进制对象的 **BinaryFormatter**，另一个是用于 SOAP 对象的 **SOAPFormatter**（SOAP 是轻型协议，用于在诸如 web 的分布式环境中交换结构化信息）。相对来说，**SOAPFormatter** 比 **BinaryFormatter** 要快。**SOAPFormatter** 用于某些基本的数据类型（Int、Byte、Decimal 等），而 **BinaryFormatter** 用于复杂的对象。有关 **SOAPFormatter** 和 **BinaryFormatter** 的详细信息请参阅 Microsoft MSDN 文档。

要将对象传递到其中一个格式化程序，请使用 **GetObjectData()** 方法。要反串行化对象（从数据流恢复对象），请使用 **SetObjectData()** 方法。

任一支持 **ISerializable** 接口的对象都将被自动恢复或反序列化。ASP.NET 框架在 **HttpApplication.BeginRequest** 后自动反序列化 **context.Session[]** 数组。**MapInfo.Engine.Session** 在 **HttpApplication.BeginRequest** 句柄中设置，以便将对象反序列化到 **MapInfo.Engine.Session** 中。

序列化 / 反序列化 Session 对象

以下实例说明了如何将 **Session** 对象序列化/反序列化。

```
// Create a MemoryStream to serialize into
MemoryStream stream = new MemoryStream();
// Serialize the MapXtreme Session object
BinaryFormatter formatter = new BinaryFormatter();
formatter.Serialize(stream, Session.Current);
stream.Position = 0;
// Make changes to the Session object to make sure the
// deserialization works correctly
...
// Recreate the MapXtreme Session object from the stream
// Note: this will replace the current MapXtreme Session object with the
// contents of the stream
formatter = new BinaryFormatter();
formatter.Deserialize(stream);
```

传递到 **formatter.Serialize** 方法的数据流参数可以是从 **System.IO.Stream** 派生的任何内容。

该序列化功能设计用于 **MapXtreme** 的状态管理图元。如将该信息保存到磁盘，则以后尝试重载该信息时（用本产品的其它版本）不保证能正常工作。

有关序列化和状态管理的详细信息请参阅实现 **StateManager**。

永久性

MapXtreme 中的永久化是将应用程序对象存储为 **MapInfo Workspace (.MWS)** 文件（**XML** 格式）的过程。永久化和反永久化 **ISession** 对象可通过使用 **WorkSpacePersistence** 和 **WorkSpaceLoader** 类来完成。这两个类将写入和读出 **.MWS** 文件。应用程序可写出工作空间（**Session** 的副本），并将工作空间文件的内容应用到新的 **Session** 对象，从而创建 **Session** 的克隆。

VB 示例：

```
Public Shared Sub MapInfo_Persistence_WorkSpacePersistenceSave()
    ' Create a named connection point to "D:\data\version2"
    Dim info As NamedConnectionInfo = New NamedConnectionInfo("file", _
        ConnectionMethod.FilePath, "D:\data\version2")
    Session.Current.Catalog.NamedConnections.Add("MyDataFolder", _
        info)

    ' Create a map
    Dim map As Map =
        Session.Current.MapFactory.CreateEmptyMap("MyMap", _
            "MyMapAlias", New Size(400, 400))
    Dim table As Table =
        Session.Current.Catalog.OpenTable("MyDataFolder", _
            "myTableAlias", "Seamless\Lines - NYALBA\SeamCapDist.TAB")
    map.Layers.Add(New FeatureLayer(table))

    ' Save the Session to a workspace file
    Dim w As WorkSpacePersistence = New WorkSpacePersistence
    w.Save("c:\workspace\mySeamless.mws")
End Sub
```

注意，添加工作空间（使用 **WorkSpaceLoader** 类）是个累积的过程。要确保 **Mapxtreme Session** 仅包含新工作空间文件的内容，请首先在 **ISession** 对象上调用 **Reload()** 方法。该方法清除 **Session** 状态并在应用程序 **Web.config** 文件中重新加载预定义的工作空间。

有关永久性和 **XML** 唯一的详细信息，请参阅[附录 C：理解 **MapInfo** 工作空间](#)。

打开和保存包含命名资源的工作空间

MapXtreme 支持命名资源的地图定义、地图图层、数据源定义和样式。之前 **MapXtreme** 支持的命名连接。命名资源是对另一位置中资源的引用，通过名称加以标识。

可通过 **MapInfo.Engine** 命名空间中的 **INamedResourceResolver** 接口以编程方式打开和保存包含命名资源的 **MapInfo Workspace** 格式文件 (**.MWS**)。此接口包含用于在分析 **MWS** 文件时解析命名资源的函数，还包含可在保存 **MWS** 文件时生成命名资源 **xml** 节点的函数。

打开 MWS: **ResolveResource()**

打开和分析包含命名资源的 MWS 时，**ResolveResource()** 函数将返回与命名资源的名称和类型对应的 **XmlNode**。

例如，当分析 WMS 的命名样式时，会以字符串的形式传递 **NamedResourceType.Style** 和 **XmlNode**。**ResourceResolver** 分析参数，并以 **XmlNode** 的形式返回命名资源节点。

保存 MWS: **GetResourceName()**

当保存 MWS 文件时，在内存中分析的文件将包含所有展开的节点。要将命名资源写回至 MWS 文件中，您需要实现 **GetResourceName** 的所有风格，**GetResourceName** 会获取 **Table**、**Map**、**Style** 和 **IMapLayer** 的所有对象。这些函数将返回相对应对象的 **NamedResource** 节点。保存的 MWS 文件中包含命名资源的条目。

使用 **MapXtreme** 注册您的实现。

要使用 **MapXtreme** 注册 **INamedResourceResolver** 的实现，您必须设置 **Session.Current.NamedResourceResolver** 属性。现在，**MapXtreme** 会在打开和保存 MWS 文件时引用您的实现对象。

设置参数选择

如果您仅想保存特定类型的命名资源而使其它命名资源保持不变，您必须使用 **WorkspacePersistence** 对象中的 **PreferenceForNamedResource** 属性设置参数选择。默认情况下，其被设置为 **NameResourcePreference.All**。您也可以设置其它值，例如，设置为 **NameResourcePreference.PreferNamedTables** 以便仅将表格保存为命名资源。要将多个资源保存为命名资源，请使用 OR 运算符。例如：

```
NameResourcePreference.PreferNamedTables |  
NameResourcePreference.PreferNamedLayers.
```

有关详细信息，请参阅《开发人员参考》中的 **INamedResourceResolver**。

Selection 类

Selection 是保存图元列表的 **IResultSetFeatureCollection** 对象的集合。这些图元是表中行的子集。这些图元可以是属性边界、街道网络、手机信号发射塔位置或自然图元（例如河流）。显示在 **Map** 上时这些图元通常用特殊的突出显示绘制。对于 **Selection** 中的给定表，只可以有一个 **IResultSetFeatureCollection**。

ISession 中可以有多个 **Selection**。**Selections** 集合包含了应用程序中所有的选择集。始终至少有一个选择集，名为 **DefaultSelection**。

每个 **Selection** 必须具有一个名称和惟一的别名。默认情况下，地图选择集工具修改 **Selection**（如果使用）。可以分别设置每个工具来使用任何特定的 **Selection**。

MapXtreme 中的选择集不是副本；对于 **Selection** 中的给定表它是对 **IResultSetFeatureCollection** 的引用。如果关闭正处理的表后尝试修改 **Selection**，则对 **IResultSetFeatureCollection** 的引用将无效，会导致异常。

图元可以通过使用工具或搜索方法来选择。有关构建 **Windows** 窗体应用程序时可以使用的不同选择工具的讨论，请参阅[第 7 章：桌面应用程序、控件、对话框和工具](#)。有关选择集 **Web** 控件和工具的信息，请参阅[第 5 章：Web 应用程序、控件和工具](#)。

图元也可以通过源于 **MapInfo.Data.Catalog** 类的搜索方法来选择，其中 **MapInfo.Data.Catalog** 类返回 **IResultSetFeatureCollection** 集合。**Selection** 对象可传入用于填充或更改 **Selection** 的搜索中。

图元也可以通过源于 **Data.MICommand** 类的 **ExecuteFeatureCollection** 方法来选择。这种情况下，会对 **MapInfo** 数据提供方执行 SQL 命令。

有关图元、表、**Catalog** 和 **MICommand** 的详细信息，请参阅[第 8 章：使用数据](#)。

使用选择集的属性

Selection 类的属性用于设置（必需的）名称和别名、将选择集设置为 **Visible** 或 **Editable**，或者用于获得选择集的 **Style**。要确定 **Selection** 是否显示其突出显示以指示其是否被选中或可用于编辑，请分别使用 **Visible** 或 **Editable** 属性。

如果 **Editable** 属性设置为 **true**，则正处理的表也必须是可编辑的。

Style 属性指示 **Selection** 的样式并返回 **Selection** 组合样式的引用。更改 **Style** 属性时，会通知 **Selection** 对象相应更改。下次绘制 **Selection** 对象时，更改生效。有关 **Style** 的讨论请参阅[第 15 章：为地图设置样式](#)。

突出显示和导出选择集

选择集通常用特殊的突出显示在地图上绘制，以此将这些图元与周围未选中的图元区分。突出显示的部分由 **Mapping.FeatureViewer.DrawSelections** 属性控制。该属性设置为 **true** 时，如果这些图元绘制在可见的图层上，则这些图元用突出显示的选择集绘制。

同样，**MapExport.ExportSelection** 属性可用于控制选择集是否绘制到导出图像中。

SelectionChangedEvent

委派方法被附加到 **SelectionChangedEvent**，以便接收该选择集已更改的通知。例如，如果添加记录则激发 **SelectionChangedEvent**。

Selection 和 Selections 类上的 ISerializable 接口

ISerializable 接口在 Selection 和 Selections 类上实现。以下代码示例说明了如何将 Selections 对象串行化或反串行化：

```
// Create a MemoryStream to serialize into
MemoryStream stream = new MemoryStream();

// Serialize the Selections object
BinaryFormatter formatter = new BinaryFormatter();
formatter.Serialize(stream, Session.Current.Selections);
stream.Position = 0;

// Make changes to the Session's Selections object to make sure the
// deserialization works correctly.
...

// Recreate the Selections object from the stream.
// Note: this replaces the current Session's Selections object with
// the contents of the stream
formatter = new BinaryFormatter();
formatter.Deserialize(stream);
```

Selection 代码示例

以下是常用选择集操作的代码示例。其它代码示例包括在 MapXtreme 开发人员参考的多个主题中。

选择另一个图元中的图元

使用 MapXtreme 的常用搜索技术是在另一个图元中查找图元。通过该操作可以查找邮政编码边界内的所有客户或某个地区内正在构建的所有高速公路。请参照以下示例。参数 f 代表 MapInfo.Data.Feature。

VB 示例：

```
Dim si As MapInfo.Data.SearchInfo =
    MapInfo.Data.SearchInfoFactory.SearchWithinFeature(f, _
        MapInfo.Data.ContainsType.Centroid)
Dim irfc As MapInfo.Data.IResultSetFeatureCollection =
    MapInfo.Engine.Session.Current.Catalog.Search("USCty_8k", si)

MapInfo.Engine.Session.Current.Selections.DefaultSelection.Clear()
MapInfo.Engine.Session.Current.Selections.DefaultSelection.Add(irfc)
irfc.Close()
```

检查选择集的表

请参照以下代码示例学习如何获得表中的选择集计数。

VB 示例：

```
Public Shared Sub MapInfo_Engine_Selection2()
    Dim session As ISession = MapInfo.Engine.Session.Current
    Dim tableUsa As Table = session.Catalog("usa")

    ' Get fc for selection on usa.
    Dim fc As IResultSetFeatureCollection =
        session.Selections.DefaultSelection(tableUsa)
    Dim nCount As Integer = 0
    If Not fc Is Nothing Then
        nCount = fc.Count
    End If
End Sub
```

也可以使用 **MapInfo SQL** 查询和 **ADO.NET** 数据提供方执行选择集操作。请参阅第 8 章：使用数据。

从表中返回所有列

以下示例说明了如何从选择集中返回所有列：

VB 示例：

```
Dim connection As MIConnection = New MIConnection
    Connection.Open()
    Dim lyr As FeatureLayer = MapControl1.Map.Layers("usa")
    Dim ti As MapInfo.Data.Table =
        MapInfo.Engine.Session.Current.Catalog.GetTable("usa")
    Dim si As MapInfo.Data.SearchInfo =
        MapInfo.Data.SearchInfoFactory.SearchAll()
    si.QueryDefinition.SetColumns("*")
    Dim irfc As MapInfo.Data.IResultSetFeatureCollection =
        MapInfo.Engine.Session.Current.Catalog.Search(ti.Alias, si)
    Dim l As MapInfo.Data.Feature
    For Each l In irfc
        Dim column As MapInfo.Data.Column
        For Each column In l.Columns
            MessageBox.Show(column.ToString())
        Next
    Next
```

更改选择集后的地图视图

以下示例说明了如何更改缩放以显示选择集中的所有图元。

VB 示例：

```
Me.MapControl1.Map.Bounds =
    MapInfo.Engine.Session.Current.Selections.DefaultSelection.Envelope.Bounds
```

事件参数

MapInfo.Engine 命名空间包含了各种为事件提供数据的事件参数类。有关详细信息请参阅联机帮助。某些事件参数类包括：

- **CollectionCancelableEventArgs** - 为可以被取消的集合事件提供数据。
- **CollectionEventArgs** - 提供集合事件的数据。
- **NodeSelectionChangedEventArgs** - 节点选择集更改时激发这些事件参数。
- **SelectionChangedEventArgs** - 其它对象可将委派附加到该事件以在选择集更改时获得通知。

异常

Engine 命名空间包含了各种异常类。有关详细信息请参阅联机帮助。某些异常类包括：

- **ResourceNotFoundException** - 在 **Resource** 表中未找到请求的对象时抛出此类异常。
- **ResourceTypeMismatchException** - 当从 **Resource** 中读取的对象不是预期类型时抛出异常。
- **TimeoutException** - 等待池中的 **ISession** 成为可用时，如果 **Current** 超时则抛出异常。

创建表达式

在整个 **MapXtreme** 中，表达式用于确切描述需要在地图绘制应用程序中显示和分析的所有信息。本章介绍了如何为各种各样的产品创建表达式，包括数据访问、创建主题、标注地图等。

在本章中：

- ◆ 表达式概述 200
- ◆ 创建表达式 200
- ◆ **Where** 子句 - 布尔表达式 201
- ◆ 表达式中的函数 201
- ◆ 表达式示例 202

表达式概述

表达式是用于描述和格式化数据的语句。例如，会有“*a median income of more than \$50,000*”或“*female percent of population*”这样的英文表达式。

表达式使用列名、常数（即特定数据值）和作用在列和常数上的函数和运算符形成。运算符和函数以 MapInfo SQL 语言定义，这是为 MapXtreme 和其它 MapInfo .NET 支持的产品开发的功能。有关详细信息，请通过 Visual Studio 中的“帮助查看器”参阅《MapInfo SQL 参考》。

使用表达式可以生成大部分数据。通过使用表达式，您可以：

- 仅显示感兴趣的列和行。
- 根据现有列的内容通过计算新值派生新的列。
- 合计数据来处理分类汇总而不是整个表。
- 将两个或多个表的数据组合到一个结果表中。

要用的数据集包括的对象和信息要比项目所需的对象和信息多。很多情况下，运用完整数据产品的子集更为容易。例如，如果要通过人口普查记录跟踪某个郡的犯罪统计，则不必处理整个州的人口普查记录。可以使用表达式仅提取该郡的人口普查记录。

在整个 MapXtreme 中，表达式用于以下情况：

- SQL 语句 (`select, insert, update, delete, group by, order by`)
- 将表达式作为参数使用的 SQL 函数（例如 `MI_Area()` 中的几何体参数是返回几何体对象的表达式。）
- 添加列 (`MapInfo.Data.Table.AddColumn` 根据表达式创建临时列。)
- 图元搜索 (`SearchInfo` 和 `SearchInfoFactory`)
- 主题 (`FeatureStyleModifier`)
- 标注 (`LabelModifier`)
- InfoTips (`FeatureLayer`、`MapTools`)
- 表达式对话框

创建表达式

最简单的表达式由常数组成，例如“2”（数字示例）或“Friday”（文本示例）。

其它简单表达式由列名称组成，例如：

POP_2000
STATE

在 `Select` 语句中请求特定的多个列时，例如这些列一起称为表达式列表。

```
Select colA, colB, colC from Table1, Table2  
Select colA/2, ColB/ColC from Table1
```

也可以编写对数据执行数学运算的表达式。

例如，**RENT + UTILITIES** 是将两列相加在一起的表达式。该表达式可以用在 SQL 语句中查找每月总费用少于 \$800 的公寓。

Where 子句 - 布尔表达式

布尔表达式是搜索条件，生成 **True** 或 **False**。例如，表达式

`2 < 5`

是布尔表达式，因为结果是 **True**。

所有包含关系运算符（例如小于号 `<`）的表达式是布尔表达式。运算符 **AND**、**OR** 和 **NOT** 是布尔运算符。布尔表达式也称为比较表达式、条件表达式和关系表达式。

```
POP_2000 > 500000
POP_2000 <= POP_1990
PROVINCE <> 'Ontario'
County = 'Columbia' AND VALUE >= 250000
```

MapInfo SQL 中支持的运算符的定义可在 **MapInfo SQL** 参考中找到，通过 **Visual Studio** 集成的帮助（在“动态帮助”目录窗格中查找“**MapInfo SQL** 参考”）可联机查看 **MapInfo SQL** 参考。

布尔表达式用于 SQL 语句的“**where** 子句”中。**where** 子句是控制返回的行（生成 **True** 的行）的表达式。

例如，该语句中的布尔表达式后跟 **WHERE**。只有落在 **France** 边界内位于 **Europe** 表中的对象才返回 **True**。

```
"SELECT * FROM Europe WHERE MI_Geometry within @France";
```

表达式中的函数

MapXtreme 中的函数用于创建更加复杂的表达式来检索满足特定标准的数据。例如，**MapInfo SQL** 支持处理字符串、日期/时间和数字的大量常用数据库函数。

MapInfo SQL 中最强大的函数是那些利用绘制地图数据的空间性质的函数。这些地理函数用于创建新的几何体，度量面积和长度，返回空间信息，验证多个几何体之间的空间关系等。支持的函数在 **MapInfo SQL** 参考中定义。

查看边界表的面积（例如学校地区）就是在表达式中使用函数的一个示例。使用函数 **MI_Area()** 返回表中每个记录的面积。

其它在表达式中使用函数的示例可以在以下的“表达式示例”一节中找到。

DateTIme 和 Time 表达式

当将 `DateTime` 和 `Time` 表达式与 `MapXtreme` 搭配使用时，请注意以下内容：

- 如果在表达式中单独使用 `DateTime` 列或 `Time` 列，将使用当前的区域设置格式化这些列。
- 如果 `DateTime` 或 `Time` 列位于表达式中，则其字符串值为“`TimeToNumber` 或 `DateToNumber + 空格 + TimetoNumber`”。
- 不支持对 `Time` 或 `DateTime` 使用运算符 `math`。您可以为 `Date` 加上数字，但不能为 `Time` 或 `DateTime` 加上数字。

表达式示例

以下章节着重说明了在 `MapXtreme` 中使用表达式的各种情况。

SQL 语句示例

该示例将从位于 `Germany` 且人口超过 1 百万的 `Eurcity_1K` 表选择所有记录。

```
Select * from Eurcity_1K WHERE (MI_Geometry MI_Within @Germany) AND  
Tot_Pop > 1000000
```

以下示例根据表中的 `Time` 和 `Date` 列进行选择。本示例还将从 "CrimeActivity" 表中选择深夜 12:00:00 到凌晨 06:00:00 之间发生的所有犯罪记录。

```
SELECT * FROM CrimeActivity WHERE CrimeTime BETWEEN '12:00:00 AM' AND  
'6:00:00 AM'
```

此处 `CrimeTime` 是存储罪犯作案时间的 `Time` 列。

本例将从 "Employee" 表中选择在 1970 年 12 月 31 日之前出生的员工姓名。

```
SELECT Names FROM Employee WHERE BirthDay < '12/31/1970 12:00:00 AM'
```

此处 `BirthDay` 是存储员工生日的 `Date` 列。

MapInfo SQL 函数示例

以下表达式使用 `MapInfo SQL` 函数查找缓冲区内的图元。

```
Obj CentroidWithin MI_Buffer(Obj, 5, 'km', 'Spherical', 24)
```

该表达式使用 `MapInfo SQL` 特殊关键字，为名为 ‘`Obj`’ 的地理对象所保留。该关键字说明了诸如坐标系和边界的对象的几何体。该关键字与以前版本的 `MapX` 和 `MapInfo Professional` 兼容。且等价于列名称 `MI_Geometry`。

注意，`km` 和 `Spherical` 包括在单引号内。在 `MapInfo SQL` 中，字符串文字必须包括在单引号内，而标识符（例如列名称、表名称、别名等）应该包括在双引号中。只有分析的逻辑不能正确分析标识符时，标识符才需要加引号。这样就会包括标识符，这些标识符的名称或其它特殊字符中包含空格。

要查找位于缓冲区外的图元，可以使用以下表达式：

```
NOT Obj.CentroidWithin MI_Buffer(Obj, 5, 'km', 'Spherical', 24)
```

增加列示例

使用 `AddColumns` 方法将临时列添加到表时，提供的列包含了定义如何计算列值的表达式。如果多个源记录要匹配到添加列的表中的单一记录，则表达式可以包含聚合函数。

以下示例使用表示人口密度的表达式 "Pop_1990 / MI_Area(Obj, 'sq mi', 'Spherical')"。这些表达式前是它们新的列名称。`PopDensity1990` 和 `PopDensity2000`。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableAddColumns(ByVal miTable As Table)
    Dim NewCols As Columns = New Columns
    NewCols.Add(New Column("PopDensity1990", "Pop_1990 / MI_Area(Obj, 'sq mi', 'Spherical')"))
    NewCols.Add(New Column("PopDensity2000", "Pop_2000 / MI_Area(Obj, 'sq mi', 'Spherical')"))
    miTable.AddColumn(NewCols)
End Sub
```

有关增加列的详细信息，请参阅[将表达式列增加到表](#)。

图元搜索示例

以下示例使用布尔表达式 `SearchWhere("State='FL")`，该表达式执行时将为包含 FL 的每行返回值 1。

VB 示例：

```
Public Shared Sub MapInfo_Data_SearchInfo(ByVal catalog As Catalog)
    Dim fFlorida As Feature = catalog.SearchForFeature("usa",
        MapInfo.Data.SearchInfoFactory.SearchWhere("State='FL'"))
    Dim si As SearchInfo =
        MapInfo.Data.SearchInfoFactory.SearchWithinGeometry(fFlorida.Geometry,
            ContainsType.Centroid)
    Dim fc As IResultSetFeatureCollection =
        MapInfo.Engine.Session.Current.Catalog.Search("uscty_1k", si)

    ' Set the map view to show search results
    MapInfo.Engine.Session.Current.MapFactory(0).SetView(fc.Envelope)
    ' Set the view of the first map.

    ' Add results to selection.
    MapInfo.Engine.Session.Current.Selections.DefaultSelection.Add(fc)
End Sub
```

有关 `Feature` 类和搜索方法的详细信息，请参阅 [Features](#) 和 [Feature 集合](#)。

主题图表达式示例

饼图主题是根据主题表达式的数字值绘制饼图的对象主题。主题的表达式由三列组成：
"Pop_Native"、"Pop_Asian"、"Pop_Other"。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Thematics_PieTheme(ByVal _  
    map As Map)  
    ' Load a map based on one table  
    map.Load(New MapTableLoader("world.tab"))  
    Dim lyr As FeatureLayer = CType(map.Layers("world"), FeatureLayer)  
  
    ' Create a new pie theme  
    Dim pieTheme As MapInfo.Mapping.Thematics.PieTheme = New _  
        MapInfo.Mapping.Thematics.PieTheme(map, lyr.Table, "Pop_Native", _  
            "Pop_Asian", "Pop_Other")  
  
    ' Create an object theme layer based on that pie theme  
    Dim thmLayer As ObjectThemeLayer = New ObjectThemeLayer("World _  
        Pop Growth Rate", Nothing, pieTheme)  
  
    'Add object theme to the map's layer collection.  
    map.Layers.Add(thmLayer)  
End Sub
```

有关主题的详细信息，请参阅[第 14 章：使用主题和图例](#)。

使用日期和时间的标签表达式

表达式是标注修饰符的属性，该标注修饰符用于修改用于生成标注的默认属性。代码示例中的表达式用于显示此形式的标签：Date/Time: <日期和时间>。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_OverrideLabelModifier(ByVal _  
    modifier As OverrideLabelModifier)  
    modifier.Name = "'Date/Time: ' + DateTimeToString( dateTimeColumn,  
        'm/d/yyyy hh:mm tt')"  
End Sub
```

关于标注的详细信息请参阅[Labels](#)。

InfoTips 表达式示例

InfoTips 是工具悬停在 Feature 上时显示的文本项。表达式可以与标注所用方式类似的方式用于生成 InfoTips 的文本。使用 MapTool.SetInfoTipExpression 静态帮助程序函数来设置 InfoTip。该函数负责创建 InfoTip 哈希表并将图层条目添加到哈希表（如果哈希表还不存在）。可以设置每个工具启用或禁用 InfoTips。

以下示例将生成两行文本标注来显示 Table 别名和光标悬停的对象质心的 X 或 Y 坐标。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_HowDoICreateExprForInfoTip(ByVal  
mapControl1 As MapControl)  
    MapTool.SetInfoTipExpression(mapControl1.Tools.MapToolProperties,_  
        CType(mapControl1.Map.Layers(0), FeatureLayer), "@TableAlias + _  
        char(13) + 'Centroid X:' + MI_CentroidX(obj) + ' Y:' + _  
        MI_CentroidY(obj)")  
End Sub
```

有关 InfoTips 的详细信息，请参阅[图层控制](#)。

从 DBMS 访问数据

MapXtreme 提供空间服务器访问功能。这是一项强大的功能，开发人员可使用它连接存储在空间服务器（例如，在 Microsoft SQL Server 或 Oracle Spatial 数据库上运行的 MapInfo SpatialWare）中的实时数据。空间服务器允许公司将其地图数据存放在企业数据库中，以便集中管理并获得更高的安全性。SpatialWare 这样的空间服务器为公司的空间数据提供了高级查询处理和提高的性能。

在本章中：

- ◆ 访问远程空间数据..... 207
- ◆ 通过 .TAB 文件访问远程表..... 207
- ◆ 不使用 .TAB 文件访问远程表..... 207
- ◆ 使用 X/Y 列映射 DBMS 数据..... 208
- ◆ 访问 Oracle 数据库中的数据..... 208
- ◆ 访问 MS SQL Server 中的数据..... 211
- ◆ DBMS 连接字符串格式..... 214
- ◆ 在服务器表查询中定义可制图的表..... 216
- ◆ 访问属性数据..... 217
- ◆ 性能问题..... 218
- ◆ 运用缓存..... 218
- ◆ MapInfo_MapCatalog..... 220
- ◆ 向 MapInfo_MapCatalog 添加行..... 222
- ◆ 每记录样式..... 226
- ◆ 故障诊断..... 227

访问远程空间数据

可以使用具有不同 DBMS 服务器的 MapXtreme 访问数据。这些服务器包括：

- Microsoft Access 2007 和 Excel 2007
- Microsoft Access 2003
- Oracle 11G (11.1.0.6.0 和 11.1.0.7.0)
- Oracle 10G、10GR2,
- Microsoft SQL Server 2012 (配有 SQL Native Client 11)
- Microsoft SQL Server 2008 (配有 SQL Native Client 10)
- MapInfo SpatialWare® 4.9/4.9.2 for Microsoft SQL Server 2005
- MapInfo SpatialWare® 4.9.2 for Microsoft SQL Server 2008 (32 位版和 64 位版)

可以使用 `MapInfo.Data` 命名空间中的 `TableInfoServer` 类，从 DBMS 中的数据添加表。

有关添加空间数据的详细信息包括在以下章节中。

通过 .TAB 文件访问远程表

MapXtreme 应用程序可以“实时”访问 DBMS 数据，或可以打开 MapInfo Professional 链接的表。但是，链接的表是只读的，不能通过应用程序刷新。表中数据实际上是来自远程数据库的，不会反映位于本地链接的版本中的数据。

可以创建 .TAB 文件以提供对远程数据的访问。要使用 MapInfo Professional 生成 .TAB 文件，请选择“文件”>“打开 DBMS 表”。

.TAB 文件是文本文件，可以使用任何文本编辑器创建 .tab 文件。创建 .tab 文件以后，访问该文件的方式与通过 Catalog 对象或通过 Workspace 管理器以编程方式访问任何其他 MapInfo .TAB 文件的方式相同。

不使用 .TAB 文件访问远程表

应用程序不需要 .TAB 文件就可以访问远程数据。以下代码示例说明了该过程。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfoServer(ByVal connection As _
    MIConnection)
    ' Note: Do not specify any columns. These are determined
    ' dynamically from the query

    Dim ti As TableInfoServer = New TableInfoServer("Provinces")
    ti.ConnectString = "SRVR=ontario;UID=mapx;PWD=mapx"
    ti.Query = "Select * From Provinces"
```

```
ti.Toolkit = ServerToolkit.Oci

ti.CacheSettings.CacheType = CacheOption.Off ' On is the default
Dim tbl As Table = connection.Catalog.OpenTable(ti)

End Sub
```

使用 X/Y 列映射 DBMS 数据

可以访问来自 DBMS 表的数据，该 DBMS 表具有 X/Y 坐标。必须创建 MapInfo_MapCatalog，将表注册为 SpatialType 4.0 并指定两个列名作为坐标。这两个列应该在该表上被编入索引。通过 ODBC 连接到 DBMS，然后在查询中将新列指定为 "Obj" 或 "MI_Geometry"。

访问 Oracle 数据库中的数据

要从 MapXtreme 应用程序连接到 Oracle 数据库，必须安装 Oracle OCI 连接客户端并授予适当的权限。请参阅 Oracle 文档以获取详细信息。

几何体转换

下表显示了从 MapXtreme 对象到 Oracle Spatial (SDO_GEOMETRY) 的转换过程。

从 MapInfo	到 Oracle
NULL 几何体	NULL
Point	1 POINT
MultiCurve (IsLegacyLine = true)	2 LINESTRING 几何体包含一行字符串
Polygon	3 POLYGON 几何体包含一个多边形。
FeatureGeometryCollection	4 Collection 几何体是元素的异种集合。
MultiPoint	5 MULTIPONT
MultiCurve	6 MULTILINESTRING 几何体具有多行字符串。
MultiPolygon	7 MULTIPOLYGON 几何体具有多个多边形。
Ellipse	NULL
LegacyArc	NULL

从 MapInfo	到 Oracle
Rectangle	NULL
LegacyText	NULL
RoundedRectangle	NULL
PieTheme, BarTheme	NULL

下表描述从 Oracle (GTYPES) 到 MapInfo Spatial 对象的转换。

从 Oracle GTYPES	到 MapInfo
0 *UNKNOWN_GEOMETRY (Spatial 忽略此几何体。)	
1 POINT 几何体包含一个点。	Point
2 LINESTRING 几何体包含一个线字符串。	MultiCurve
3 POLYGON 几何体包含一个多边形。	MultiPolygon
4 *Collection 几何体是元素的异类集合。	FeatureGeometryCollection
5 MULTIPPOINT 几何体具有多个点。	MultiPoint
6 MULTILINESTRING 几何体具有多个线字符串。	MultiCurve
7 MULTIPOLYGON 几何体具有多个多边形（多外部边界）。	MultiPolygon

*从 MapInfo 对象格式转换或转换到 MapInfo 对象格式时，可能会丢失一些数据。如果已为 NULL，这些类型则被解释（如有可能）为单点 SDO_POINTTYPE 值。它们“抓取”将被解释为 NULL 几何体的有序数组的第一个点。

对 Z 和 M 值的 Oracle 支持

MapXtreme 支持读取和写入带有 Z 和 M 值的 Oracle GTYPE。要确定 Z 和 M 的显示情况和顺序，必须先检查该表 USER_SDO_GEOM_METADATA 中的 DIM_INFO 数组。MapXtreme 检查以下尺寸名称（区分大小写）：

- 对于 Z 尺寸：“Z”、“Elevation”、“Depth” 和 “Z Dimension”
- 对于 M 尺寸：“M”、“Measure” 和 “M Dimension”

表如果包含 M 和/或 Z 值，现在则会针对所显示尺寸返回包含数据的 FeatureGeometry 对象。FeatureGeometry 实例如插入或更新到 Oracle 表，该实例将保留 Oracle 表支持的新几何体 (XYZM) 四个尺寸中的每一个尺寸。如果几何体包含 Oracle 表不支持的尺寸，在插入/更新操作期间，这些尺

寸的值将被忽略。如果几何体不包含 Oracle 表支持的尺寸，在插入/更新操作期间，程序将提供 NULL 值以代替缺失的尺寸。例如，如果要向使用尺寸 X、Y 和 M 定义的表中插入没有 Z 或 M 值的几何体，表中保存的 M 值将为 NULL。

任何这类的插入/更新操作能否成功，也取决于其他的服务器端验证（其中包括的明确的列约束条件和值验证必须遵循 SDO_GEOM_METADATA 系统表中指定的尺寸边界）。

SDO_Geometry Arc 和 Circle 转换

可将圆和圆弧解析为具有 25 段/360 度圆的 MultiCurves。

可视化不可转换的 Oracle 对象

MapXtreme 应用程序不能转换的 Oracle 空间对象在 SDO_Spatial 点的位置生成带有默认样式（黑色的星型标记）的 Point 对象，或生成纵坐标数组中的第一个 SDO_Spatial 纵坐标。这样就可以在不可转换的对象所属的正确图形区域中可视化这些对象。不可转换对象的示例是：用户定义的对象 GTypes 0、4、5，或包含无法识别的 GTypes、ETypes 的无效 SDO_geometries，或解释。使用 SDO_VALIDATE_GEOMETRY()，第二个类也会失败。

质心支持

MapXtreme 应用程序使用 SDO_POINT 作为多边形的质心值。质心功能用于定位标注，此外还影响对象的工具选择。如果点位于区域内，则 Oracle SDO_GEOMETRY.SDO_POINT_TYPE 字段（如果非 NULL 的话）将被解释为图元质心。如果点位于区域外，则始终计算它的质心。



当前在 MapXtreme 中没有设置区域图元质心的方法或工具，但可以读取并使用存储的质心。

Oracle Spatial 引用支持 (SRID)

可使用空间引用系统定义 Oracle SDO_GEOMETRY 列。通过在 USER_SDO_GEOM_METADATA 中提供 Oracle SRID，还通过在存储的 SDO_GEOMETRY 值中指定 SRID 来执行此操作。如果表包含具有指定的 SRID 的 Oracle Spatial 列，则 MapXtreme 应用程序可以查询并正确解释数据。

MapInfo_MapCatalog 必须包含与数据的 SRID 中指示的字符串相同的 MapInfo Professional CoordSys 字符串，因为在当前用于解释和更新数据的 MapInfo_MapCatalog 中就是 Coordsys。

如果 Spatial 列未包含 SRID 值（该值为 NULL），则 MapXtreme 应用程序还能够通过在 MapCatalog 中定义的 MapInfo Professional Coordsys 来解释数据。

在将使用纬度/经度坐标系 (Geodetic Data) 的表加载到 Oracle Spatial 时，验证所有几何体坐标是否位于经度 (-180,180) 和纬度 (-90, 90) 之间非常重要。超出该范围的 Geodetic 数据坐标在 Oracle Spatial 中不受支持，并且可能引起问题。可以在加载前使用 MapInfo Professional 检查数据；或可以在将数据加载到 Oracle Spatial 后，使用 Oracle Spatial SDO_Geom.VALIDATE_LAYER() 函数检查数据。

OCI 连接对话框

`MapInfo.Data.DBMSConnectionCollection` 类通过订阅 `ConnectionFailedEvent` 事件处理程序支持 `ConnectionFailed` 事件。激发后，该事件会显示 `OCILoginDlg` 以使用户可以更改登录信息并再次重试到 Oracle 数据库的连接。该句柄专门用于 MapXtreme OCI 工具包。还可通过 `MapInfo.Windows.Dialogs` 命名空间中的 `LoadMapWizard` 类获得该事件。

访问 MS SQL Server 中的数据

MapXtreme 支持 Microsoft SQL Server 2005、SQL Server 2008 和 SQL Server 2012 中的数据。以下信息可用于 SQL Server 2008。

支持 SQL Server 2008

MapXtreme 支持读写 Microsoft SQL Server 2008 中的数据，其中包括空间数据类型 `GEOMETRY` 和 `GEOGRAPHY`，以及这两种空间格式的 M 值和 Z 值支持。

要访问 SQL Server 2008 中的数据，MapXtreme 需要 SQL Server Native Client 10。然后，数据将进行处理，其处理方式类似于处理 MapXtreme 支持¹ 的其他远程数据库管理系统中的数据。可以使用 `MapInfo.Data.TableInfoServer` 类来定义连接串和 SQL 语句，以便在远程表上执行。MapXtreme 内部使用 ODBC 访问远程数据库。

下表显示在 SQL Server 2008 提供特定对象类型的情况下，对象在 MapXtreme 中的处理方式。

SQL Server 2008 Spatial <code>GEOGRAPHY</code> 或 <code>GEOMETRY</code>	MapXtreme FeatureGeometry
Sql Server 2008 Spatial <code>GEOGRAPHY/GEOMETRY</code>	<code>FeatureGeometry</code>
Point	<code>Point</code>
LineString	<code>MultiCurve</code>
Polygon	<code>Multipolygon</code>

1. 有关完整的列表，请参阅第 30 页中的 安装要求。

SQL Server 2008 Spatial GEOGRAPHY 或 GEOMETRY	MapXtreme FeatureGeometry
MultiPoint	MultiPoint
MultiLineString	MultiCurve
MultiPolygon	MultiPolygon
GeometryCollection	FeatureGeometryCollection
仅包含 Points 和/或 MultiPoints 的 GeometryCollection	MultiPoint
仅包含 LineStrings 和/或 MultiLineString 的 Geometrycollection	MultiCurve
仅包含 Polygons 和/或 MultiPolygons 的 Geometrycollection	MultiPolygon
一个 EMPTY GEOMETRY/GEOGRAPHY, 即 Point empty	NULL

此表显示 MapXtreme FeatureGeometry 重新写入 SQL Server 2008 的方式

MapXtreme FeatureGeometry	SQL Server 2008 Spatial GEOGRAPHY 或 GEOMETRY
Point	Point
MultiPoint	MultiPoint
仅包含一个 Point 的 MultiPoint	Point
MultiCurve	MultiLineString
仅包含一个 Curve (由两个点构成) 的 MultiCurve	LineString
Multipolygon	MultiPolygon
FeatureGeometryCollection	GeometryCollection *
Rectangle	NULL
RoundedRectangle	NULL
Ellipse	NULL

MapXtreme FeatureGeometry	SQL Server 2008 Spatial GEOGRAPHY 或 GEOMETRY
LegacyArc	NULL
LegacyText	NULL

* GeometryCollection 可以包含以下类型中的任何一项或全部项：MultiPoint、MultiLineString 和 MultiPolygon。

SQL Server 2008 提供新型的日期和时间信息。下表显示日期和时间类型映射到 MapXtreme 日期和时间类型的方式。

SQL Server	MapXtreme
Date	Date
Time	Time
DateTime	DateTime
SmallDateTime	DateTime
DateTime2	DateTime
DateTimeOffset	不支持

SQL Server 2008 中的空间表必须在 MapInfo_MapCatalog 中注册，以便 MapXtreme 能理解所读取的内容。

MapCatalog 提供四种可表示 SQL Server 2008 表的新 spatialcolumn 值：

- 17.x for GEOMETRY（无 M 和 Z 值）
- 18.x for GEOGRAPHY（无 M 和 Z 值）
- 20.x for GEOMETRY（有 M 和 Z 值）
- 21.x for GEOGRAPHY（有 M 和 Z 值）。

可使用 MapInfo Professional 或 EasyLoader Data 上载数据，也可以使用 MapInfo Professional 使现有数据可制图，此操作会在 MapCatalog 中创建一个条目。请参阅 [MapInfo_MapCatalog](#) 以了解有关 MapCatalog 的更多信息。

MapXtreme 支持在 MapInfo Professional（表版本 900、950 和 1000）和 EasyLoader 中创建的 SQL Server 2008 表。

DBMS 连接字符串格式

ODBC 连接字符串格式

ODBC 连接字符串的格式由多个使用分号 (;) 分隔的子句定义。每个子句都具有格式：**Key=Value**。重要的键在以下列出。

关键字	描述
DLG=	<p>控制连接对话框显示的数字：</p> <p>0 - 禁用连接对话框。</p> <p>1 - 显示连接对话框。</p> <p>2 - 仅在需要时显示连接对话框（即，不是所有需要的信息都已提供时）[默认]。</p> <p>① 使用对象池的 ASP.NET 应用程序必须在用于 ODBC 的连接字符串中具有 DLG=0 子句。</p>
DSN=	<p>指定 ODBC 数据源名称。</p> <p>注意：如果使用 DSN= 语法键，则指定的名称必须与用户系统上使用的数据源名称匹配。注意：不同的用户可能使用不同的名称来引用相同的数据源。如果无法预先了解要使用的数据源名称，则使用 DRIVER= 语法键，而不使用 DSN= 语法键。</p>
DRIVER=	<p>指定已安装驱动程序的正确驱动程序名称。用来替代 DSN= 语法键。</p> <p>示例：</p> <p><code>DRIVER={SQL Server}</code></p> <p>① 它不应用于 Oracle Spatial。</p>
UID=	必要时指定数据源需要的 UserId 。
PWD=	必要时指定数据源的用户口令。对于要匹配的两个字符串，口令不必位于连接字符串中。如果两个表位于相同的数据库，则连接字符串相同。
<p>① 连接属性 / 参数不必按顺序，且可以使用对话框从现有的连接池获得连接以避免冗余的连接。 在 API 的先前版本中，如果每次使用对话框连接到相同的数据库，或如果没有按照文档中的顺序对连接字符串中的连接关键字进行排序，则连接不能共享且会获得多个连接。</p>	

Web 应用程序中的 ODBC 图层和对象池

通过 ODBC 将远程图层添加到使用对象池的 ASP.NET 应用程序时，确保在连接字符串中具有 DLG=0 子句。这将避免显示可能超时的不必要用户和口令对话框。这适用于 TAB 文件和工作空间。下面的连接字符串示例显示了突出显示的 DLG=0 子句。

```
<ConnectionString>DRIVER={SQL Server};DATABASE=Devel;Server=Paladin;  
UID=devel;PWD=devel;QuotedID=True;Trusted_Connection=False;  
Network=DBMSSOCN;Address=PALADIN,1433;DLG=0</ConnectionString>
```

在使用 TableInfoServer 且对象池已打开时，要访问 SQL Server，则在连接字符串中指定 "DLG=SQL_DRIVER_NOPROMPT" 以避免抛出 MapInfo.Data.TableException：无法打开表。

Oracle Spatial 连接字符串格式

这些是 Oracle Spatial 关键字。该字符串由多个使用分号 (;) 分隔的子句定义。每个子句都具有格式：Key=Value。重要的键在下表中列出。

关键字	描述
SRVR=	反映在 Oracle Net8 EasyConfig 实用程序中设置的服务器的服务名称。Oracle 连接需要该值，但该值对 ODBC 连接不适用。
UID=	必要时指定数据源需要的 UserId。
PWD=	必要时指定数据源的用户口令。

样本连接字符串

此处是用于 Oracle Spatial、Microsoft SQL Server 2008 和 SpatialWare ODBC 驱动程序的样本连接字符串。

Oracle Spatial 连接字符串：

```
UID=george;PWD=password;SRVR=OracleSpatial19i
```

Microsoft SQL Server 2008 连接字符串：

```
DRIVER={<driver>};  
SERVER=<server>;UID=<uid>;PWD=<pwd>;Database=<database>
```

where <driver> for SQL Server Spatial should be the most current available, SQL Server Native Client 10.0.

SpatialWare for SQL Server 字符串：

```
DRIVER={SQL SERVER};  
SERVER=ServerName;UID=Troll;PWD=secret;Database=GEORGETOWN
```

在服务器表查询中定义可制图的表

为服务器表指定的查询定义来自 DBMS 的数据的结果集，表示正在添加表中的数据。可以将相当复杂的查询公式化以执行强大的服务器端分析，用于在 MapXtreme 中定义可制图的表。MapXtreme 应用程序在内部使用该查询以访问数据。

MapXtreme 根据访问地图数据的查询以及选择/键查询生成几个内部查询。从中选择几何体列的表必须注册到服务器上的 MapInfo MapCatalog 中。MapXtreme 需要此操作以获取某些关于几何体列的元数据，例如坐标系、空间类型和默认样式。

要根据查询顺序定义可制图的表，则该查询必须包含几何体列和键列。有时候，对于有关小数据集更复杂的查询（其中空间索引或空间谓词将导致该查询失败），可以指定 TableInfoServer.MbrSearch=false 以启用要映射的结果。

几何体列

如果未指定 MapXtreme 应用程序可以识别的几何体列，则将打开该表，但不能将它添加到地图中（该表是不可绘制的）。MapXtreme 通过在 MapCatalog 中查找表的几何体列并检查该列的结果集数据类型来确定几何体列。一般可通过伪列名 "Obj" 来引用几何体列，也可用几何体列的特定列名来对其进行引用。需要使用这种形式来引用 X/Y 可制图的图层的几何体列。可以通过任何服务器支持的几何体函数/表达式来指定几何体列。

示例

```
Select Obj 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
```

Oracle sdo_buffer 示例：

```
Select mi_prinx, mdsys.sdo_geom.sdo_buffer(geoloc, (select diminfo from
sdo_geom_metadata where table_name = 'ALINE'), 20) from aline where prinx
= 1
```

Oracle 构造函数示例：

```
Select 1 "mi_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
```

SQL Server 2008 Spatial 函数示例：

```
select location_id, geography::Point(lat, long, 4326 /*WGS84*/) as geog
from dbo.store_locations
```

Spatialware 函数示例：

```
select sw_member, ST_Buffer(sw_geometry, 66.0, 0.1) from rdbssdata
select ST_Overlap(flood100.sw_geometry, lake.sw_geometry) from flood100,
lake where ST_Overlaps(flood100.sw_geometry, lake.sw_geometry)
```

键列

键列必须在查询中返回以使它作为表打开。即，启用您的 **MapXtreme** 应用程序标识结果集中的每一行以对图层执行添加底纹、选择和标注操作。

在绝大多数情况下，不需要在查询中指定键列。

为了唯一地引用结果集中的行，**MapXtreme** 应用程序可以查找并决定要使用的最佳键列，然后可将它们添加到查询中（如果该查询不存在键列）。在绝大多数情况下，这是主键/唯一的索引。

对 Oracle Spatial 表，可以使用 **MI_PRINX**。

对于某些查询，**MapXtreme** 应用程序可能无法标识该键。在有关视图或同义词的查询中会出现这种情况。视图或同义词必须出现在 **MapInfo MapCatalog** 中。在绝大多数情况下，它们还必须使用基础 **Spatial** 索引系统按需进行注册。由于 **MapXtreme** 不能决定视图和同义词上面的键，所以必须提供一种机制来允许应用程序开发人员/查询编写者标识位于结果表中的键列。该键列必须是单一列并且在结果集中必须是与其他列不同的值。要标识将用作键列的那个列，可以指定列别名 **prinx** 或 **mi_prinx**（例如，`select custid mi_prinx, custname, Obj from mycust`）。

示例

```
Select customer_id mi_prinx, obj from customer_view
```

列别名“**mi_prinx**”用于标识和使用作为该表的键列的 **customer_id** 列。另外，可以在 `create view` 语句中为需要的键列指定别名，从而自动为有关视图的任何查询标识键列。

示例

```
Create view customer_view as select customer_id mi_prinx, geoloc from
customer
```

通常，如果 **prinx** 的列名或列别名，或 **mi_prinx** 在结果集中找到，则该列用作这个表的键列。这使应用程序/查询编写者可以指定需要的键列。

访问属性数据

要使用所有可用数据列，则指定一个查询，例如 `Select * From tablename`。不需要指定 *（星号），可以特别指定想要使用的列。为了获取最佳性能，对您的查询进行了限制，使它仅检索需要的列。

在添加 **DBMS** 表时，出于性能原因，应该仅指定想要在应用程序中使用的查询中的列。这些列是空间列、键列（如果未指定，则将自动添加它们）以及想要标注或从其创建主题的列。可以为任何可制图的表使用伪列“**OBJ**”以引用包含空间数据的列。使用具有 **X/Y** 列的表上的 **MapMarker MDIGEOADDRESS** 列的表需要此操作。

可以使用任何服务器端表达式/函数来指定列。还要避免从实际应用程序的表中选择 *。

下面的代码示例使用 **TableInfoServer** 定义服务器表并使用此定义将图层添加到地图。现在可以根据表中的列标注或放置主题。

VB 示例：

```
Public Shared Sub MapInfo_Data_TableInfoServer(ByVal connection As _  
    MICConnection)  
    ' Note: Do not specify any columns. These are determined  
    ' dynamically from the query  
  
    Dim ti As TableInfoServer = New TableInfoServer("Provinces")  
    ti.ConnectionString = "SRVR=ontario;UID=mapx;PWD=mapx"  
    ti.Query = "Select * From Provinces"  
    ti.Toolkit = ServerToolkit.Oci  
  
    ti.CacheSettings.CacheType = CacheOption.Off ' On is the default  
    Dim tbl As Table = connection.Catalog.OpenTable(ti)  
End Sub
```

性能问题

与数据库服务器建立连接可能需要花费几秒钟。这是一次性开销，在第一次打开表时发生。

地图显示速度取决于从服务器检索了多少数据。在某些情况下，从服务器显示地图明显比从本地文件显示地图要慢。显示速度还取决于 **MapXtreme** 应用程序是否已缓存将要显示的地图图元。

运用缓存

了解如何在 **MapXtreme** 中运用缓存来提高应用程序的性能。下面的章节说明了什么是缓存，如何在 **MapXtreme** 对象模型中使用缓存，以及 **TableInfoServer** 对象的 **CacheSettings** 属性。

什么是缓存？

为了避免读取本地文件，应用程序可以从远程数据库访问 **MapXtreme** 图元。为了避免每次对地图进行操作时都要从数据库中读取这些记录，**MapXtreme** 可以把这些记录临时存储在缓存中。这可以限制应用程序和远程数据库之间的调用次数。可对服务器表中的记录（即绘图、主题、标注等）进行缓存以提高应用程序性能。在读取服务器表数据读取并将其绘制到 **Map** 窗口时，可在内部对其进行缓存。所有后续重绘都从缓存中读取，而不是进入服务器数据库获取同样的数据。缓存能够显著提高重绘性能。

开发人员可以使用若干设置来定制缓存的使用。通过为 **TableInfoServer** 对象的 **CacheSettings** 属性指定值来添加服务器表并且该服务器表是默认的 ON 时，可以启用缓存。

缓存如何工作

对于每个缓存的记录，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。对于记录很多和（或）记录很大（例如，属性数据每个记录的字节数）的表，进行缓存可能会使用很大的内存。如果应用程序试图缓存过多数据，则可能需要使用太多虚拟内存，这会降低性能。应用程序应有选择地决定如何利用缓存。MapXtreme 提供了各种用于控制缓存的机制。

TableInfoServer 对象和 CacheSettings 属性

将表添加到地图时，默认启用缓存，但可能需要使用 TableInfo 对象的 CacheSettings 属性进行进一步配置。此属性有四个可能的值：ON、OFF、ALL 和 USER，对于 TableInfoServer，默认为 ON，对于其他 TableInfo 对象，默认为 OFF。

参数	描述
OFF	值 'Off' 表示表将根本不使用缓存。所有数据操作将直接进入数据库服务器。
ON	<p>缓存被启用，并且表自动执行基于地图视图（中心或缩放）的缓存。用户还可以通过缓存约束对象来控制缓存。</p> <p>缓存的表以最能改善标准地图操作的方式来维护记录缓存。缓存维护至少包含显示在每个 Map（该表位于其中且可见）当前窗口中的所有记录。一旦初始地图窗口已经缓存，平移和缩放操作（完全位于缓存的初始扩展内）就访问缓存的记录且不必查询数据库。如果平移或缩放操作位于缓存区域外，则表将新的地图窗口 MBR（视图）添加到缓存，然后从数据库服务器获得缺少的记录并将这些记录添加到缓存。开始时不丢弃旧的地图视图，相反还维护先前地图视图的内部历史记录。要避免缓存增长过大，可以在表的缓存上设置一些控制来决定何时丢弃旧的缓冲视图（地图窗口 MBR 区域）。这些控件是 CacheParameters 对象的属性，可在最初打开表时进行设置。这允许开发人员做出一些限制，例如缓存所用内存或磁盘空间的最大容量，在历史记录中维护先前地图窗口视图的最大数目，在缓存中维护记录的最大数目，和（或）缓存历史记录中允许保留的旧地图窗口视图的最长时间。可以单独或组合使用这些限制以适应应用程序所需提供的最优缓存管理。</p> <p>ON 是 TableInfoServer 的 CacheParameter 设置的默认设置。对于其他 TableInfo 数据源，默认设置为 OFF。例如，默认情况下不缓存 TAB 文件。</p>

参数	描述
USER	<p>LayerInfo CACHE 参数的值 USER 表示应用程序创建了缓存，但只有应用程序开发人员指定的那些记录放入缓存中。这些机制可用于指定要将哪些记录放入缓存，记录类型包括 BoundConstraint、FeaturesConstraint 和 AllFeaturesConstraint 对象。约束一词意味着这些对象正在约束要包括特定记录的缓存。BoundsConstraint 对象可用于将全部记录放入 MBR 图元与 MBR 约束相交的缓存中。</p> <p>FeaturesConstraint 对象可用于将特定记录添加至缓存。例如，如果正在执行一个分析，它涉及多个步骤和（或）对图元的 Feature 和 RowValues 的读取，可能在从 Layer.Search、Layer.SearchWithinDistance 等返回的一组图元上进行，那么在分析期间将这些记录放入本地缓存并在分析完成时删除就非常有用。FeaturesConstraint 提供了这个功能。如果应用程序将要执行可能击中每个记录的密集分析操作，则可能需要这个功能临时缓存用于图层的整个数据集。通过使用 AllFeaturesConstraint 可以实现此功能。这些缓存约束对象还可在将缓存设置为 ON 时使用。在这种情况下，它们可以将记录添加到缓存，但该缓存对先前地图窗口视图的缓存历史无效。这些缓存约束还可在将缓存设置为 OFF 或 ALL（缓存不起作用）时使用。</p> <hr/> <p>i 该约束对象对非服务器表无效。</p>
ALL	整个表被缓存。使用该选项，表的数据只需从服务器检索一次，以后就可以从本地进行访问。要刷新缓存中的数据，请对该表使用 Refresh 方法。

如果试图缓存过多数据或过多表，则会强制使用虚拟内存，这样可能会降低性能。

MapInfo_MapCatalog

为了显示地图上的数据，**MapXtreme** 应用程序需要访问特殊表（也称为 **MapInfo_MapCatalog**）。每个数据库必须创建一个目录才能将数据库中的任何表作为 **MapXtreme** 应用程序中的地图图层进行查看。**MapCatalog** 必须在每个想要从数据库进行访问的可制图的表中包括关于空间列的信息。**MapInfo EasyLoader** 实用程序在将表上载到数据库时自动在 **MapInfo_MapCatalog** 中插入适当的行。

应用程序可以使用服务器上已存在的 **MapInfo_MapCatalog**。（各种 **MapInfo** 客户端应用程序可共享同一个目录）。如果服务器上没有 **MapInfo_MapCatalog**，则必须创建一个。**MapXtreme** 支持将各个图元的样式信息存储在远程数据库中。

将空间数据加载到 DBMS

如果空间数据是 **MapInfo** 表形式的数据，则可以将那些数据导入到 DBMS 数据库。

要将数据载入 Microsoft SQL Server、MapInfo SpatialWare for SQL Server 和 Oracle Spatial，请使用 MapInfo EasyLoader，它是和 MapInfo Professional 一起发布的，可以从以下网址获取：www.pbinsight.com。如果没有 MapInfo_MapCatalog，EasyLoader 实用程序将在上载表时自动创建 MapInfo_MapCatalog。

手动创建 MapInfo MapCatalog

如果您不是 MapInfo Professional 或 EasyLoader 用户，您或数据库管理员必须手动创建 MapCatalog，具体操作如下。每个服务器/数据库只能创建一次 MapCatalog。

1. 在可制图的表所属的特定数据库中创建用户 MAPINFO。
2. 在该数据库中创建表 MAPINFO_MAPCATALOG。

Create Table 语句需要等价于下列 SQL Create Table 语句：

```
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,
    VIEW_X_LL Float,
    VIEW_Y_LL Float,
    VIEW_X_UR Float,
    VIEW_Y_UR Float,
    COORDINATESYSTEM Char(254),
    SYMBOL Char(254),
    XCOLUMNNAME Char(32),
    YCOLUMNNAME Char(32),
    RENDITIONTYPE INTEGER,
    RENDITIONCOLUMN CHAR(32),
    RENDITIONTABLE CHAR(32),
    NUMBER_ROWS INTEGER
)
```



表的结构与这个语句完全相同十分重要。可以进行的惟一替换是支持 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. 在 TABLENAME 和 OWNERNAME 上创建惟一的索引，因此对于每个所有者只有一个表可以是可制图的。

```
create unique index mapcat_i1
on mapinfo.mapinfo_catalog (OwnerName,TableName)
```

4. 对于 MAPINFO_MAPCATALOG 的 Grant Select、Update、Insert 和 Delete 特权。这允许用户使表成为可制图的。

```
grant select, insert, update, delete on mapinfo.mapinfo_mapcatalog to
public
```

向 **MapInfo_MapCatalog** 添加行

对于每个想要从应用程序进行访问的空间表，需要将一行添加到 MAPINFO_MAPCATALOG 表。如果没有使用 MapInfo Professional 来管理 MapInfo_MapCatalog，则必须手动将行添加到 MAPINFO_MAPCATALOG 表。下表说明了每个列的语法和含义：

列名	要指定的值	示例
SPATIALTYPE	<p>MapInfo 空间对象格式</p> <p>1: 在 X/Y 列被索引的 Point 图层 具有 micode (已序列化的 quadtree 键)</p> <p>4: X/Y 列中的 Point 图层</p> <p>5.x: SpatialWare for Oracle</p> <p>6.x: Ingres SQL - 不支持</p> <p>7.x: Sybase SQS - 不支持</p> <p>8.x: Oracle SDO 版本 2 - 不支持</p> <p>x: MapInfo Geocoding DataBlade SpatialWare 点模块</p> <p>x: MapInfo Geocoding DataBlade XY 模块</p> <p>x: SpatialWare IDS/UDO 数据库</p> <p>13.x: Oracle Spatial</p> <p>x: SpatialWare for Microsoft SQL Server</p> <p>17.x: SQL Server 2008 GEOMETRY (无 M 和 Z 值)</p> <p>18.X: SQL Server 2008 GEOGRAPHY (无 M 和 Z 值)</p> <p>20.x: SQL Server 2008 GEOMETRY (有 M 和 Z 值)</p> <p>21.x: SQL Server 2008 GEOGRAPHY (有 M 和 Z 值)</p> <p>空间对象类型</p> <p>x.0: 仅支持点</p> <p>x.1: 仅支持直线</p> <p>x.2: 仅支持区域</p> <p>x.3: 支持所有类型</p> <p>① 该列说明了用来存储和索引数据的“空间对象格式”以及在该列中受支持和不受支持的“空间对象”类型。小数点左边的数字是“空间对象格式”。小数点右边的数字表示了可存储在该列的“空间对象类型”的类型。</p> <p>映射到 <code>MapInfo.GeometryColumn.PredominantGeometryType</code> 和 <code>Has<Line/Point/Region/Text>Geometries</code></p>	<p>14.0 = SQL Server 14.1 14.2 14.3</p>
TABLENAME	表的名称。	STATES
OWNERNAME	表的所有者名称。	BOB

列名	要指定的值	示例
SPATIALCOLUMN	包含空间图元的列的名称（如果有）： SW_GEOMETRY (使用 SpatialWare Type/IDS/UDO 可制图) NO_COLUMN (可通过 X-Y 地图化) MI_SQL_MICODE (使用 MI Code 可制图) 或 IDS/UDO 的名称，或 ST_SPATIAL 数据类型 Oracle 列。 Oracle SDO_GEOGRAPHY 列的名称。	SW_GEOMETRY
DB_X_LL	图层的最小边界矩形的左下角的 X 坐标，使用由 COORDINATESYSTEM 指示的单位（详情如下）。 映射到 MapInfo.Data.GeometryColumn.Bounds	- 360
DB_Y_LL	左下边界上的 Y 值。	- 90
DB_X_UR	右上边界的 X 值。	360
DB_Y_UR	右上边界的 Y 值。	90
VIEW_X_LL	默认视图左下角的 X 坐标。 如果它是第一个打开的表，则仅应用默认视图。 映射到 MapInfo.Data.GeometryColumn.DefaultView	-180
VIEW_Y_LL	默认视图左下边界 Y 值。	-45
VIEW_X_UR	默认视图右上边界 X 值。	180
VIEW_Y_UR	默认视图右上边界 Y 值。	45
COORDINATESYSTEM	表示 MapInfo CoordSys 子句（但一开始不具有关键字 CoordSys）的字符串，该字符串指定地图投影、坐标单位等。对于简单经度/纬度地图，指定 Earth Projection 1, 0。 映射到 MapInfo.Data.GeometryColumn.CoordSys	Earth Projection 1, 0

列名	要指定的值	示例
SYMBOL	<p>MapInfo Symbol 子句（如果图层仅包含点），或 Symbol 子句，它后跟指示线性图元样式的 Pen 子句，而该 Pen 子句后跟指示区域边界样式的其他 Pen 子句，最后又后跟 Brush 子句。</p> <p>映射到 <code>MapInfo.Data.GeometryColumn.DefaultStyle</code></p>	<code>Symbol(35, 0, 12)</code> <code>Pen(1, 2, 0)</code> <code>Pen(1, 2, 0)</code> <code>Brush(2, 255, 255)</code>
XCOLUMNNAME	<p>对于 X/Y 可制图的表，指定包含 X 坐标的列的名称。如果没有这样的列（即，如果该表使用单一空间列而不使用 X-Y 列对），则指定 NO_COLUMN 或保留为空。</p> <p>映射到 <code>MapInfo.Data.SpatialSchemaXY</code></p>	NO_COLUMN
YCOLUMNNAME	<p>对于 X/Y 可制图的表，指定包含 Y 坐标的列的名称，或指定 NO_COLUMN</p> <p>映射到 <code>MapInfo.Data.SpatialSchemaXY</code></p>	NO_COLUMN
RENDITIONTYPE	<p>这指示对象样式信息的应用方式。</p> <p>0 - 表示将对表中的所有对象应用 MapCatalog 的符号字段中所指定的样式。没有有效的每记录样式。将用表的默认样式读取/更新对象。</p> <p>1 - 表示表将采用每记录样式。表中有一个专用于存储表中每个对象样式信息的 MapBasic 字符串表示形式的列（与目前已在 MapCatalog 的 SYMBOL 列中应用的格式相同）。表中的样式列记录在 RENDITIONCOLUMN 中。</p>	0 或 1
RENDITIONCOLUMN	<p>如果 RENDITIONTYPE 为 1，则该字段存储包含样式信息的空间表中的列名称。自动将该列添加到任何针对表的查询，并在更新该对象时对该列进行维护（更新）。用户不应在查询中指定此列，因为在相交或更新语句时可能会引起问题。将此列包括在 select 子句（不包括通配符“*”）中的查询可以通过 Dataset 对象访问值。在样式列中具有 NULL 值的行将具有样式，该样式来自要应用到该对象的 MapCatalog 的 SYMBOL 字段。</p> <p>使用 MIDBType.Style 创建 <code>MapInfo.Column.DataType</code></p>	MI_SYMBOL

列名	要指定的值	示例
RENDITIONTABLE	当前不使用，但保留以供将来使用。	NULL
NUMBER_ROWS	当前由 MapInfo Professional 使用。	NULL

每记录样式

每记录样式支持将图元带到空间数据库实现，该实现已长期在 **MapInfo TAB** 文件中可用。特别是，它允许单一表中的每个几何体具有自己的样式。例如，**Oracle Spatial** 中的单个“公共机构”表可以具有学校、市政大厅、图书馆和警察局，并且每个点类型将使用自己的符号表示（即学校符号用于所有学校）。类似的，**SpatialWare SQL Server** 中的单一道路表可能具有不同的道路类型，如显示为单像素黑色直线的街道、显示为双像素红色直线的辅道以及显示为红色平行直线的十字路口。

要使用每记录样式，必须通过相应的 RENDITIONTYPE、RENDITIONCOLUMN 和 RENDITIONTABLE 设置，以 **MapCatalog** 中的条目来呈现表。

 如果没有这些列，则会对所有对象应用表的默认样式。

Symbol、Pen、Brush 子句语法

如果手动创建 MAPINFO_MAPCATALOG 表以提供对远程空间数据库的支持，则将需要指定符号样式，以及可能的直线和填充样式。

指定点样式

使用 **Symbol** 子句来指定点样式。有三类 **Symbol** 子句：一种用于指定 **MapInfo 3.0** 样式符号，一种用于指定 **TrueType** 字体符号，一种用于指定位图符号。

Symbol 语法	示例
Symbol(shape, color, size)	Symbol(35, 0, 12)
或	
Symbol(shape,color,size,font,fontstyle,rotation)	Symbol(64, 255, 12, "MapInfo Weather", 17, 0)
或	
Symbol(bitmapname,color,size,customstyle)	Symbol("sign.bmp", 255, 18, 0)

指定直线样式

使用 **Pen** 子句来指定直线样式。在 **MapInfo_MapCatalog** 中，可能需要指定两个 **pen** 子句：一个用于指定线性图元的外观，另一个用于指定区域边界的外观。

Pen 语法	示例
Pen(thickness, pattern, color)	Pen(1, 2, 0)

指定填充样式

使用 Brush 子句来指定闭合图元（区域）的样式。

Brush 语法	示例
Brush(pattern,color,backgroundcolor)	Brush(2, 255, 65535)

有关 MapXtreme Styles API 的讨论，请参阅[第 15 章：为地图设置样式](#)。样式图案请参阅[附录 F：样式查找](#)。

文本对象限制

`LegacyText` 对象具有它们自己用来显示样式的方式，与使用 `MI_Style` 列分开。因此，任何形式的文本对象都需要单独对待，它们彼此之间互不相同。嵌入用于任何文本对象的样式并将 `NULL` 值插入到类型列中。

故障诊断

如果在使用 `SpatialWare` 或 `Oracle` 应用程序时遇到问题，则使用下表来帮助分析和解决问题。

问题说明	可能原因	解决方案
表是不可匹配的。	尝试对 <code>SpatialWare</code> 图层进行数据绑定。	<code>SpatialWare</code> 图层目前不支持 <code>AddColumns</code> 。

问题说明	可能原因	解决方案
使用指定的索引没有找到任何对象。	针对不存在的表执行查询。	检查表名是否正确并且大小写正确。还有，该表需要是可制图的。
	在空间查询的结果中未包含任何空间对象。	使用 EasyLoader Upload 实用程序使该表成为可制图的表。
	针对非空间表执行查询。	检查查询以获取可能的语法错误。还要确保查询结果包括在 MapInfo_MapCatalog 的空间列中指定的字段。
地图显示为具有不正确的缩放级别。例如，地图可能缩得太小，不能标识任何地理位置。	DBMS 图层的 MBR 由 MapInfo_MapCatalog 表确定。该表以不同于所需输出的缩放级别在 MapCatalog 结果中进行扩展。	使用 MapInfo Professional MBX 工具 MISETMBR.MBX 编辑 MapInfo_MapCatalog 中的扩展（ DB_X_LL 、 DB_X_UR 、 DB_Y_LL 和 DB_Y_UR ）。

将地图绘制功能增加到 应用程序

使用 `MapInfo.Mapping` 命名空间可以将地图绘制功能添加到应用程序。本章解释了如何使用 `Mapping` 命名空间增强地图绘制应用程序。

在本章中：

◆ <code>MapInfo.Mapping</code> 命名空间简介	230
◆ 基本地图绘制类	230
◆ <code>Layers</code>	232
◆ <code>Labels</code>	236
◆ <code>Adornments</code>	240
◆ 图元样式修饰符	242
◆ 打印地图	244

MapInfo.Mapping 命名空间简介

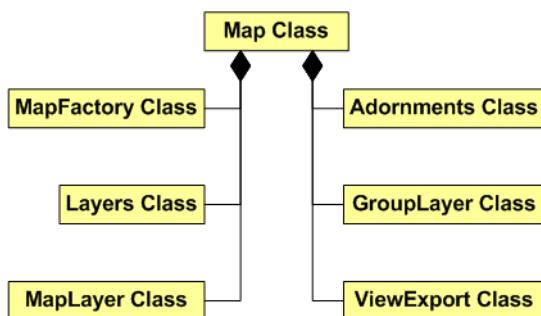
MapInfo.Mapping 命名空间包含了用于创建、显示和导出地图、图层、修饰符、主题、图例和标注的类、接口和枚举。

Map 类是桌面应用程序中地图的顶层对象。每个 Map 对象正好包含一个地图。每个 Map 具有一个 Adornment 集合和一个 Layer 集合。开发 web 应用程序时，Map 对象附加到 MapExport 对象，以便将该地图图像导出到位图或数据流。

如 [Layers](#) 中所述，从根本上讲，地图是一组互相层叠的图层。使用 Mapping 命名空间类，应用程序可按照需要设计用于操控这些图层。

基本地图绘制类

本节讨论了如何在 MapInfo.Mapping 命名空间中使用基本地图类。下图说明了 Map 层次的 UML 表示。



MapExport

MapExport 类用于将 Map 导出到图像。该类的属性指定了图像的各个方面，例如 BorderPen、ExportSize、Format 等等。

MapExport 支持几种不同的图像格式，包括：BMP（默认）、WBMP、WMF、EMF、GIF、J2K、JPG、PNG、PSD、TIFF、TIFFCymk 和 TIFFLzw。出现性能问题时，可以使用 .NET framework API 而不是使用 LEADTOOLS API 导出 BMP、GIF、JPG、PNG 和 TIFF 格式。两种方法的导出速度可能会有所不同，但导出图像的质量相同。

要选择格式，请将 ExportFormat 属性设置为以下任一格式（如 ExportFormat.WindowsPNG）。

① 此外，LegendExport 类还会使用 .NET framework 和 LEADTOOLS API 导出这些格式的文件。

使用 MapExport 的示例

如果正使用 MapControl，则必须在导出地图前将其克隆，具体如下示例所示：

VB 示例：

```
Public Shared Sub MapInfo_Mapping_MapExport(ByVal mapControl1 As MapControl)
    map 对象来源于 mapcontrol 并且通过句柄关联，因此必须克隆
    Dim NewMap As Map = CType(mapControl1.Map.Clone(), Map)
    Dim exportObject As MapExport = New MapExport(NewMap)
    exportObject.ExportSize = New MapInfo.Mapping.ExportSize(2931, 4104)
    exportObject.Format = ExportFormat.Gif
    exportObject.Export("C:\Temp\ExportImage.gif")
End Sub
```

Map

Map 类包含了可以在地图中放入的任何内容。对于桌面应用程序，Map 对象放置在 MapControl 对象中；对于 web 应用程序，Map 对象附加到 MapExport 对象。

Map 具有以下属性：

- Bounds
- Center
- 缩放
- Scale
- Size
- Rotation
- DisplayTransform
- DisplayCoordSys

每个 Map 都具有 Layers 集合，用于保存组成地图的所有图层（请参阅 [Layers](#)），还具有一个包含所有地图修饰的 Adornment 集合。Adornments 包括 Legends、Titles 和 Scalebars（请参阅[图例概述](#)）。

MapFactory

MapFactory 类包含了用于从 Geosets、工作空间和表的列表创建地图的对象。MapFactory 的功能也类似于在特殊会话中创建的所有地图的集合容器，而且 MapFactory 可以跟踪特殊会话中创建的所有地图的集合。

使用 MapFactory 的示例

该示例使用 MapFactory 的 CreateEmptyMap 方法创建空地图，地图大小为 300 乘 300 像素。

```
' Create a new map
Dim map As Map = Session.Current.MapFactory.CreateEmptyMap(New Size(300,
300))
```

MapLoader

MapLoader 类提供了一种机制，用于从 Geoset 文件、XML Workspace 文件或 TAB 文件加载地图的图层。对于每种要加载的地图类型，都有可用于加载地图的 MapLoader 的子类。这些子类包括 MapGeosetLoader、MapWorkspaceLoader 和 MapTableLoader。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_MapLoad()
    ' 生成一个空地图
    Dim map As Map = _
        MapInfo.Engine.Session.Current.MapFactory.CreateEmptyMap(New Size(400, _
            300))
    ' Create a maploader. Make sure that Session.Current.TableSearchPath
    ' points to the folder with the tables in it
    Dim tl As MapTableLoader = New MapTableLoader("ocean.tab", _
        "usa.tab", "mexico.tab", "us_hiway.tab")
    ' Load tables into a map
    map.Load(tl)
End Sub
```

MapViewList、MapView

这些类包含的对象有助于维护 Map 的上一个和下一个视图列表。可以使用 MapViewList 类作为一种简便方法来浏览地图视图的堆栈，使用提供每个地图视图细节（Name、Center 和 Zoom）的 MapView 类显示这些视图。

MapXtreme 在 <MapXtreme Install Directory>\MapInfo\

MapXtreme\7.x.x\Samples\Desktop\Features 下提供了一个 MapViewList 的 VB 示例。

MapControl

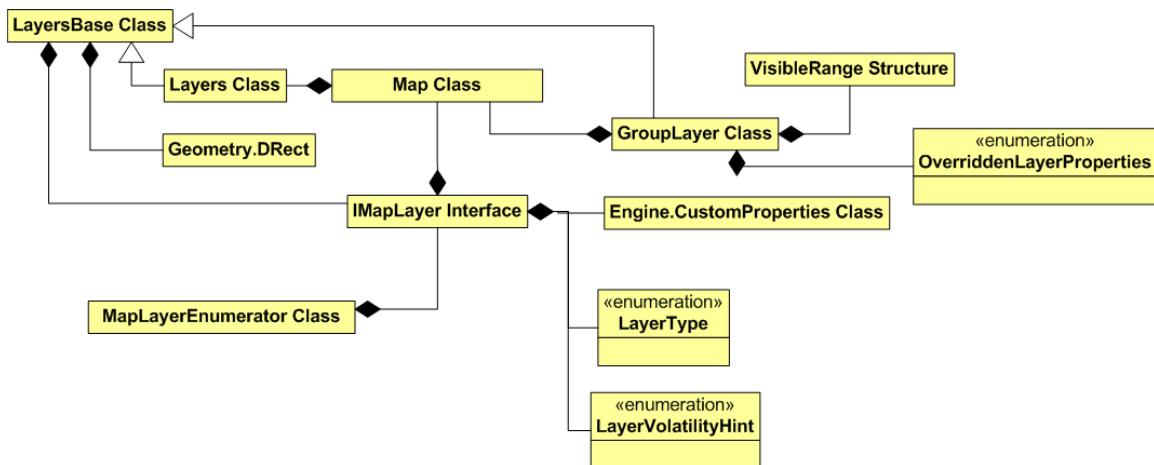
MapControl 类包含了允许用户在屏幕上查看地图的对象。将 MapControl 添加到 Windows Form 来查看地图。MapControl 还保存了 MapTools 集合。MapControl 的版本也存在于 web 应用程序。

用于桌面应用程序的 MapControl 类位于 MapInfo.Windows.Controls 命名空间中。有关桌面 MapControl 的示例，请运行“开始”菜单中的 Workspace 管理器。另请参阅 第 24 章：Workspace 管理器。

用于 web 应用程序的 MapControl 类位于 MapInfo.WebControls 命名空间中。请参阅 Web 控件的体系结构。有关如何创建 Web 应用程序的概览图的教程，请参阅“开始”菜单中“学习资源”的教程部分。

Layers

下一节讨论了 Layers 对象和类。下图是 Layers 层次的 UML 表示。



FeatureLayer

FeatureLayer 是从 Table 显示 Features 的图层。例如表示世界国家的区域对象的图层是 **FeatureLayer**。
FeatureLayers 可以是本地的 .TAB 数据、远程 RDB、无缝或光栅数据。

Layers

每个地图包含了由 **Layers** 类表示的 **FeatureLayers** 的集合。集合中的顺序是绘制图层的顺序。集合类的方法包括 Add、Insert、Move 和 Delete。**Layers** 集合从 **LayersBase** 类派生。通过图层枚举的最好方式是使用 **Layer** 筛选器。

要更改图层，图层首先需要是可编辑的。通过更改其 **LayerControl** 中的设置或通过编程方式更改其 **EditMode** 属性，可以使图层成为可编辑的。图层可以编辑后，该图层中的图元就可以移动、调整大小或删除。



对特殊图层所做的任何编辑会立即生效，因此选择可编辑图层中的图元时要小心操作。

要实现筛选器，请使用 **MapLayerFilterFactory** 类。该类允许创建源于一组常用筛选器的筛选器，例如图层类型或所有可见的图层。要创建自己的筛选器，请编写实现 **IMapLayerFilter** 接口的类。

MapLayer

MapLayer 类是任何图层的基类。该类实现 **IMapLayer** 接口。属性包括 **Enabled**、**VisibleZoomRange**、**Name** 和 **Alias**。使用该类访问一般的图层属性。

UserDrawLayer

UserDrawLayer 是抽象类，允许重载绘制方法绘制自己的图层。它提供了用于将自定义对象增加到 MapControl 顶部的有效方法，即使用 Windows 绘制方法，而不是使用地图坐标创建新图元并将其实际增加到地图中。

ObjectThemeLayer

ObjectThemeLayer 类包含了三种不同主题中的一个主题（饼图、条形图和分级符号）。该图层的行为与其它图层类似，且可以分组，还具有可视性集合等等。

GroupLayer

这是表示一组图层的类，该组图层可以一起移动和开/关。GroupLayer 是由 IMapLayer 组成的 LayersBase 集合。该对象也支持 AnimationLayers 功能。

如果图层是在 VolatilityHint 等于 Animate 的组中，则只有该组内的那些图层需要重绘。如果 Layer 中的 VolatilityHint 等于 CachefPossible 或 Normal，则需要重新绘制所有图层。

LabelLayer

LabelLayer 类负责生成标注并在地图中特定的图层位置绘制它们。请参阅 [LabelLayer](#) 和 [标注图层设置](#)。

GraticuleLayer

该类用于显示地图窗口中网格的经线和纬线。请参阅 [网格图层](#)。

图层筛选器

IMapLayersFilter 和 IMapLayersFilterFactory 接口提供了对图层枚举的支持。

IVisibilityConstraint

IVisibilityConstraint 是确定特殊的对象是否可见的接口。该接口由大量类型实现，包括所有 Layer 类型、LabelModifiers、FeatureStyleModifiers 和 Themes。

代码示例：动画图层

下面的 VB 示例演示了如何设置图层动画。

```
Private Sub btnInitializeObjects_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles btnInitializeObjects.Click  
    Dim Cat As Catalog = MapInfo.Engine.Session.Current.Catalog
```

```
'Create Temp layer
Dim tblInfoTemp As New TableInfoMemTable("Animation")
Dim tblTemp As Table = Cat.GetTable("Animation")
If IsNothing(tblTemp) = False Then 'Table exists close it
Cat.CloseTable("Animation")
End If
tblInfoTemp.Columns.Add(ColumnFactory.CreateFeatureGeometryColumn(Map_
Control1.Map.GetDisplayCoordSys()))
tblInfoTemp.Columns.Add(ColumnFactory.CreateStyleColumn())
tblInfoTemp.Columns.Add(ColumnFactory.CreateStringColumn("Name", 40))
tblInfoTemp.Columns.Add(ColumnFactory.CreateStringColumn("Dept", 15))
tblInfoTemp.Columns.Add(ColumnFactory.CreateIntColumn("Level"))

tblTemp = Cat.CreateTable(tblInfoTemp)

Dim lyr As New FeatureLayer(tblTemp)
MapControl1.Map.Layers.Add(lyr)

'Create Points
Dim pt As FeatureGeometry = New Point(lyr.CoordSys, New DPoint(-76, 42))
Dim cs As New CompositeStyle(New SimpleVectorPointStyle(37, _
System.Drawing.Color.Red, 10))
Dim ftr As New Feature(tblTemp.TableInfo.Columns)
ftr.Geometry = pt
ftr.Style = cs
ftr.Item("Name") = "Kelly"
ftr.Item("Dept") = "Sales"
ftr.Item("Level") = 3
tblTemp.InsertFeature(ftr)

Dim pt2 As FeatureGeometry = New Point(lyr.CoordSys, New DPoint(-119, 34))
Dim cs2 As New CompositeStyle(New SimpleVectorPointStyle(44, _
System.Drawing.Color.Purple, 10))
Dim ftr2 As New Feature(tblTemp.TableInfo.Columns)
ftr2.Geometry = pt2
ftr2.Style = cs2
ftr2.Item("Name") = "Greg"
ftr2.Item("Dept") = "Marketing"
ftr2.Item("Level") = 2
tblTemp.InsertFeature(ftr2)
End Sub

Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
Dim cat As Catalog = MapInfo.Engine.Session.Current.Catalog
Dim tbl As Table = cat.GetTable("Animation")
If IsNothing(tbl) = False Then
'Update the position of the points
Dim si As SearchInfo = MapInfo.Data.SearchInfoFactory.SearchWhere("Name =
- 'Kelly'")
```

```
Dim ftr As Feature = cat.SearchForFeature(tbl, si)
Dim si2 As SearchInfo =
    MapInfo.Data.SearchInfoFactory.SearchWhere("Name = 'Greg'")
Dim ftr2 As Feature = cat.SearchForFeature(tbl, si2)

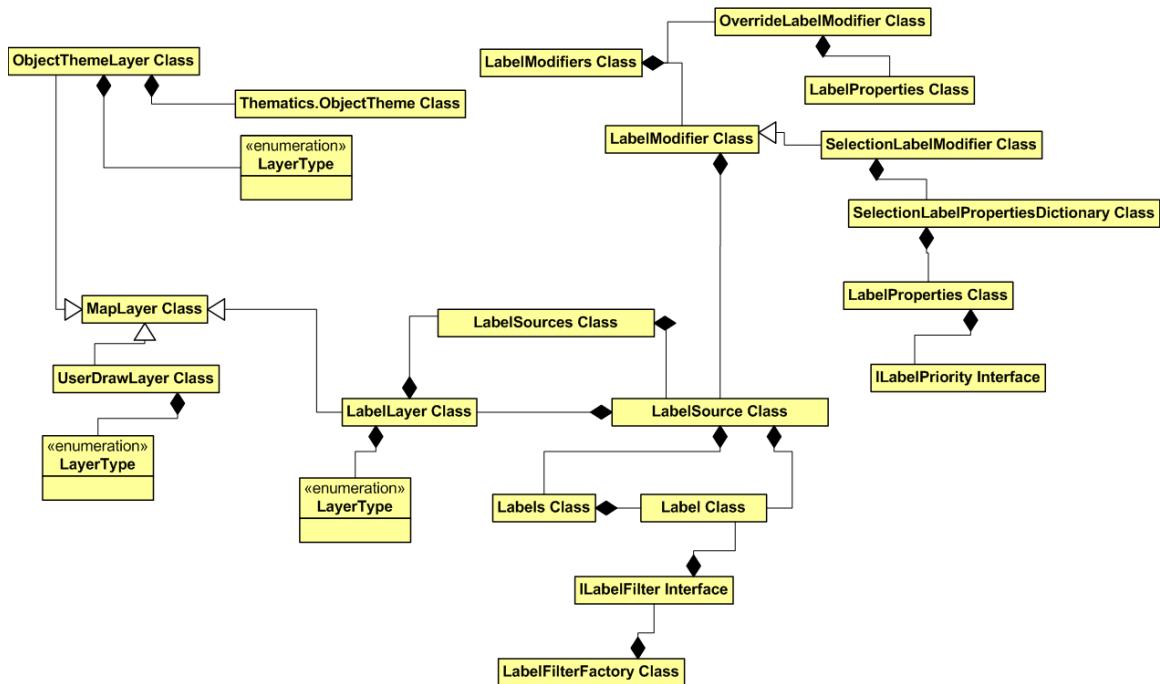
If TimeOfDay.Now.Second Mod 4 = 0 Then
    ftr.Geometry.GetGeometryEditor().OffsetByXY(-5, -25, DistanceUnit.Mile, _
DistanceType.Spherical)
    ftr2.Geometry.GetGeometryEditor().OffsetByXY(0, 25, DistanceUnit.Mile, _
DistanceType.Spherical)
Else
    ftr.Geometry.GetGeometryEditor().OffsetByXY(-10, 0, DistanceUnit.Mile, _
DistanceType.Spherical)
    ftr2.Geometry.GetGeometryEditor().OffsetByXY(10, 5, DistanceUnit.Mile, _
DistanceType.Spherical)
End If
ftr.Geometry.EditingComplete()
ftr2.Geometry.EditingComplete()
ftr.Update()
ftr2.Update()

End If

End Sub
```

Labels

下一节讨论了 **Labels** 对象和类。下图是 **Labels** 层次的 UML 表示。



LabelLayer

LabelLayer 类允许分开对标注和图层进行排序。**LabelLayer** 是一种 **MapLayer**，且其行为也与 **MapLayer** 相同。这类似于 **MapLayer** 支持将 **LabelLayer** 放置在可以放置 **MapLayer** 的任何位置，允许相对于地图中的其它图层定位 Labels。每个 **LabelLayer** 由 **LabelSources** 组成且用作这些对象的集合。

LabelSource

根据数据源和定义如何标注该源的规则，**LabelSource** 类将图形化地理层显示为标注。要绘制 **LabelSource**，则将其添加到地图上的 **LabelLayer**。**LabelLayer** 提供了在 **Map** 内的位置，还提供了管理与其它 **LabelSources** 交互的规则。要使用 **LabelSource** 类，请指定要获得数据的位置的表 (MITableView)，同时指定定义标注文本的表达式和其它相应的默认标注属性。

LabelModifier

LabelModifier 类用于修改用于生成标注的默认属性。**LabelLayer** 为其 Sources 集合中的每个 **LabelSource** 生成标注时，首先使用 **DefaultLabelProperties** 生成每个标注。然后使用 **Modifiers** 集合中的每个可见 **LabelModifier**。

ILabelSourceFilter

该接口允许在筛选特定规则的 `LabelLayer` 中通过 `LabelSource` 对象的集合进行枚举。此外也可以实现该接口来定义自己的筛选规则。

LabelProperties

该类具有诸如样式、位置等标注属性信息。该类支持稀疏标注属性的表示。使用 `LabelModifier` 仅修改标注属性的一部分时这会非常有用。该类也允许设置标注的优先权和改善的位置。

生成标注

`LabelLayer` 类在地图绘制或调用 `LabelLayer.Refresh` 方法时生成标注。需要考虑每个可视标注源。地图中可以有多个标注图层。

要为当前地图视图中源表的每个图元生成标注，`LabelLayer` 类执行以下操作：

1. 将 `DefaultLabelProperties` 属性用作属性集的开始属性，用于生成标注。
2. 调用 `Modifiers` 集合（如果有）中每个可视标注修饰符的 `Modify` 方法。这样每个修饰符都可以更改用于生成标注的标注属性。
3. 执行可视性约束检查，以决定是否保存标注。
 - a. 通过可视性约束与当前地图缩放/比例的比较，检查标注可视性。
 - b. 如果标注可视且在该标注上不允许重叠和重复，则使用其它现有标注检查重叠和重复。如发现有任何重叠或重复，则使用标注优先权决定保存哪个标注。
4. 如果可视性约束检查成功，则将标注增加到生成标注的集合。

注意，标注生成规则仅在每个标注图层中适用，不适用于整个地图。例如，如果将所有标注源（包含在地图的标注图层中）的 `AllowOverlap` 属性都设置为 `False`，则一个标注图层中的标注仍将与另一个标注图层中的标注重叠，因为标注的生成是在两个标注图层之间独立进行的。

如果地图还未根据当前地图视图绘制生成标注，请使用 `LabelLayer.Refresh` 方法。

访问生成的标注

可以通过 `LabelSource.Labels` 属性访问生成的标注。该集合表示位于当前地图视图的标注。它们不表示地图中所有标注。集合中项目的更改与地图视图的更改同步。

标注优先权

标注优先权决定在 `AllowOverlap` 或 `AllowDuplicates` 被设置为 `False` 时同一 `LabelLayer` 中生成哪些标注。注意：正如生成标注中所述，每个标注图层都彼此相对独立，因此，其它图层中不同的重叠或重复设置不在此处显示。

标注与同一图层中另一个标注重叠，或为另一个标注的副本时，将比较两个标注的优先权以确定保存哪个标注。首先，该过程比较每个标注（或标注交互互源）的优先权。将保存具有较高优先权的标注。

MapXtreme 提供了两个用于控制标注显示的优先权等级：主优先权和次优先权。这允许将优先权分成组和子组。例如，可以让人口较多的大城市标注具有的优先权高于人口较少的小城镇。不过，也可以增加一个用于提高小城镇主优先权的修饰符，使小城镇的优先权有机会高于大城市。

Major 优先权设置为空（Visual Basic 中的 **Nothing**）时，使用的值是标注在 **LabelLayer** 中 **LabelSource** 位置的反向值。索引位置越高，优先权越低。例如，如果 **LabelSource** 是位于索引 3 且索引有 10 个 **LabelSources**（从 0 到 9 的索引），则 **Major** 优先权将为 6（基于总数的反向索引）。

Minor 优先权设置为空（Visual Basic 中的 **Nothing**）时，使用的值是基于被标注的 **Feature** 的 **Key**。使用键的数字表示的反向值，相对于 **Table** 中的行数。例如，如果 **Table** 中有 10 行，对于行 ID 为 7 的图元的标注，次优先权默认为 3 ($10 - 7 = 3$)。如果键不为数值，则次优先权默认为 0。

Major 优先权相等时，具有较高 **Minor** 优先权的标注保存。

要创建优先权，请使用生成数字值的表达式。例如，在 C# 中生成数字值的表达式，可以是数字类型或表达式的字段，例如字段值中首字母的 **ASC** 值：

```
"(1/Asc(Country))*1e6"
```

标注图层的可选择性

要控制标注图层的可选择性，请通过编程方式或通过作为可编辑复选框的 **Workspace** 管理器的标注来使用 **SelectMapToolProperties.LabelsAreEditable** 属性。

LayerHelper.SetSelectable 方法不影响特定图层类型（包括标注、WMS/WFS、光栅和组图层）的可选择性。

代码示例：创建 **LabelLayer**

以下示例说明了如何使用与 **Labels** 关联的类。

VB 示例：

```
' Open usa table using the data catalog
Dim table As Table = Session.Current.Catalog.OpenTable("usa.tab")

' Create a new map
Dim map As Map = Session.Current.MapFactory.CreateEmptyMap(New _
    Size(300, 300))

' Create a new feature layer that references the table and add it to the
map
Dim featureLayer As New FeatureLayer(table)
map.Layers.Add(featureLayer)

' Create a new label layer and add it to the map.
' Note that if you call MapInfo.Mapping.LayersBase.Add" method instead of
' MapInfo.Mapping.LayersBase.Insert method it will automatically position
the
```

```
' label layer before the feature layer

Dim labelLayer As New MapInfo.Mapping.LabelLayer()
map.Layers.Insert(0, labelLayer)

' Create a new label source that references the table
Dim source As New MapInfo.Mapping.LabelSource(table)

' Change its caption expression to be a specific column from the table
' called "State_Name"
source.DefaultLabelProperties.Caption = "State_Name"

' Append the label source to the label layer so that it shows on the map
labelLayer.Sources.Append(source)
```

曲线标注

曲线标注是沿着某条线的曲线部分的标注。曲线标注改善了由多条线组成的地图图元（例如街道和河流）的外观。

通过在 **Workspace** 管理器中选择标注图层，然后在“位置”选项卡中选中与多段弯曲，即可生成曲线标注。请参阅[曲线标注](#)。

要通过 API 渲染曲线标注，请使用 **ILayout** 界面和 **UseRelativeOrientation** 属性。有关沿几何弯曲的标注，要指定 **MapInfo.Text.RelativeOrientation**。**FollowPath**。请参阅 **ILayout** 界面下“开发人员参考”中的代码示例。

Adornments

Adornments 类是地图修饰的无序集合。修饰是 **Legend**、**Title**、**Scalebar** 或某些其它用户定义的对象。每个地图包含一个 **Adornments** 集合。每个修饰属于一个 **AdornmentControl**。

要创建自己的修饰，请从 **IAdornment** 接口和 **AdornmentControl** 抽象类派生一个类。

图例

图例用于与主题一起使用。有关主题的详细信息请参阅[第 14 章：使用主题和图例](#)。图例是使用 **LegendFactory** 类创建的。**Legend** 由一个或多个图例框架组成。每个框架不是 **ThemeLegendFrame** 就是 **CartographicLegendFrame**。通过从创建的 **Legend** 对象使用 **LegendFrameFactory** 类，可以创建两种 **LegendFrames**。对于图例及其框架的操控和定制，请使用 **MapInfo.Mapping.Legends** 命名空间中的类。名称空间中的类包括：**CartographicLegendFrame**、**ThemeLegendFrame**、**LegendFormat**、**LegendControl** 及其它。

Legend 的大小可以通过 **Legend.Size** 属性设置，但 **LegendFrames** 的大小无法设置。**LegendFrame** 的大小由它所包含的数据量控制。

VB 示例：

```
Private Sub Page_Load(ByVal sender As Object, ByVal e As _
System.EventArgs)
LegendControl1.Map = MapControl1.Map
If Not IsPostBack Then
    Dim normalLyr() As MapInfo.Mapping.LayerType = New _
MapInfo.Mapping.LayerType(1) {}
    normalLyr(0) = MapInfo.Mapping.LayerType.Normal
    Dim filter As MapInfo.Mapping.IMapLayerFilter = _
MapInfo.Mapping.MapLayerFilterFactory.FilterByLayerType(normalLyr)
    Dim frame As MapInfo.Mapping.Legends.LegendFrame
    Dim NewLegend As MapInfo.Mapping.Legends.Legend = _
MapControl1.Map.Legends.CreateLegend(New System.Drawing.Size(5, 5))
    NewLegend.Format.FrameAlignment = _
MapInfo.Mapping.Legends.LegendFrameAlignment.Horizontal
    Dim ftrLayer As MapInfo.Mapping.FeatureLayer
    For Each ftrLayer In _
MapControl1.Map.Layers.GetMapLayerEnumerator(filter)
        frame = _
MapInfo.Mapping.Legends.LegendFrameFactory.CreateCartographic_-
LegendFrame(ftrLayer)
        NewLegend.Frames.Append(frame)
    Next
    LegendControl1.Legend = NewLegend
    Else
        LegendControl1.Legend = MapControl1.Map.Legends(0)
    End If
End Sub
```

ScaleBar 修饰

比例栏是用于度量地图距离的线性对象，它使用地图单位（如千米或英尺）。使用 ScaleBarAdornment 类创建 ScaleBarAdornmentControl，并将控件增加到地图。如果试图将 ScaleBarAdornment 增加到地图本身，则显示 ScaleBarAdornment，但不具有 ScaleBarAdornmentControl，ScaleBar 不会链接到地图本身。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_ScaleBarAdornment(ByVal mapControl1 As _
MapControl)
    ' Create a scalebar
    Dim sba As ScaleBarAdornment = New ScaleBarAdornment(mapControl1.Map)

    ' Position the scalebar at the lower right corner of map
    Dim x As Integer = mapControl1.Map.Size.Width - sba.Size.Width
    Dim y As Integer = mapControl1.Map.Size.Height - sba.Size.Height
    sba.Location = New System.Drawing.Point(x, y)

    ' Add the control to the map
```

```
Dim sbac As ScaleBarAdornmentControl = New
ScaleBarAdornmentControl(sba, _
mapControl1.Map)
mapControl1.AddAdornment(sba, sbac)
End Sub
```

标题修饰

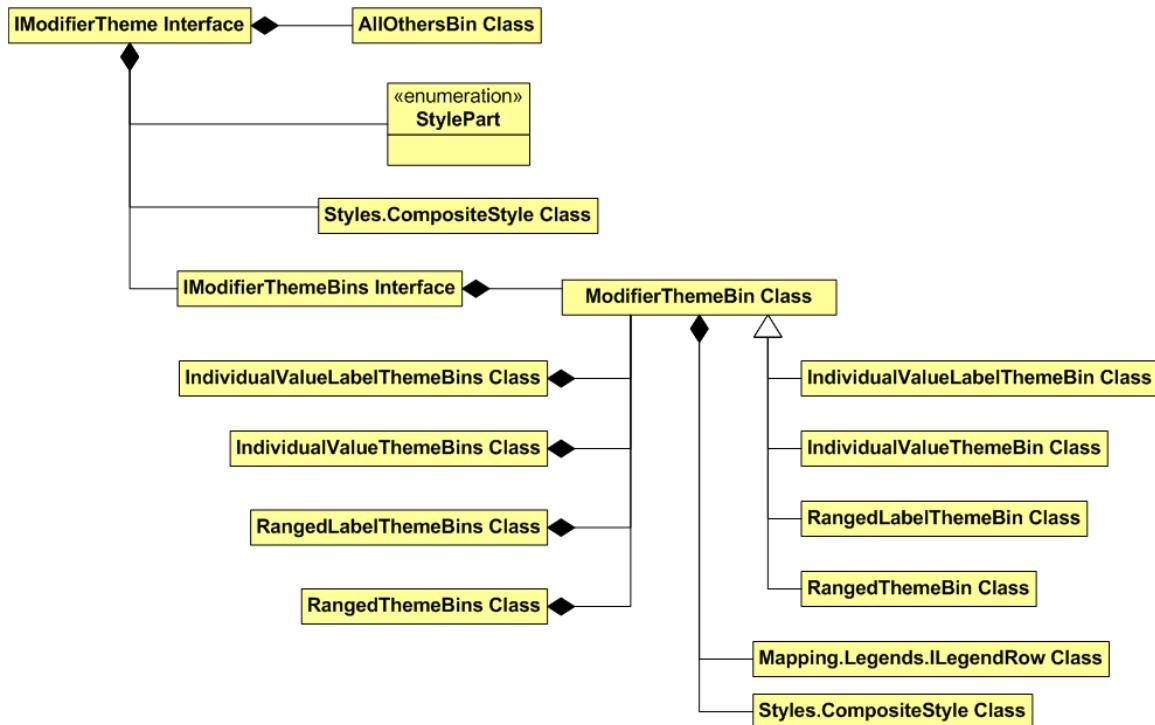
标题修饰是绘制在地图上的文本对象，表示地图标题或提供用于阐明地图上其它信息的文本。在 MapXtreme 中，使用 **TitleAdornment** 类创建标题并将其增加到地图。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_TitleAdornment(ByVal mapControl1 As
MapControl)
    ' Create a Titlebar
    Dim ta As New MapInfo.Mapping.TitleAdornment(New
System.Drawing.Size(100, 50), mapControl1.Map)
    ta.Title = "This is a watermark"
    mapControl1.Map.Adornments.Append(ta)
End Sub
```

图元样式修饰符

绘制 **Feature** 的样式之前，图元样式修饰符允许用一种方式更改或修改 **Feature** 的样式。这些类使用 **CompositeStyle** 对象的稀疏属性方面仅修改您感兴趣的样式部分。**Range**、**Individual Value** 和 **Dot Density** 主题是样式修饰符。有关主题的详细信息请参阅[第 14 章：使用主题和图例](#)。下图说明了 **Modifier** 和主题层次的 UML 表示。



FeatureStyleModifier

这是所有修饰符必须派生的抽象基类。IndividualValue、Ranged 和 DotDensity 主题全是 FeatureStyleModifier 对象。可以创建从 FeatureStyleModifier 派生的自己的类，还可以重载 Modify() 方法。

FeatureStyleModifiers

FeatureStyleModifiers 类是包含在每个 **FeatureLayer** 内的 **FeatureStyleModifier** 对象的有序集合。绘制图元的几何体之前按顺序调用集合中的每个修饰符。

FeatureOverrideStyleModifier

FeatureOverrideStyleModifier 是简单的 **FeatureStyleModifier**。该类提供了 Layer 级别上的样式重载功能。**FeatureOverrideStyleModifier** 具有组合样式且实现 **IVisibilityConstraint**。这类似于 MapX 和 MapInfo Professional 中的功能。

打印地图

应用程序创建后，您也许想要用户能够打印生成的地图。提供的 `MapInfo.Printing` 命名空间可用于简化应用程序的功能（包括打印）。该命名空间中的类根据 .NET Framework 类生成并用于打印，因此，需要的构造类似于任何其它 Windows 应用程序。除了常规打印外，我们还提供了多项附加功能，用于优化所创建地图的打印版本。

有关从 **MapXtreme** 应用程序打印的详细信息，请参阅[附录 E：从 MapXtreme 应用程序进行打印](#)。

查找位置

`MapInfo.Data.Find` 命名空间包括了在使用地址、十字路口或街道名称搜索地图图元时要使用的类。

在本章中：

- ◆ [查找功能概述](#) 246
- ◆ [Data.Find 命名空间概述](#) 248
- ◆ [微调查找过程](#) 254

查找功能概述

想通过地址、十字路口或街道名称定位地图图元时可以使用查找。要使用选择工具或查询来查找位置上的图元，请使用 **Data** 命名空间中的 **Search** 类（请参阅[搜索图元](#)）。

一次成功的 **Find** 操作可能会出现完全匹配、一个或多个近似匹配或者没有匹配（匹配失败）。该操作可由多种属性和后备选项进行控制，有关内容将在 [Data.Find 命名空间概述](#) 中进行讨论。下节描述了 **MapXtreme** 如何进行匹配。对 **Find** 过程了解的越多，对成功进行查找的操作控制就越多。

查找过程

MapXtreme 通过将地址、十字路口或位置名称与图元表中的信息进行匹配来定位地图图元。例如，如果提供关于 **Washington, D.C.**（华盛顿特区）的表，将可查找位于华盛顿特区的 **1600 Pennsylvania Avenue**。街道处于开放状态。

查找地图图元也与上述方法类似。例如，如果有一张可制图的路标表，上面包含有“白宫”和它的地理参考（可制图的）位置，就可以找到“白宫”的位置。在按位置名称进行查找时不必提供它的地址。

必须提供构成十字路口的两个街道名称才能查找该十字路口。

MapXtreme 尝试进行完全匹配，在该匹配中结果和输入的地址，位置名称或交点逐字匹配。如果无法进行完全匹配，**MapXtreme** 会尝试查找基于其匹配规则和设置的模糊匹配。如果无法查找到模糊匹配，则返回匹配失败。注意，匹配不区分大小写。大写和小写字符之间可以成功匹配。

街道地址通常由街道号码、名称以及表示街道前缀（如，**North**）和后缀（如，**Road**）的缩写组成。地址可以采用各种形式并且可以包括附加的信息，例如，公寓号码或路径。另外，输入的地址可能缺少某些重要部分，像街道类型。**MapXtreme** 会检查地址的每个部分并对其应用特定规则以查找匹配项。

下面的章节描述了 **MapXtreme** 如何处理包含街道名称、街道缩写、地址号码、优化表以及结果的特定信息和条件。

与街道名称匹配

与街道名称匹配是最直接明了的特性，它根据搜索表中的信息对地址的字符进行评估。例如，如果街道名称 **LaSalle** 位于表中，则 **MapXtreme** 为它返回完全匹配 **LaSalle**。但是，如果将该地址拼写为 **La Salle** 或 **LaSal**，则返回近似匹配。

与街道缩写匹配

地址记录中的街道缩写千差万别。有时，完全缺少街道的组成部分。然而，在很多情况下，**MapXtreme** 仍可以获得完全匹配，即使地址与搜索表稍有不同。**MapXtreme** 引用称为 **Mapinfo.abb** 的标准地址缩写和替换列表来查找适合的匹配。该列表包括了街道前缀和后缀缩写的标准拼写，如，**ST** 表示 **Street**，**Av** 表示 **Avenue**。不过必须设置属性，告诉 **MapXtreme** 使用该缩写文件，而且它还是添加获得完全匹配的机会或比不使用该文件找到更多近似匹配的好方法。

下表说明了地址中的各种变化以及根据缩写文件它是否可以获得完全匹配。第一列包含想要查找的街道名称。第二列包含来自源表的相应街道名称。第三列说明没有获得匹配的原因。第四列指示特定问题是否可以通过使用缩写等价文件进行更正。该表假设地址位于表中的单一列。而且，街道号码也位于同一列中，我们在此处没有指示街道号码，因为它们的处理方式与街道名称的处理方式不相同。

要查找的地址	搜索表地址	备注	是否可以使用缩写文件对其进行更正以获得完全匹配
LaSalle Street	LaSalle St	“Street”与“St”不匹配。	是
LaSalle Ave	LaSalle Av	“Ave”与“Av”不匹配。	是
LaSalle Ave	LaSalle St	“Ave”与“St”不匹配。	否
LaSalle	LaSalle St	目标缺少“St”。	否 如果缺少街道缩写，那么 MapXtreme 无法判断它原来的样子。
LaSalle St North	LaSalle St	搜索表中不存在“North”。	否
LaSalle St North	LaSalle St N	“North”与 N 不匹配。	是
LaSalle St Apt 3	LaSalle St	源中没有与之匹配的公寓号码。	是。忽略该公寓号码。
Tenth St	10th St	“Tenth”与“10th”不匹配。	是
10th Av	Tenth Av	“10th”与“Tenth”不匹配。	是
Saint John's Lane	St John's Lane	“Saint”与“St”不匹配。	是

如果发现已经重复多次由于缩写而不能获得匹配的情况，那么可以：

- 编辑地址使它更符合缩写文件，或者
- 使用文本编辑器编辑缩写文件以添加您的特定缩写。mapinfo.abb 位于 C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x 中。有关编辑缩写文件的详细信息，请参阅[微调查找过程](#)。

匹配地址号码

MapXtreme 在地址号码位于街道名称前面（类似“北美”地址）时或地址号码跟在街道名称后面（一般是“欧洲”地址）时可以进行匹配。默认情况下，MapXtreme 假设地址号码位于街道名称的前面。当地址号码跟在街道名称后面时，必须对属性进行设置。

MapXtreme 将比较输入地址号码与一定范围的地址号码列表。通常，该表包括了该范围覆盖的那部分相应街道的街道地址号码。地址范围可精确匹配到街道的一侧，因为地址范围往往是街道的一侧是奇数，另一侧是偶数。

在 **MapXtreme** 不能精确匹配地址范围的情况下，您可能想要在最接近的范围内获得能被视为匹配的近似匹配。仅为非常精确的查找使用严格的完全匹配要求。大多数情况下，没有必要使用这样的精确级别。近似匹配通常是可以接受的。

使用优化边界表匹配

MapXtreme 还可以在一个可能存在多个可能匹配的表中查找地址。要避免找到错误的地址，可以指定优化表和列以将匹配集中到较小的区域内。

在搜索覆盖全郡的街道表并查找 **Main St** 时，此精度很有用。该郡很可能不止一个镇有 **Main St**。通过提供精确的城镇边界表，您可以指定仅在 A 镇查找 **Main St**。查找 **Main St**。

可以按需要使用任何类型的优化边界表，例如，邮政编码边界或人口调查区域。此外，还可以指定进行 **Find** 的第二个优化边界。

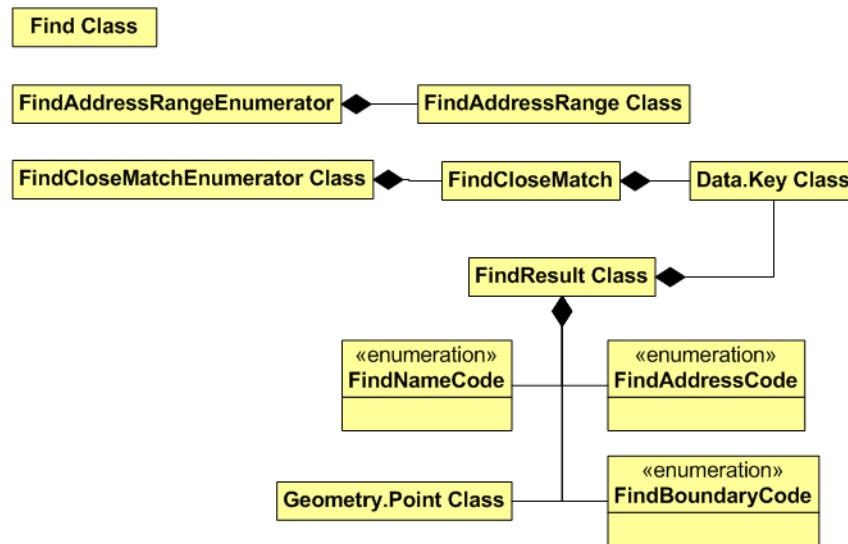
查找结果

MapXtreme 返回完全匹配、一个或多个近似匹配或者未找到匹配。查找返回的结果取决于很多因素，包括输入数据的质量和为该查找操作设置的条件。查找还将返回说明地址的每个部分如何匹配（或未找到匹配）的信息。

Data.Find 命名空间概述

MapInfo.Data.Find 命名空间包含了使您可以在可制图的表中定位地图图元、街道地址或十字路口的类。**Find** 类的属性和方法用于设置 **Find**。使用 **FindAddressRange**、**FindCloseMatch** 和 **FindResult** 类返回 **Find** 结果。

下面的 UML 图表说明了 **Find** 命名空间。



查找

Find 对象用于在给定的可制图的表中定位地图对象、街道地址或十字路口。Find 对象对表进行搜索以获取匹配并在 FindResult 对象中返回结果。

要在 MapXtreme 中使用 Find，必须具有：

- 可制图的表（即包含几何体对象的表）
- 编入索引的列，在该列执行搜索
- 要搜索的项，例如位置名称、街道地址或十字路口
- （可选）优化表，在该表中缩小搜索范围以获得更具体的位置。

Find 类提供了若干控制搜索操作的属性。例如，可以限制要返回的近似匹配的数目 (CloseMatchesMax) 或指示希望使用缩写文件来添加匹配的可能性 (UseAbbreviations)。

属性	描述
AddressNumberAfterStreet	指定地址号码是否位于街道名称的后面（例如“Smith Street 107”）。
ChooseAlternateBoundary	指定是否与优化区域中找到的记录匹配，而不是与指定的优化区域中的记录匹配。
ChooseClosestAddressRange	指定地址号码不匹配时是否使用类中最接近的可用地址号码。
ChooseClosestObject	指定找不到完全匹配时是否查找最接近的对象。
CloseMatchesMax	指定如果找不到完全匹配则要返回多少个接近的匹配。

属性	描述
InsetDistance	正值表示到直线结尾的距离，用于调节地址位置的放置。
InsetPercentage	表示在放置地址的位置直线长度的百分比。
InsetUnit	表示 Inset 所用的距离单位。
OffsetDistance	表示从街道将地址位置的放置往后偏移的距离。
OffsetUnit	表示 Offset 所用的距离单位。
UseAbbreviations	指定从缩写文件替换的缩写是否用于查找匹配（例如，将“Smith Street”替换为“Smith St”）。
UseCloseMatches	指定如果找不到完全匹配是否返回“N”个接近的匹配。
UseInsetAsPercentage	指定 Inset 被用作百分比还是距离。

Find 类提供 4 种搜索方法：其中两种方法使用 / 不使用优化边界搜索地址或图元，另外两种方法使用 / 不使用优化边界搜索十字路口。

方法	描述
Search	在可制图的表中搜索命名的位置并返回 FindResult 对象。
SearchIntersection	在可制图的街道表中搜索给定的交点，返回 FindResult 对象。
Dispose	释放 Find 对象保存的非管理资源。在使用完 Find 对象后需要调用此方法。

.FindAddressRange

FindAddressRange 对象表示从 Find.Search 方法返回的地址范围项。在未找到街道地址、地址号码不在给定街道的最小 / 最大地址范围内或未指定地址号码时，FindAddressRange 对象作为 FindResult 对象的一部分返回。

代码示例

```
public void GetAddressRangesOnStreetTable()
{
    Table_table;
    _table =
Session.Current.Catalog.OpenTable("North_Greenbush.tab");

    Find _find = new Find(_table,_table.TableInfo.Columns[1]);
    FindResult _findResult= _find.Search("Meadow Dr");
```

```
    If (((!_findResult.ExactMatch) && (_findResult.NameresultCode ==  
        FindNameCode.ExactMatch) && (findResult.AddressresultCode ==  
        FindAddressCode.AddressNumNotSpecified))  
{  
    FindAddressRangeEnumerator _enum =  
        _findResult.GetAddressRangeEnumerator();  
    FindAddressRange _findAddressRange;  
    int _iIndex = 0;  
  
    while (_enum.MoveNext())  
    {  
        _findAddressRange = _enum.Current;  
        Console.WriteLine(" _findAddressRange.BeginRange");  
        Console.WriteLine(" _findAddressRange.EndRange");  
  
        _iIndex++;  
    }  
  
    if (_table != null)  
    {  
        _table.Close();  
        _table = null;  
    }  
}  
}  
find.Dispose();  
}
```

FindCloseMatch

FindCloseMatch 对象表示从 Find Search 方法返回的近似匹配项。该对象作为 FindResult 对象的一部分返回。近似匹配项是一个返回匹配，它与请求的搜索名称近似匹配。

要使用此功能，必须先设置 Find 对象的 UseCloseMatches 和 CloseMatchesMax 属性才能执行搜索。例如，如果尝试搜索“Washington Street”并将 UseCloseMatches 设置为 true，那么得到的近似匹配为“Washington Ave”。

代码示例

```
public void CloseMatchesOnStreetTable()  
{  
    Table _table;  
    _table = Session.Current.Catalog.OpenTable("Rensselaer.tab");  
  
    Find _find = new Find(_table, _table.TableInfo.Columns[1]);  
    _find.UseCloseMatches = true;  
    _find.CloseMatchesMax = 5;  
  
    FindResult _findResult = _find.Search("70 Washington");
```

```
if (((!_findResult.ExactMatch) && (_findResult.NameresultCode ==  
    FindNameCode.ExactMatchNotFound))  
{  
    FindCloseMatchEnumerator _enum =  
        _findResult.GetCloseMatchEnumerator();  
    FindCloseMatch _findCloseMatch;  
    int _iIndex = 0;  
  
    while (_enum.MoveNext())  
    {  
        _findCloseMatch = _enum.Current;  
  
        Console.WriteLine(_findCloseMatch.Name);  
        _iIndex++;  
    }  
}  
  
if (_table != null)  
{  
    _table.Close();  
    _table = null;  
}  
find.Dispose();  
}
```

FindResult

FindResult 类以属性形式返回 Find.Search 方法中的信息，这些属性说明要进行哪种类型的匹配，如下表所示。如果成功，FoundKey 属性包含要定位的对象的 Key。如果成功，FoundPoint 属性包含已定位对象的点。

属性	描述
AddressOutOfRange	指定传入的地址是否超出范围。
AddressresultCode	标识地址部分搜索的结果代码并返回 FindAddressCode 枚举。
BoundaryresultCode	标识优化边界部分搜索的结果代码并返回 FindBoundaryCode 枚举。
ExactMatch	指定是否找到完全匹配。
FoundKey	指定已定位对象的 Key。
FoundPoint	指定已定位对象的 Point。
IntersectionNotFound	指定是否找不到交集。
MultipleMatches	指定是否找到多个匹配。
NameresultCode	标识要搜索的名称的结果代码并返回 FindNameCode 枚举。

属性	描述
resultCode	如果 Find 生成了完全匹配，则值为 1。如果 Find 生成了近似匹配，则值大于 1。如果 Find 未找到匹配的地址，则结果为负值。
useSubstitution	指定是否从缩写文件使用替换。

FindAddressCode 枚举

标识地址部分搜索的结果代码并且该代码由 `FindResult.AddressResultCode` 属性返回。

 该结果代码只应在搜索街道或十字路口时使用。

成员名称	描述
ExactMatch	找到完全匹配。
SideOfStreetUndetermined	未决定街道的哪一侧。
WithinMinMax	地址号码在最小（或）最大范围内。
NotWithinMinMax	地址号码不在最小（或）最大范围内。
AddressNumNotSpecified	地址号码未指定。
StreetsDoNotIntersect	街道不相交。
NoMapObjectForRowMatched	匹配的行不具有地图对象。

FindBoundaryCode 枚举

标识优化边界部分搜索的结果代码并且该代码由 `FindResult.BoundaryResultCode` 属性返回。改进边界用于区分具有相同名称的多个图元。

 该结果代码只应在区域用于优化搜索时使用。

成员名称	描述
ExactMatch	找到完全匹配。
FoundInOneOtherRegion	仅在指定区域之外的某一个区域中找到该名称。
FoundInMoreThanOneRegion	在除指定区域外的多个区域中找到该名称。
NoRegionSpecifiedOneMatch	未指定优化的区域，且找到一个匹配。

成员名称	描述
NoRegionSpecifiedMultipleMatches	未指定区域，且找到多个匹配。
MultipleMatchesFound	在指定区域中多次找到该名称。

FindNameCode 枚举

标识正搜索的名称的结果代码并且该代码由 FindResult.NameresultCode 属性返回。

成员名称	描述
ExactMatch	找到完全匹配。
SubstitutionUsed	来自使用的缩写文件的替换。
ExactMatchNotFound	未找到完全匹配。
NoObjectNameSpecified	未指定对象名称；找不到匹配。
CloseMatch	找到接近的匹配。

微调查找过程

在本章的开始我们就已经指出，对 **Find** 的工作原理了解的越多，就越能更好地使用它的属性和输入信息，从而提高成功匹配的机会。本节提供了一些帮助，具有用于优化 **Find** 期间返回结果的策略。

编辑 Mapinfo.abb 文件

Mapinfo.abb file 是随 MapXtreme 一起交付的缩写和替换文件。如果输入地址中的缩写可以在缩写文件中找到，则可以使用该文件来提高 **Find** 机会。有关内容包括在与街道缩写匹配一节中。本节包括了可以包含在此文件中的附加信息类型。

可以在任何文本编辑器或字处理程序中编辑 **Mapinfo.abb**。打开该文件并根据需要添加关键字。下面是该文件的标准版本：

```
!Version 3.0
FIRST      1ST
SECOND     2ND
THIRD      3RD
FOURTH    4TH
FIFTH      5TH
SIXTH      6TH
SEVENTH    7TH
EIGHTH    8TH
NINTH      9TH
```

```
TENTH      10TH
NORTH      N
SOUTH      S
EAST       E
WEST       W
ALLEY      AL
AVENUE     AV
AVE        AV
BOULEVARD BLVD
BRIDGE     BR
CIRCLE     CIR
COURT      CT
DRIVE      DR
EXTENSION  EXT
HIGHWAY    HWY
INTERSTATE I
LANE        LN
MOUNT      MT
PARK        PK
PARKWAY    PKWY
PLACE      PL
PLAZA      PLZ
POINT      PT
RAILROAD   RR
ROAD       RD
ROUTE      RT
SAINT      ST
SQUARE     SQ
STREET     ST
STR        ST
TERRACE    TER
!EOLNOSPACE

,
;
#
!EOLSPACE
FLOOR
SUITE
"P.O.BOX"
!NOSPACE
。
\!
\!
\!
!SPACE
"STATE HIGHWAY"STHWY"
"N ST" "NORTH ST"
"S ST" "SOUTH ST"
"E ST" "EAST ST"
"W ST" "WEST ST"
"N AV" "NORTH AV"
"S AV" "SOUTH AV"
```

```
"E AV" EAST AV"  
"W AV" WEST AV"
```

可以对此文件进行添加以适应多方面的问题。最重要的是可以进行几种不同类型的添加。**MapXtreme** 识别四种类型的替换项并对这些类型分别进行解释。每种类型前面都有关键字，用来在缩写文件中对其进行标识。

以空格分隔的简单替换	!SPACE
简单截断	!EOLNOSPACE
以空格分隔的截断	!EOLSPACE
简单替换	!NOSPACE

为了使 **MapXtreme** 知道如何在缩写文件中解释一行或一组行，必须将指示适当解释策略的关键字放在行首。

当缩写文件中的所有项使用默认解释时，则没有必要将任何关键字放在行首。当缩写文件的开头没有任何关键字时，**MapXtreme** 将把初始项视为需要默认解释。但是，一旦添加了其它类型的替换对，则必须添加关键字。

以空格分隔的替换

默认值是以空格分隔的简单替换。这意味着：**MapXtreme** 将目标地址中以空格分隔的标记与地址文件中的行进行比较。以空格分隔的标记是字符前后带有空格的字符串。例如，**MapXtreme** 将把 “Ave” 与 “Park Ave” 中的 “Av” 进行匹配，但是不会将 “Avery Blvd” 与 “Avry Blvd” 进行匹配（尽管两个街道名称都包含字符串 “Ave”）。字符串 “Ave” 只在 “Park Ave” 中由空格定界，而未在 “Avery Blvd.” 中以空格定界。在 “Avery Blvd” 中，“Ave” 后跟 “r” 而不是空格。

缩写文件中的所有项都将接收这个默认解释。可以添加要接收同样处理的其它项。例如，可能想要添加 “WK WALK” 对，以使 **MapXtreme** 知道将目标地址中的 “WK” 当作 “WALK” 进行解释。同样地，可能想要添加类似如下的配对：“WAY WY”。

使用关键字 “!SPACE” 来指示以空格分隔的简单替换。将对跟在 “!SPACE” 后面的项进行默认解释（这使可以按其它顺序安排缩写文件的内容）。在 **MapXtreme** 遇到另一个关键字时，它将切换到指示的解释策略。

简单截断

在简单截断中，**MapXtreme** 查找地址中的项并将它以及后跟的内容完全忽略。这些项不一定要用空格分隔。这个策略对于处理类似如下的地址非常有用：

123 Appian Way, Mail Stop 829

7305 Van Zandt # 23

在第一种情况中，想要 MapXtreme 忽略逗号以及它后跟的内容。在第二种情况中，想要 MapXtreme 忽略 # 号以及后跟的内容。要处理这样的情况，请将下面的行添加到缩写文件中：

```
!EOLNOSPACE  
,  
#
```

“!EOLNOSPACE” 是指示下面的项将被视为简单截断的关键字。之后，我们得到了一个具有逗号的行和一个具有 # 号的行。不论 MapXtreme 何时在地址中遇到逗号或 # 号，都将忽略它以及后跟的内容。上面的示例变为：

123 Appian Way

7305 Van Zandt

以空格分隔的截断

在以空格分隔的截断中，MapXtreme 查找那些以空格分隔的项并将它们以及后跟的内容删除。例如：

73 Appian Way Suite 829

3033 Van Zandt Room 202

要处理这样的情况，请将下面的行添加到缩写文件中：

```
!EOLSPACE  
SUITE  
ROOM"
```

“!EOLSPACE” 是指示下面的项将被视为简单截断的关键字。之后，我们得到了一个具有 “Suite” 的行和一个具有 “ROOM” 的行。无论 MapXtreme 何时遇到这些标记，它都将截断地址。上面的示例变为：

73 Appian Way

3033 Van Zandt

简单替换

MapXtreme 使用简单替换从地址中删除项，否则不执行任何操作。使用它来处理：

433 Van-Rensselaer

91 St Albans'

目标是删除连字符和单引号。将下面的项添加到缩写文件中：

```
!NOSPACE  
-  
,
```

“NOSPACE” 是调用简单替换的的关键字，并且下面行中的连字符和单引号都是要删除的标记。上面的示例变为：

369 VanRensselaer

91 St Albans

合法的空格

可能会有想为含有空格的搜索字符串指示替换的情况。可以在这些情况中使用双引号。将双引号放在：

- 行首；或者
- 搜索的字符串和替换之间；或者
- 行尾。

例如，可能想要使用“STHWY”替换“State Highway”。要执行此操作，请使用下面一行：

```
"State Highway"STHWY"
```

这对细微的问题提供了解决方案，街道名称与缩写文件中的项匹配。例如，“North St”和“Park Av”都具有与缩写文件中的项相匹配的初始字符串。因此，MapXtreme 将使用“N”替换“North”以得出“N St”并使用“Pk”替换“Park”以得出“Pk Av”。可以将下面的行添加到“缩写”文件中以调整这些替换：

```
"N ST"North ST"  
"PK AV"PARK AV"
```

注意：这些行必须位于使用“N”替换“North”和使用“PK”替换“Park”的项的后面。如果这些行位于上述项的前面，则不起作用。因此：

```
...  
...  
NORTH N  
...  
...  
PARK PK  
...  
...  
"N ST"NORTH ST"  
"PK AV"PARK AV"  
...  
...
```

MapXtreme 遇到 NORTH N 时，会将 NORTH ST 改为 N ST。当它遇到“N ST”NORTH ST”时，会将 N ST 改为 NORTH ST。PARK AV 的处理方式与此相同。

特殊字符

MapXtreme 使用惊叹号 (!)、双引号 ("") 和反斜杠 (\) 作为特殊字符。这些字符告诉 MapXtreme 如何处理跟在它们后面的字符串，但通常它们自己不被视为替换字符串中的字符。惊叹号告诉 MapXtreme 不应将字符串解释为缩写。双引号告诉 MapXtreme 字符串中的空格是合法的。而反斜杠则告诉 MapXtreme 将特殊字符视为普通字符。

当想要在一行中将任何字符用作简单字符时，就在它们的前面加上反斜杠。因此：

```
\!  
\!  
\!
```

在缩写文件中增加行

通过添加新行，可以将新的项添加到该文件中。除了考虑到对其它行的影响，希望使用一个替换对以外，添加新行的顺序不是很重要。一行中第一个项和第二个项之间的空格数也不重要。

错误的地址范围

地址包含源表以外的数字范围时，**MapXtreme** 无法与之匹配。这样的地址可能落入范围值之间的间隙或超出范围界限。要处理该问题：

1. 将 `Find.ChooseClosestAddressRange` 属性设置为 `true`，然后执行搜索。
2. 通过查看 `FindAddressCode` 枚举（由 `FindResult.AddressOutOfRange` 属性返回）解析失败的匹配。

该地址可能用于在生成源表后增加的街道分段。在这种情况下，要对源表进行编辑，使其反映该街道地址的整个范围。

不准确的城镇名

查找操作**MapXtreme**的最后一步是确定要将匹配的街道地址放在哪个区域。只有在设置查找过程时有此要求，**MapXtreme** 才会执行此步骤。通常将城镇或城市名作为改进边界。由于人们通常不使用 Census Bureau 分配到各地址的城镇名，因此就产生了一些问题。由于几乎美国所有的电子地图都基于 Census Bureau 地图，因此就产生一些问题。

例如，地址 “50 Wolf Rd., Albany, NY” 实际位于 Colonie 的小镇内。因此，在目标地址中该城镇名对应的地址无法与源文件中正确的城镇相匹配。

解决该问题的一个办法是使用 `ChooseAlternateBoundary` 属性。启用该选项时，**MapXtreme** 会尝试将地址与其所在边界相匹配（假定该地址只位于一个边界内）。地址位于多个边界之内时，`Find` 操作失败。

处理该问题的另一个方法是将 ZIP Code 而不是城镇或城市名作为改进边界。

使用主题和图例

MapXtreme 提供了多个用于将主题和图例添加到地图的选项。以下章节说明了可用的不同类型主题和图例并解释了如何使用它们。

在本章中：

◆ 主题概述	261
◆ GraduatedSymbolTheme	262
◆ PieTheme	263
◆ BarTheme	264
◆ RangedTheme	265
◆ RangedLabelTheme	267
◆ 范围主题和序列化	268
◆ IndividualValueTheme	268
◆ 使用自定义位图符号创建 IndividualValueTheme	269
◆ IndividualValueLabelTheme	270
◆ IndividualValue 主题和序列化	271
◆ DotDensityTheme	271
◆ 图例概述	273

主题概述

主题地图绘制用于显示数据的趋势，而这种趋势在表格数据中难于表现。主题通常使用某块或某几块数据。可以使用来自数据源（例如，本地 MapInfo 表）的数据将地图以不同的主题加上底纹。例如，可以根据每个州的平均温度为“美国”地图加上底纹。在看见红色时，就可以知道此地天气炎热（高温），但是在看见蓝色时，就知道此地天气寒冷（低温）。

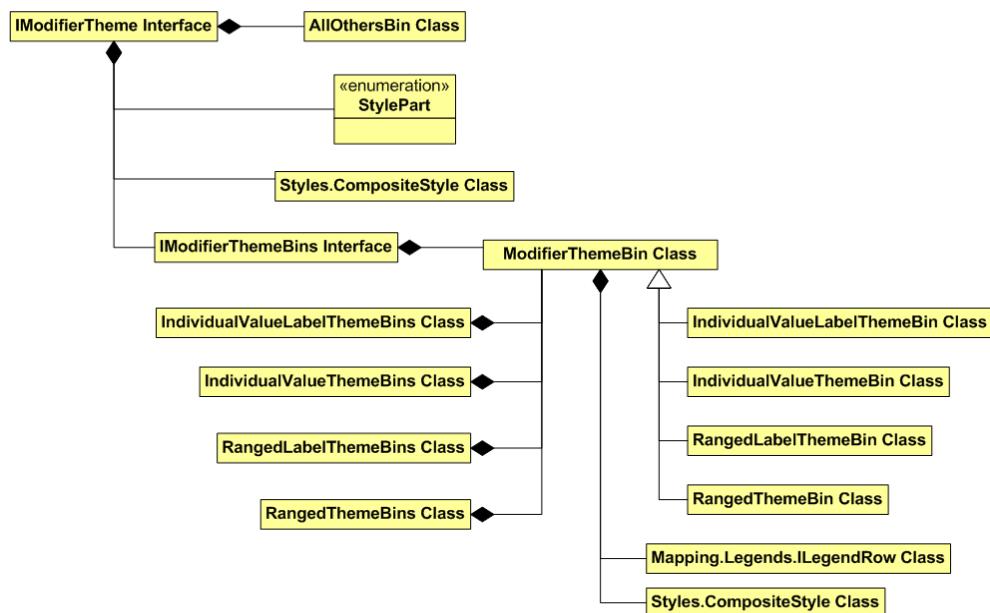
主题使用颜色、填充图案或符号底纹来表示数据。主题地图显示数据有多种用法。可以通过将这些颜色、图案或符号指定给地图对象（根据数据中的特定值）来创建不同的主题地图。

Mapping.Thematics 命名空间

MapInfo.Mapping.Thematics 命名空间包含的类以 Feature 图层上的样式重载和 Object 主题实现来主题。对象主题添加新的图层时修饰符主题更改样式。所有主题都实现 ITheme 接口。

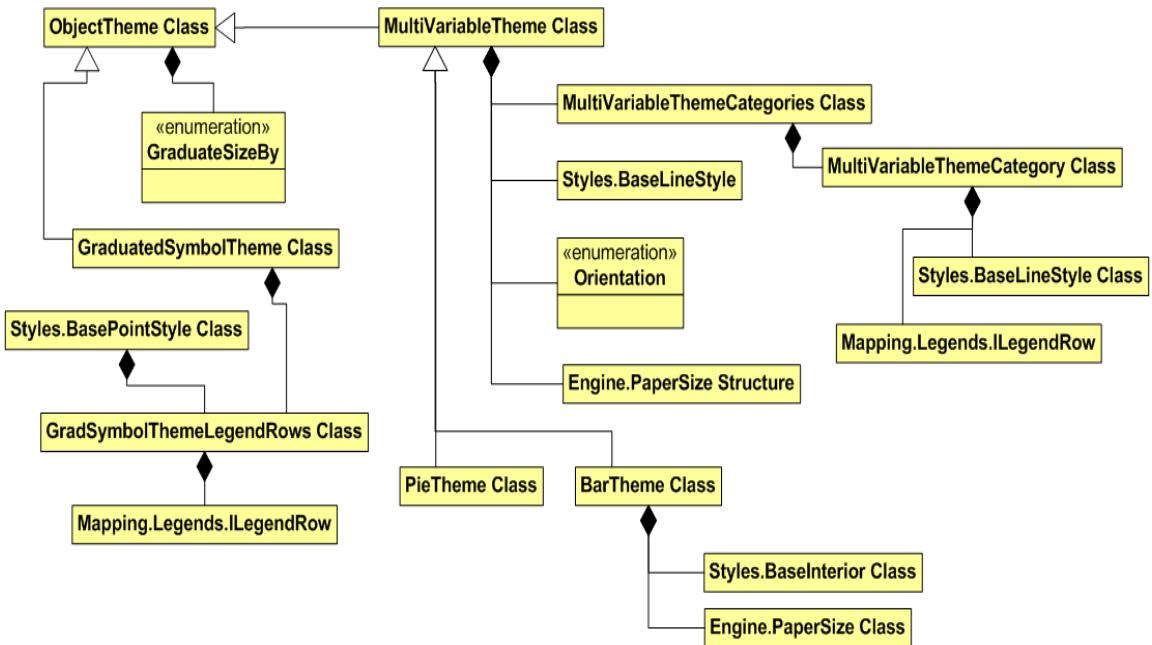
修饰符主题

图元修饰符主题的示例是范围、单值和点密度主题地图。它们修改图层中的现有图元。下面的 UML 图表概述了修饰符主题层次。



对象主题

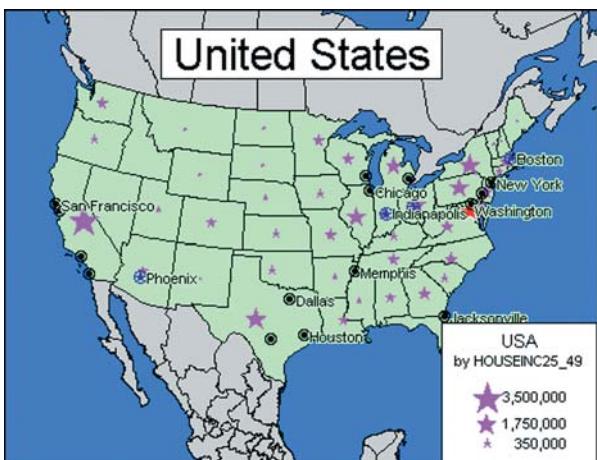
对象主题包括分级符号、饼图和条形图。这些主题创建表示数据值的对象。下面的 UML 图概述了对象主题层次。



GraduatedSymbolTheme

分级符号主题是包含点图元的对象主题，其符号大小基于主题表达式的数字值。

例如，用户可以用分级符号来显示某一地区人口特定阶段的家庭收入。



分级符号主题地图的示例。

何时使用分级符号主题

分级符号地图仅使用数值数据。没必要根据每个餐厅所提供的烹调类型来创建分级符号。但是，如果想要按城市显示家庭收入的分布情况，则适合使用分级符号。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Thematics_GraduatedSymbolTheme(ByVal map As Map)

    ' Load a map based on one table
    map.Load(New MapTableLoader("world.tab"))
    Dim lyr As FeatureLayer = CType(map.Layers("world"), FeatureLayer)

    ' Create a new graduated symbol theme
    Dim gradTheme As GraduatedSymbolTheme = New _
        GraduatedSymbolTheme(lyr.Table, "Pop_Native")

    ' Create an object theme layer based on that graduated symbol theme
    Dim thmLayer As ObjectThemeLayer = New ObjectThemeLayer("World Pop _Growth Rate", Nothing, gradTheme)

    'Add object theme to the map's layer collection.
    map.Layers.Add(thmLayer)

    ' Adjust how we graduate the size.
    gradTheme.GraduateSizeBy = GraduateSizeBy.Constant
    thmLayer.RebuildTheme()
End Sub
```

PieTheme

饼状主题是包含饼状图的对象主题，其中饼块表示每个数据值。在饼图中可以比较单个饼图中的饼块，或检查所有饼图的特定饼块。饼图还能够比较整体中的各个部分。

何时使用饼图主题

饼图在分析人口统计学数据时特别有用。例如，对于“美国”人口统计学信息的数据集，数据集显示多个主要人口统计组的人口。通过饼图，可以显示每个人口统计组的人口，并查看组成饼图的每个饼块。这样就可以查看人口统计组在每一个州或整个“美国”的分布情况。还可以通过查看某一个人口统计组，了解不同州内人口的相对差异。

VB 示例：

```
Public Shared Sub MapInfoMappingThematicsPieTheme(ByVal map As Map)
    ' Load a map based on one table
    map.Load(New MapTableLoader("mexico.tab"))
    Dim lyr As FeatureLayer = CType(map.Layers("mexico"), FeatureLayer)
```

```
' Create a new pie theme
Dim pieTheme As MapInfo.Mapping.Thematics.PieTheme = New _
    MapInfo.Mapping.Thematics.PieTheme(map, lyr.Table, "Cars_91", _
    "Buses_91", "Trucks_91")

' Create an object theme layer based on that pie theme
Dim thmLayer As ObjectThemeLayer = New ObjectThemeLayer("Count by _"
    "Vehicle Type", Nothing, pieTheme)

'Add object theme to the map's layer collection.
map.Layers.Add(thmLayer)

' DataValueAtSize is calculated automatically if not specified in the
' pie's constructor. But, you can adjust it. If you do so here, you
' have to rebuild the theme. You can adjust it before creating the
' object theme layer, and that way the pies won't need to be built
' twice.

pieTheme.DataValueAtSize /= 2
pieTheme.GraduateSizeBy = GraduateSizeBy.Constant
thmLayer.RebuildTheme()
End Sub
```

打印包括饼形 / 条形主题的地图

为打印包括饼形或条形主题图而复制地图时，请务必考虑页面尺寸以获得预期的结果。页面尺寸用于计算打印输出中饼形/条形的尺寸。使用 100% 的页面尺寸，以便在打印预览中图形/主题能够显示与其在 MapControl 上相同类似的尺寸。

BarTheme

条形主题是包含条形图的对象主题，其中条形表示每个数据值。条形图为了对象质心的每个地图对象（图元）生成，使您能够通过对条形高度的比较来分析特定图表中的多个主题变量。

何时使用条形图主题

条形主题可用于在地图的所有图元中检查同一变量。例如，有一个关于美国各州边界 的表，表中包含男女人口信息。使用条形图，您可以创建为每个州显示两个条形图的主题图：一个条形图表示女性人口，另一个则表示男性人口。可以比较每个州的人口差异，或者检查多个州并比较与其它州的人口差异。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Themes_BarTheme(ByVal map As Map)
    ' Load a map based on one table.
```

```
map.Load(New MapTableLoader( "world.tab" ))
Dim lyr As FeatureLayer = CType(map.Layers( "world" ), FeatureLayer)

    ' Create a new bar theme.
Dim barTheme As MapInfo.Mapping.Thematics.BarTheme = New _
MapInfo.Mapping.Thematics.BarTheme(map, lyr.Table, "Pop_Native", _
"Pop_Asian", "Pop_Other")

    ' Create an object theme layer based on that bar theme.
Dim thmLayer As ObjectThemeLayer = New ObjectThemeLayer( "World _ _
Pop" , Nothing, barTheme)

    ' Add object theme to the map' s layer collection.
map.Layers.Add(thmLayer)

    ' Stack the bars and graduate by a constant amount.
barTheme.Stacked = True
barTheme.GraduateSizeBy = GraduateSizeBy.Constant
thmLayer.RebuildTheme()

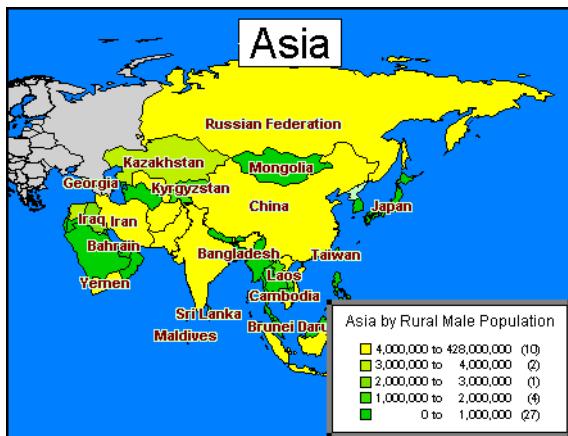
End Sub
```

控制饼状主题图和条形主题图的显示尺寸

MapXtreme 提供支持创建饼状和条形对象主题图的抽象基类 **MultiVariableTheme**。由 **ObjectTheme** 类派生的类提供两个属性：**DataValueAtSize** 和 **Size**，这些属性可控制对象主题几何体在特定值下的大小。**DataValueAtSize** 的默认值设为绘制地图的图元的最大数据值。“尺寸”值可控制饼图的宽度和条形图的高度（以图纸单位计算）。

RangedTheme

范围主题会根据特定标准将已归类数据显示为范围（容器）。在 **MapXtreme** 中，范围主题会修改现有的图层，以反映该项标准。不会创建新图层，原因是范围主题已在 **MapX** 和 **MapXtreme** 的先前版本中创建。在创建范围主题地图时，**MapXtreme** 将所有数据集行集中在范围内，并为每行分配对象的颜色、样式或其对应范围的直线。



范围主题地图示例。

何时使用范围主题

例如，在有某区域人口统计数据时，可以使用范围主题。例如，可将亚洲的农村男性人口集中在容器内，并加上底色，以便指示存在于该区域内的人口范围。

数据集中所有记录都被分配到某一范围，然后根据该范围使用样式进行绘制。例如，如果颜色的范围是从黄色到绿色，则可将人口最高国家的底色设置为黄色，人口最低的则为绿色，其它范围选择由黄到绿之间任意颜色。地图显示时，这些颜色会明确指示出人口最高和最低国家的位置。（有关如何将半透明效果应用到范围主题图地图的指示，请参阅[如何对主题图应用半透明效果](#)。）

当区域的大小与数据值的大小不直接相关时，也可以使用范围。

范围值的类型

MapXtreme 可以使用五种分布方法自动创建范围：

- 数量相等
- 范围相等
- 标准偏差
- 自然间隔
- 分位数
- 定制

数量相等

“数量相等”在每个范围内具有相同数目的记录。如果想使用“数量相等”将 100 条记录分组到 4 个范围内，MapXtreme 对这些范围进行计算，根据设置的舍入因子，为每个范围近似填充 25 条记录。

使用“数量相等”（或任何其它范围方法）时，注意任何可能影响主题地图的极大数据值（在统计信息中，这些值作为外部值引用）非常重要。

范围相等

“范围相等”按相等大小的区域划分记录。例如，表中有个字段的数据值范围为 1 到 100。您希望创建包含四个范围相等的主题图。那么，MapX 生成的范围为 1 - 25、26 - 50、51 - 75 和 76 - 100。切记：MapXtreme 可能会创建不包含任何数据记录的范围，这取决于数据的分布状况。

标准偏差

使用“标准方差”创建范围时，中间范围按平均值进行间隔，并且中间范围上面和下面的范围按该平均值上面或下面的标准方差进行间隔。

自然间隔

自然间隔是显示非均匀分布数据的一种方式。它根据这样一种算法来创建范围：使用每个范围的平均值在所有范围中平均分布数据。使用“自然间隔”分布数据，可使每个范围的平均值尽可能接近该范围内的每个范围值。这可以确保各个范围可以很好地被它们的平均值表示，并且每个范围内的数据值彼此都非常接近。

分位数

分位数是显示非均匀分布数据的另外一种方式。分位数使用两个变量表达式。例如，使用 Quantile 分布方法显示人口识字率。

定制范围

如果以上分布方法都不能满足需要，则可以使用方法 `DistributionMethod.CustomRanges` 创建定制范围。请参阅 MapXtreme 开发人员参考帮助中 `MapInfo.Thematics.RangedTheme.Recompute` 方法的代码示例。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Thematics_RangedTheme(ByVal map As Map)
    ' Create a ranged theme.
    Dim lyr As FeatureLayer = CType(map.Layers(0), FeatureLayer)
    Dim theme As MapInfo.Mapping.Thematics.RangedTheme = New _
        MapInfo.Mapping.Thematics.RangedTheme(lyr, "Pop_1990/Area(obj,_
        'sq mi')", "PopDensity", _
        5, DistributionMethod.EqualCountPerRange)

    ' Add the ranged theme to the layer.
    lyr.Modifiers.Append(theme)
End Sub
```

RangedLabelTheme

该类创建范围主题，在其中标注使用范围样式绘制。有关范围主题更加详细的讨论，请参阅 [RangedTheme](#)。

何时使用 RangedLabelTheme 类

用标注来传递要标注的信息时，可使用范围标注主题。例如，可在标注城市或城镇人口时使用有范围的标注主题。人口较多的城市的标注字体会比人口较少的城镇大。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Thematics_RangedLabelTheme (ByVal
labelSource As MapInfo.Mapping.LabelSource, ByVal columnExpr As _
String, ByVal themealias As String)
    ' Create new ranged label theme based on the label source of a
    ' LabelLayer already in the map. It will use 5 bins of equal range.
    Dim rangedLabelTheme As RangedLabelTheme = New _
        RangedLabelTheme(labelSource.Table, columnExpr, themealias, 5, _
        DistributionMethod.EqualCountPerRange)

    ' Add the label modifier to the label layer.
    Dim labelModifier As MapInfo.Mapping.LabelModifier = _
        CType(rangedLabelTheme, MapInfo.Mapping.LabelModifier)
    labelSource.Modifiers.Insert(0, labelModifier)

End Sub
```

范围主题和序列化

从 7.0.0 版开始，为提高性能已更改了 RangedThemes 和 RangeLabelThemes 的序列化和反序列化。

更改发生在主题序列化中，此时主题的容器或类别记录计数都已被序列化。当反序列化时，不再调用记录计数，而是将序列化的记录计数应用于主题图。实施此更改可提高序列化/反序列化的性能。

重要的是，要注意那些想要根据反序列化更新主题记录计数的应用程序，必须在反序列化后添加逻辑以更新主题容器或类别记录计数。有关 RangeTheme 和 RangeLabelTheme，可调用 MapInfo.Mapping.Thematic.IRangedTheme.Recompute 方法。

IndividualValueTheme

单值主题是修饰符主题，用来显示点、直线或由包含在数据集特定字段中的单值加上底纹的边界。可在独立值地图中使用数字值和名义值。MapXtreme 为每个唯一值提供独特的样式。

例如，使用单值主题图显示各块土地的分区类别。每个区域（商用、住宅、工业）将显示为不同的颜色。每块符合分区类别的土地都将加上相应底色。

何时使用 **IndividualValueTheme** 类

如果通过名义数据在点、直线或边界加上底纹，则只能通过单值执行。名义数据可以是非数字数据（例如，名称、提供的烹饪类型或销售的汽车品牌）或其数字不表示测量的数字数据。例如，名义数据可以是包含 ID 数值的列。

数据被视为数值数据，可用于范围地图和单值地图。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Thematics_IndividualValueTheme(ByVal_
    map As Map)

    ' Load a map based on one table
    map.Load(New MapTableLoader( "World.tab" ))
    Dim fLyr As FeatureLayer = CType(map.Layers( "world" ), FeatureLayer)

    ' Create an individual value theme
    Dim thm As IndividualValueTheme = New _
        IndividualValueTheme(fLyr, "Country", "World Pop")

    ' Add the theme to the FeatureStyleModifiers list
    fLyr.Modifiers.Append(thm)
End Sub
```

使用自定义位图符号创建 **IndividualValueTheme**

以下 C# 示例说明了如何使用自定义位图符号创建 **IndividualValueTheme**。可用位图符号表包括在[定制符号](#)中。

```
// Open a connection to the Catalog
MapInfo.Data.MIConnection conn = new MapInfo.Data.MIConnection();
conn.Open();

// Retrieve a table from the Catalog
MapInfo.Data.Table ti=conn.Catalog.GetTable("usa_caps");
// Add it as a layer to MapControl
MapInfo.Mapping.FeatureLayer fl=mapControl1.Map.Layers["usa_caps"] _ 
    as FeatureLayer ;

// Create a new IndividualValueTheme
MapInfo.Mapping.Thematics.IndividualValueTheme iv=new _ 
    MapInfo.Mapping.Thematics.IndividualValueTheme(fl,"state","state");

// Add a custom bitmap symbol
MapInfo.Styles.BitmapPointStyle bitmappointstyle = new _ 
    MapInfo.Styles.BitmapPointStyle("AMBU1-32.BMP", _ 
        MapInfo.Styles.BitmapStyles.All ,System.Drawing.Color.Red , 30);
// Set the style
```

```
MapInfo.Styles.CompositeStyle cs = new _  
    MapInfo.Styles.CompositeStyle(null, null, null, bitmappointstyle);  
  
    // Apply the style to the first bin  
    MapInfo.Mapping.Thematics.ModifierThemeBin mtb= iv.Bins[0];  
    mtb.Style.ApplyStyle(cs);  
  
    // Add another bitmap symbol  
    bitmappointstyle = new _MapInfo.Styles.BitmapPointStyle("BADG1-32.BMP", _  
        MapInfo.Styles.BitmapStyles.All ,System.Drawing.Color.Red , 30);  
    // Set the style  
    cs=new MapInfo.Styles.CompositeStyle(null, null, null, _  
        bitmappointstyle);  
  
    // Apply the symbol to the second bin  
    mtb= iv.Bins[1];  
    mtb.Style.ApplyStyle(cs);  
  
    // Append the style modifiers to the feature layer  
    f1.Modifiers.Append (iv);  
  
    //Close the connection  
    conn.Close();
```

IndividualValueLabelTheme

该类创建在图层的标注上进行操作的单值主题。有关单值主题更加详细的讨论，请参阅 [IndividualValueTheme](#)。

何时使用 IndividualValueLabelTheme 类

因为单值标注主题带有范围的标注主题，那么在使用这些标注来传递要标注的信息时，它们也非常有用。例如，利用街道数据时，可以使用单值标注主题用不同字体来标注不同类型的道路。在这种情况下，高速公路将使用与乡间小道不同类型的标注进行表示。

VB 示例：

```
Public Shared Sub  
MapInfo_Mapping_Thematics_IndividualValueLabelTheme(ByVal labelSource As  
MapInfo.Mapping.LabelSource, ByVal columnExpr As String, ByVal _  
themeAlias As String)  
    ' Create new individual value label theme  
    Dim theme As IndividualValueLabelTheme = New _  
    IndividualValueLabelTheme(labelSource.Table, columnExpr, themeAlias)  
  
    ' Add the label modifier to the label layer.
```

```
Dim labelModifier As MapInfo.Mapping.LabelModifier = CType(theme, _
    MapInfo.Mapping.LabelModifier)
labelSource.Modifiers.Insert(0, labelModifier)
End Sub
```

IndividualValue 主题和序列化

从 7.0.0 版开始，为提高性能已更改了 **IndividualValueThemes** 和 **IndividualValueLabelThemes** 的序列化和反序列化。

更改发生在主题序列化中，此时主题的容器或类别记录计数都已被序列化。当反序列化时，不再调用记录计数，而是将序列化的记录计数应用于主题图。实施此更改可提高序列化/反序列化的性能。

重要的是，要注意那些想要根据反序列化更新主题记录计数的应用程序，必须在反序列化后添加逻辑以更新主题容器或类别记录计数。有关 **IndividualValueTheme** 和 **IndividualValueLabelTheme**，可调用 **MapInfo.Mapping.Thematics.ILModifierTheme.RecomputeBins** 方法。

DotDensityTheme

点密度主题是绘制区域填充图案的样式修饰符，它所使用的点基于主题表达式的数字值。

点密度地图用点来表示与边界或区域关联的数据值。区域中点的总数表示该区域的数据值。如果一个郡有 10,000 个老年人，每个点表示 100 个老年人，则在郡边界中应该有 100 个点。

何时使用 DotDensityTheme 类

点密度主题图可用于显示原始数据，图中一个点代表许多信息：人口、快餐数目、某汽水品牌的批发商，等等。

例如，对于郡边界的人口明细表，则可以使用点密度主题来显示每个郡边界上的人口密度。可通过两个属性控制点密度地图。可以指定一个点的值。例如，要通过点密度主题表示纽约 Rensselaer County 中 20,000 个高校生，可以指定一个点表示 200 个学生。在该郡加上底纹时，地图会为该郡绘制 100 个点。

VB 示例：

```
Public Shared Sub MapInfo_Mapping_Thematics_DotDensityTheme(ByVal map As
Map)

    ' Load a map based on one table
    map.Load(New MapTableLoader("mexico.tab"))

    ' Create a dot density theme.
    ' Add it as a modifier.
    Dim lyr As FeatureLayer = map.Layers("mexico")
```

```
Dim thm As MapInfo.Mapping.Thematics.DotDensityTheme = New  
MapInfo.Mapping.Thematics.DotDensityTheme(lyr, "Pop_90", "mexico Pop",  
System.Drawing.Color.Red, DotDensitySize.Large)  
  
' thm.DotColor is System.Drawing.Color.Red  
' thm.DotSize is DotDensitySize.Large  
  
' Set each dot to represent 20,000 people  
thm.ValuePerDot = 20000  
  
End Sub
```

双变量主题地图

双变量主题地图绘制使用点或直线对象来表示两个主题变量。例如，星型标记可以表示一个变量，如青少年的数目，而使用蓝色填充的星型标记表示他们的年度购买量。

要在 **MapX** 中创建双变量地图，请创建两个主题地图，并将其中一个地图放在另一个地图上面，从而对对象显示两个变量。

地图和变量的类型

适用于双变量地图绘制的主题地图类型只有范围地图和单值地图。可以根据数据，在双变量地图的组合之间进行选择：

- 两个范围地图
- 一个范围地图和一个单值地图

如果具有非数字变量，则其中一个地图必须是单值地图。使用两个非数字变量不能创建双变量地图。

显示属性

如果要使用一种符号显示两种变量，那么为每个变量选择不同的符号属性就很重要。例如，不能为两个变量选择同一种颜色，因为其中一种颜色将覆盖另一种颜色。从下列组合进行选择：

- 颜色和符号类型
- 颜色和大小
- 颜色和符号类型

符号类型只应用于名义或非数值数据，因为符号类型和数量之间没有内在关联。

VB 示例：

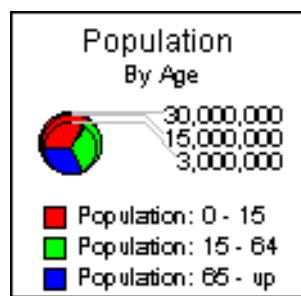
```
Public Shared Sub MapInfo_Mapping_Thematics_RangedThemeConstructor(ByVal  
lyr As FeatureLayer)  
    Dim thm As MapInfo.Mapping.Thematics.RangedTheme = New _  
        MapInfo.Mapping.Thematics.RangedTheme(lyr, "Literacy", "Pop_1994")  
    ' -  
    ' Literacy Quantile by Pop" , 4)  
    lyr.Modifiers.Append(thm)  
End Sub
```

图例概述

`MapInfo.Mapping.Legends` 命名空间包含用于创建并显示主题和制图图例的类、接口和枚举。`Legends` 是 Thematic 或 Cartographic LegendFrames 的集合。每个框架都包含 `LegendRow` 的集合，每个 `LegendRow` 都具有文本和样式属性。

主题图例

主题图例提供用于主题的颜色、符号和样式的键。该键解释显示的颜色、符号和样式。



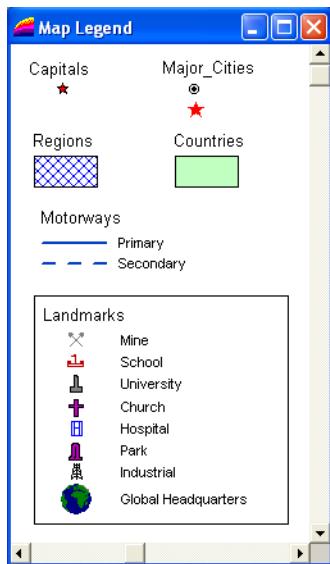
何时使用主题图例

具有包含主题的地图时，主题图例就非常有用。在显示降雨量的天气地图中，可用各种绿色底纹表示降雨。主题图例的重要性在于它可从视觉进行阐述，深绿色底纹表示最高降雨量，浅绿色底纹则表示较低降雨量。

在 `\Samples\Desktop\Features\ThemeLegend` 文件夹中，提供的样本应用程序演示如何在表上创建范围主题、创建范围主题的图例和图例框以及将其作为修饰添加到地图。

制图图例

制图图例类用于读写制图图例元数据。图例通过元数据中的文本和样式标识地图上的每个制图图元。



何时使用制图图例

当地图包含表示地图上项目的对象时，制图图例非常有用。例如，具有路标的地图就需要制图图例。医院、学校、教堂和机场都由不同的符号表示。制图图例为表示在地图上的不同类型的路标提供可视性说明。

有关演示如何创建制图图例的代码示例，请参阅《开发人员参考》中的 `MapInfo.Mapping.Legends.Legend` 类。

将图例格式化

`MapInfo.Mapping.Legends.LegendFormat` 类包含控制图例框架绘制方式的属性。可以控制显示属性，例如对齐、每行或每列中图例框架的数量、两个框架之间的间隔，以及是否应自动调整图例大小和框架位置。

`LegendFormat.FrameAlignment` 属性与 `FramesPerRow` 一起使用可用于水平对齐，`FramesPerColumn` 则用于垂直对齐。

例如，如果图例包含 10 个框架并将 `FrameAlignment` 和 `FramesPerRow` 分别设置为 `Horizontal` 和 5，则图例将框架显示为两行，每行显示五个图例。如果每行有 10 个框架，则 10 个框架显示在单一行中，宽度则为 10 个框架的宽度。

类似情况发生在垂直对齐时。如果有 10 个框架并将 `FrameAlignment` 和 `FramesPerColumn` 分别设置为 `Vertical` 和 5，则将显示 5 行，每行 2 个框架（5 行 2 列）。框架垂直向上对齐为每列 5 个框架。将 `FramesPerColumn` 设置为 10 时，图例将包含 10 行，每行 1 个框架（每列 10 个框架）。

`FramesPerRow` 和 `FramesPerColumn` 的默认设置均为 0。使用的值是由 `LegendFrameRows.Count` 或 `LegendFrameColumns.Count` 属性指示的行或列中的当前框架数目。

为地图设置样式

MapXtreme 中的样式不仅影响地图图元的外观，而且还影响了地图绘制应用程序的许多组件。样式可用于具有很多可控属性的标注、文本、主题、图例、选择集和演示文稿，所以实际上可以设计出喜欢的任何样式。

本章讨论了 **MapXtreme** 框架的样式，特别讨论了 **MapInfo.Styles** 命名空间。

在本章中：

- ◆ **MapInfo.Styles** 命名空间概述 276
- ◆ 样式说明 277
- ◆ 预定义的样式和 **StyleRepository** 类 280
- ◆ 使用样式 281
- ◆ 覆盖样式 282

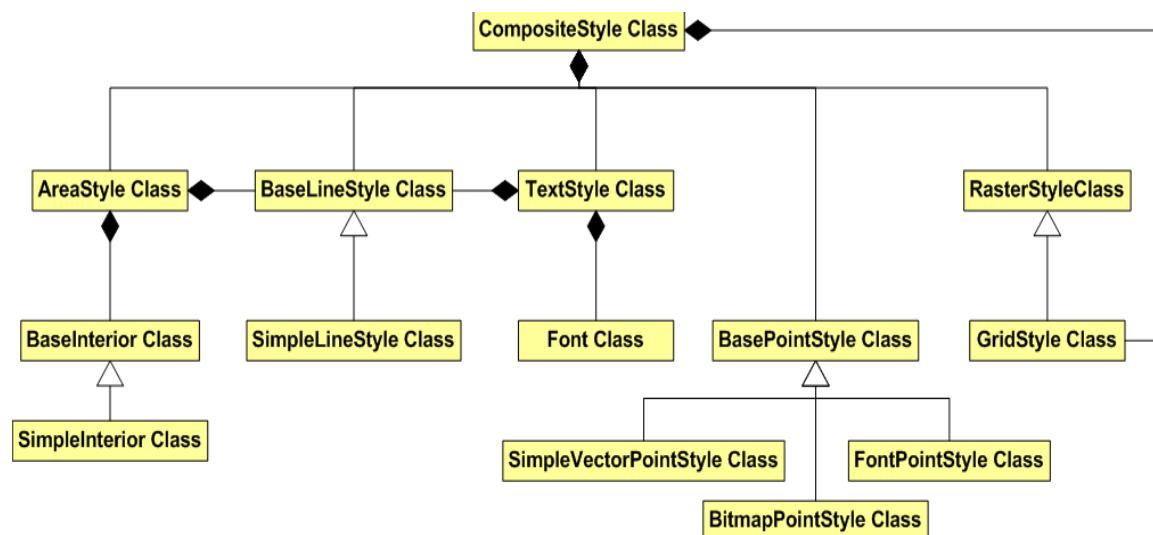
MapInfo.Styles 命名空间概述

MapInfo.Styles 命名空间强调了 MapXtreme 的新 Styles 对象模型。Style 类是所有样式的基类。从 Style 派生的类包括 AreaStyle、BaseLineStyle、BaseInterior、BasePointStyle、CompositeStyle、Font、RasterStyle 和 TextStyle。从 BaseLineStyle 派生的是 SimpleLineStyle 类。从 BaseInterior 派生的是 SimpleInterior 类。从 BasePointStyle 派生的是 BitmapPointStyle、FontPointStyle 和 SimpleVectorPointStyle 类。GridStyle 从 RasterStyle 派生以包括网格特定样式设置。此外，提供了 StockStyles 类用于创建公共样式类型。

用户无法实例化抽象样式类 Style、BaseLineStyle、BaseInterior 或 BasePointStyle。您必须创建特定的样式，如 SimpleLineStyle，或将其创建为 CompositeStyle。

Style 自己就是一个对象，不再存储于其它对象中了。每个包含几何体列的表还包括 Style 列（别名为 MI_Style），该列包含了数据类型 Style。对于样式重载，Feature 类提供了 FeatureStyleModifier 和 FeatureOverrideStyleModifiers。Style 对象模型还生成一些可用的集合类（样式库）以保存用于样式对话框控件的样式。

所有样式类支持以稀疏方式应用的能力。请参阅 FeatureOverrideStyleModifiers 一节。



Style 可在 MapXtreme 的多个区域使用，包括表示地理图元的直线、内部填充和点样式。Style 还整合了标注、文本、布局、主题、覆盖、图例和选择集。Style 属性的范围从标准颜色填充、直线宽度和磅值，到背景效果、位图点样式和旋转角度。实际上，可以想象出来的任意样式属性都可以应用于应用程序。此外，还可以更改全局样式或每个图元的样式，覆盖当前显示样式或进行永久性更改。

MapXtreme 包含了很多让您入门的样本样式。其中包括 170 多个内部填充图案，大约 120 个直线样线型和 70 个位图点样式图像。可以在任何可以创建位图的应用程序中创建位图图像，例如 MS Paint 或 Paint Shop Pro。尽管事实上对图像没有大小限制，不过 MapXtreme 显示图像的能力取决于可用的内存。图像不一定必须是方形，而且还可以具有最多 24 位颜色深度。要确保图像以所需的

高度和宽度显示，请选择 **BitmapPointStyles** 的“按实际大小显示”选项。图像创建后将其放置在 **CustSymb** 目录中。定制符号位于 **C:\Program Files\Common Files\MapInfo\MapXtreme\7.x\x\CustSymb** 文件夹。

此外，**MapXtreme** 包含了允许快速将这些符号添加到窗体或 **web** 应用程序的样式控件和对话框。请参阅 **Samples** 文件夹下的 **ChangeStyles** 和 **FeatureStyles** 样本应用程序，该文件夹还包含了本章中讨论的所有样式类。有关窗体样式控件和对话框的详细信息请参阅第 7 章：桌面应用程序、控件、对话框和工具。有关 **web** 控件的详细信息请参阅第 5 章：Web 应用程序、控件和工具。

有关支持的样式元素的可视化表示，请参阅附录 F：样式查找。

StyleFactory

StyleFactory 类可从 **MapInfo.Styles** 中获取，而 **MapInfo.Style** 包含用于从各种样式参数的类型生成 **MapInfo Style** 对象的方法。例如 **MapInfo.Styles.StyleFactory.FromMBstring** 将 **MapBasic** 样式子句作为字符串输入并返回 **CompositeStyle**。有关更多信息和代码示例，请参阅联机开发人员参考。

样式说明

AreaStyle

AreaStyle 类包含了绘图区域所用的样式属性。区域的绘制是通过 **BaseLineStyle** 和 **BaseInterior**。

BitmapPointStyle

MapInfo BitmapPointStyle 类包含了使用定制位图绘图点的样式属性。使用该类确定点的位置。

BitmapPointStyle 是支持的三种类型点样式之一，另外两种分别是 **FontPointStyle** 和 **SimpleVectorPointStyle**。

BitmapPointStyle 具有 **ShowWhiteBackground** 属性；如果设置为 **false**，则位图中的白像素为透明。默认情况下，**ShowWhiteBackground** 被设置为 **false**。例如，您或许想使用公司标识来表示全球办事处的位置，却不想遮盖即时区域内其它地图图元。其它设置可以控制 **BitmapPointStyles** 如何显示。有关更多详细信息请参阅联机帮助中的 **BitmapStyles** 枚举。

MapXtreme 包含可指导您入门的大量位图点样式。它们位于 **CustSymb** 目录中。此外也可将自己的位图图像添加到该目录。允许包含图像的最大数目是 **32,767**。

CompositeStyle

CompositeStyle 类包含了所有样式类型的集合，用于默认样式、修饰符样式和图层覆盖样式。

CompositeStyle 也可以用于描述 Collection 对象类型的样式。包含的样式类型是 AreaStyle、

BaseLineStyle 派生类、TextStyle、BasePointStyle 派生类、RasterStyle 和 GridStyle。

CompositeStyle 可以用任一或所有这些类型构造，但是必须至少包含一个样式类型。

例如，开发人员可以创建一个样式覆盖 (FeatureOverrideStyleModifier) 来改变图层中所有图元的外观。由于单一图层可包含点、线和区域，所以在生成样式覆盖时或许要指定点、线和区域样式。可以指定某一 CompositeStyle 对象中需要的所有样式类型，然后传递到 FeatureOverrideStyleModifier 构造函数。

SimpleInterior

MapInfo SimpleInterior 类包含了用于填充区域内部的样式属性。SimpleInterior 属性包括图案、前景和背景颜色、背景透明度。SimpleInterior 默认为白色实心内部图案。

Font

MapInfo Font 类包含了绘图文本所用的样式属性。字体属性包括黑体、斜体、下划线、删除线、阴影、光晕、全部大写、两倍行距、大小、前景和背景颜色。用户还可以更改字形（例如，Arial、Times New Roman）和字体大小。注意，轮廓字体属性已删除。要创建轮廓，请使用具有黑色背景颜色的光晕。

FontPointStyle

FontPointStyle 类包含了样式属性，主要使用 MapInfo.Styles.Font 类用于绘制点。用户可以定制点大小、字体颜色、旋转角度及其它字体属性。允许的最大点大小为 240 点。

GridStyle

GridStyle 是帮助程序类，包含与网格有关的显示样式信息（例如，颜色变化和空单元格颜色/透明度）。网格是连续色阶的地图，表示内插的数据值。有关网格的详细信息请参阅 [第 17 章：运用光栅和网格](#)。

RasterStyle

RasterStyle 是帮助程序类，包含光栅图像的显示样式信息（包括亮度、对比度、灰度、透明度（颜色，开/关）以及半透明度）。有关光栅图像的详细信息请参阅 [第 17 章：运用光栅和网格](#)。

Hillshade

Hillshade 是用于存储网格上 hill 底纹参数的帮助程序类。Hill shading，也称作浮雕底纹，可被添加到网格地图，以显示地图上的光源。这就为网格地图提供了更加广义的定义，对提升地图效果尤为有用。Hill shade 属性包括光源的水平和垂直角度，及垂直比例因子。有关网格的详细信息请参阅第 17 章：运用光栅和网格。

反射

Inflection 用于保存将颜色与值关联的单一变化点。网格具有表示其颜色变化的数组。网格地图是显示分级颜色在区域内更改的地图。变化产生了一个颜色与下一颜色的混合。有关网格的详细信息请参阅第 17 章：运用光栅和网格。

SimpleLineStyle

SimpleLineStyle 类包含了基于 MapBasic Pen 子句用于绘制折线的样式属性。该类用于地图图元，例如街道、电缆线路和区域周围的边界。描述 SimpleLineStyles 的属性包括线型、宽度（以像素或点为单位）和颜色。SimpleLineStyle 的默认设置是 1 像素宽的黑实线。SimpleLineStyle 的单位是像素（默认）或点。

MapInfo.Styles 命名空间中的 **LineWidth** 类是用于定义直线样式宽度和单位的帮助程序类。

BasePointStyle

这是所有 MapInfo 点样式的抽象基类。该类不能被实例化。SimpleVectorPointStyle、BitmapPointStyle 和 FontPointStyle 从该类派生。

BaseLineStyle

这是所有 MapInfo 直线样式的抽象基类。该类不能被实例化。SimpleLineStyle 从该类派生。

BaseInterior

这是所有 MapInfo 内部样式的抽象基类。该类不能被实例化。SimpleInterior 从该类派生。

StockStyles

该类包含了创建各种默认样式对象的静态方法，这些默认样式对象包括黑色、蓝色、红色和白色内部图案，黑色、蓝色和红色直线，空心内部方案和直线，默认字体和点样式。

```
SimpleLineStyle redLine = StockStyles.RedLineStyle( );
```

TextStyle

TextStyle 类包含用于绘制文本的样式属性。还包含标注线的 **MapInfo.Styles.Font** 类和 **BaseLineStyle** 派生类。**BaseLineStyle** 是可选的（**TextStyle** 可以包含也可以不包含一个这种线型）。

SimpleVectorPointStyle

该类包含使用 MapInfo 3.0 兼容专有字体用于绘制点的样式属性（**MapInfo.fnt** 随 **MapXtreme** 一起交付）。**SimpleVectorPointStyle** 属性包括了要为点绘制的实际符号的颜色、点大小和形状码。标准集包括符号 31 至 67。

 另一个符号字体集称为 **MapInfo Symbol**，是使用 **FontPointStyle** 类显示的 **TrueType** 字体集。

预定义的样式和 **StyleRepository** 类

MapXtreme 包含了大量位图图像，这些图像中含有大量可用作位图点样式的主题。此外还有 170 多种填充图案和直线样式可以使用。这些内容均通过应用程序来安装，且可以通过样式对话框（例如 **LineStyleDlg**）或各种 **StyleRepository** 类来访问。

有关支持的样式元素的可视化表示，请参阅 [附录 F: 样式查找](#)。

StyleRepository 类

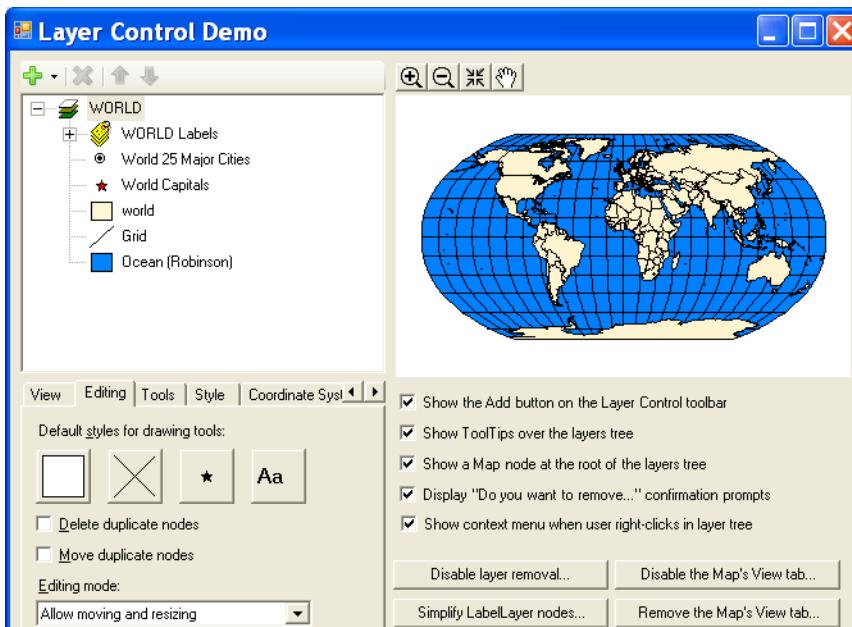
StyleRepository 类包含样式集合类（**VectorSymbolRepository**、**BitmapSymbolRepository**、**LineStyleRepository**、**InteriorStyleRepository**），这些类允许用户重复当前所有样式，并通过特定文件或目录中的新样式重新加载集合。**StyleRepository** 类还包含一个库 (**TrueTypeFontInfoRepository**)，它表示关于系统中已安装的 **TrueType** 字体的信息。

VectorSymbolRepository 表示一组源于 MapInfo 3.0 兼容符号集的符号。**BitmapSymbolRepository** 表示目前在 **CustSymb** 目录中的图像集合。**LineStyleRepository** 表示目前直线样式可以使用的线型集合。此外，**InteriorStyleRepository** 表示可用的内部图案集合。

使用样式

样式和图层控制

可以在设计阶段或运行时使用 **LayerControl** 修改和覆盖样式。有关 **MapInfo.Windows.Controls.LayerControl** 对象模型的介绍，请参阅样本应用程序 **LayerControl**。该代码示例直接在窗体上使用 **LayerControl** 对象，未使用相关的 **LayerControlDlg** 对话框进行说明，而使用独立的类。因为 **LayerControlDlg** 类公开 **LayerControl** 属性，所以该代码示例中说明的每个操作也可以应用到 **LayerControlDlg** 对象。



有关 **LayerControl** 的讨论请参阅 [第 7 章：桌面应用程序、控件、对话框和工具](#) 和 [第 24 章：Workspace 管理器](#)。

创建定制位图样式

可以在任何可以创建位图的应用程序中创建位图图像，例如 **MS Paint** 或 **Paint Shop Pro**。尽管事实上对图像没有大小限制，不过 **MapXtreme** 显示图像的能力取决于可用的内存。图像不一定必须是方形，而且还可以具有最多 24 位颜色深度。要确保图像以所要的高度和宽度显示，请选择 **BitmapPointStyles** 的“按实际大小显示”选项。图像创建后将其放置在 **CustSymb** 目录中。

覆盖样式

通过将新样式保存到表，可以永久更改图元的样式。通过覆盖当前样式（例如：范围主题覆盖区域对象的样式给其加底纹），也可更改当前显示（非永久显示）的图元样式。有关详细信息，请参阅 [第 14 章：使用主题和图例](#)。

也可覆盖标注样式。本节介绍了图元的主要样式覆盖类。有关本章中图元和标注的详细信息，请参阅 [Layers](#) 和 [Labels](#) 中的 [Mapping](#) 命名空间。

FeatureOverrideStyleModifiers

此类实现 [FeatureStyleModifier](#) 覆盖图元的样式。它的 [Style](#) 属性是组合样式对象，用于指定哪部分的图元样式要覆盖。

传递给 [Modify\(\)](#) 方法的样式对象内容会随绘制的每个图元而有较大变化。这样可增加包含样式修饰符的图层的绘制速度。因此，如果需要在应用程序的其他位置使用样式对象，请务必建立样式对象副本。还应注意：倘若是样式堆栈中的 [CompositeStyle](#) 传递给 [FeatureStyleModifier.Modify\(\)](#) 方法，则不会触发 [Changed](#) 事件。

代码示例：FeatureOverrideStyleModifier

以下示例说明了如何使用 [FeatureOverrideStyleModifier](#) 和图层 [FeatureStyleModifiers](#) 来更改地图内各种图元的样式。

在 [ChangeStyles](#) 示例应用程序的以下代码片断中，我们要用一个红色符号来覆盖世界首都图层，但点大小不变。

VB 示例：

```
'Get the layer we want
Dim _lyr As FeatureLayer = Me.mapControl1.Map.Layers("worldcap")

'Create a sparse point style
Dim vs As MapInfo.Styles.SimpleVectorPointStyle = New _
SimpleVectorPointStyle

'Just change the color and code and attributes flag to indicate that
vs.Code = 55
vs.PointSize = 25
vs.Color = System.Drawing.Color.Red

' And apply to the layer
Dim fsm As FeatureOverrideStyleModifier = New _
FeatureOverrideStyleModifier(Nothing, New _
MapInfo.Styles.CompositeStyle(vs))
_lyr.Modifiers.Append(fsm)
Me.mapControl1.Map.Zoom = New MapInfo.Geometry.Distance(6250, _
MapInfo.Geometry.DistanceUnit.Mile)
End Sub
```

空间对象和坐标系

本章介绍了 `MapInfo.Geometry` 命名空间并提供编写应用程序以创建和操控几何体对象的说明和示例。

在本章中：

- ◆ [MapInfo.Geometry 命名空间简介](#) 284
- ◆ [几何体](#) 284
- ◆ [将 FeatureGeometry 包括在地图中](#) 291
- ◆ [检查折线中的点](#) 292
- ◆ [坐标系](#) 293

MapInfo.Geometry 命名空间简介

MapInfo.Geometry 命名空间用于创建和操控几何体对象以及这些几何体对象使用的坐标系。**Geometry** 对象在地图中用于表示某一点（例如城市，用 **point** 对象表示）、边界线（例如国家边界，通过 **MultiCurve** 对象表示）和区域（例如国家或邮政编码地区，通过 **MultiPolygon** 对象表示）。

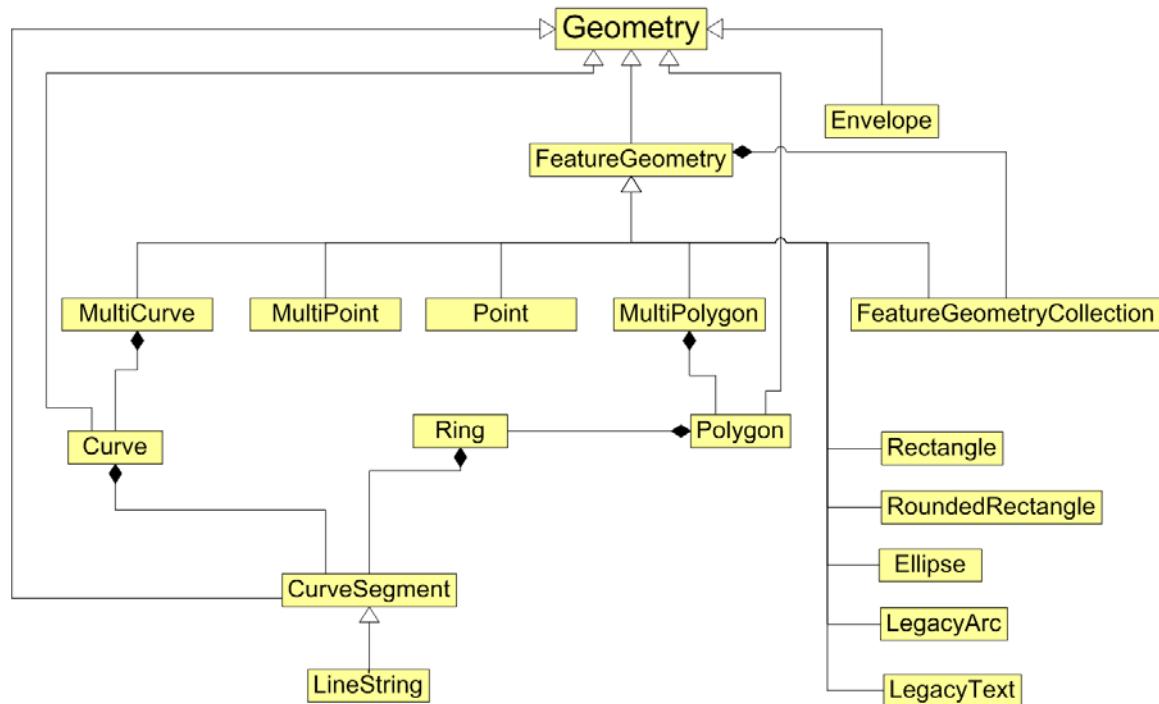
MapInfo.Geometry 命名空间中的类、接口和枚举定义了若干类型，用于表示用于显示地图上地理图元的几何体和坐标系。**Geometry** 模型提供在 **FeatureGeometry** 对象上支持 Z 和 M 值。接口允许创建和编辑几何体对象。诸如 **Buffer**、**Combine**、**Difference** 和 **Intersection** 这样的方法可以在单一对象或成对的对象上提供对象处理。

几何体

Geometry 类允许创建、编辑和对几何体对象的其它操控。从 **Geometry** 类继承和表示 **Geometry** 对象类型的类包括 **Point**、**MultiPoint**、**Polygon**、**MultiPolygon**、**Curve**、**CurveSegment**、**LineString** 和 **Ring**。以下传统类也继承自 **Geometry** 类：**Rectangle**、**RoundedRectangle**、**Ellipse**、**LegacyArc** 和 **LegacyText**。

Geometry 类表示 **MapInfo Geometry** 对象模型的最高级别。这是抽象类且不能被实例化。所有从该类派生的类包含与其坐标系有关的信息。所有类都能够制作其自身的副本，并可以比较自身与其它 **Geometry** 对象是否相等。

下图说明了 **Geometry** 模型的表示。



Geometry 对象

所有 MapXtreme 中的几何体对象都用无法更改的特定坐标系创建。如果需要改变对象的坐标系，则可以在新的坐标系中生成该对象的副本。

编辑 Geometry 对象

所有 **Geometry** 对象包含了用于检索到编辑器的接口的方法，该编辑器将对象置入编辑模式。一旦编辑完成，需要调用 **EditingComplete()** 方法来表示对象的编辑已完成。**EditingComplete()** 方法被调用时，**Geometry** 包含的对象顺序重新排序且到对象的所有引用被丢弃，要再次访问这些对象时所有引用需要重新建立。

例如，用户创建 **MultiPolygon**，然后编辑 **MultiPolygon**。如果用户无意间将内部环的节点移动到其包含的 **Polygon** 外部，则 **Polygon** 不再有效。**EditComplete** 被调用时，改组 **MultiPolygon** 内包含的所有对象来修复该问题。

MapXtreme 对象模型中的几何体对象在以下章节中说明。

FeatureGeometry 对象

FeatureGeometry 类是专为包含可以放置到表中且可以是 Features 和 FeatureCollections 一部分的类而设计。要在地图中显示的某些内容必须位于表中。FeatureGeometry 对象通过定义包括在表中。如果对象是 Geometry 的子类而不是 FeatureGeometry 的子类，则该对象无法保存到表或作为 Feature 或 FeatureCollection 的一部分包括。如果要尝试这种操作，则抛出异常或程序不能编译。FeatureGeometry 类像 Geometry 类等一样是抽象类，无法被实例化。

M 和 Z 值支持

图元几何体支持在对象的每个节点上读写 M 和 Z 值。

可通过扩展 MapXtreme Geometry 模型来实现支持读写线性对象 M 和 Z 值。FeatureGeometry 对象（Point、MultiPoint、MultiPolygon、MultiCurve 和 FeatureGeometryCollection）可保存每个节点的 X、Y、Z 和 M 值。

IsMeasured 和 Is3D 属性可确定对象是否有 M 或 Z 值。提供附加属性和方法以读取和修改每个节点的 M 或 Z 值。也可检索 M 和 Z 值的最小和最大范围。

MapXtreme 提供 FeatureGeometries 的创建和编辑功能。有关详细信息，请参阅“开发人员参考”中的 MapInfo.Geometry.FeatureGeometry 类。

线性网络应用程序中，MultiCurves 的 M 值提供有价值的信息，用于跟踪和管理资产、事件和条件。请参阅[第 21 章：线性参考](#)。

Point

Point 从 FeatureGeometry 类派生且表示地图上的某个点。Point 可以包括在 MultiPoint 集合内，然后作为一个整体操作。

使用以下代码示例对创建 Point 建模：

```
using MapInfo.Geometry;
using Mapinfo.Design.Windows;

CoordSys longLatNad83;
CoordSysFactory coordSysFactory = new CoordSysFactory();
longLatNad83 = coordSysFactory.CreateLongLat
    (MapInfo.Geometry.DatumID.NAD83);
DPoint point = new DPoint(0.0, 0.0);
Point pointGeometry = new Point(LongLatNad83, point);
```

MultiPoint

MultiPoint 包含了无序和不连续的 Point 集且可用于在多个点上执行多个操作。

使用以下代码示例对创建 MultiPoint 对象建模：

```
using MapInfo.Geometry;
```

```
CoordSys longLatNad83;
CoordSysFactory coordSysFactory = new CoordSysFactory();
longLatNad83=coordSysFactory.CreateLongLat
    (MapInfo.Geometry.DatumID.NAD83);
MultiPoint multiPointGeometry = new MultiPoint
    (longLatNad83, pointArray);
```

其中 `pointArray` 是 `DPoints` 的数组。

MultiCurve

`MultiCurve` 类从 `FeatureGeometry` 类派生并包含一组可能断开的 `Curve`。这些 `Curve` 可以用多种方式交互，也可以连接或断开，还可以彼此相交或重叠。

尽管 `Geometry` 对象模型支持每个 `Curve` 包含多个 `CurveSegment`，但当前版本的 `MapInfo` 引擎对于 `FeatureGeometry` 的一部分（即，`MultiCurve`）的每个 `Curve` 仅限于包含一个 `CurveSegment`。该限制从当前的 TAB 文件格式派生，对于本版本的 `MapInfo`，这些限制的绝大部分保持不变。因此，该限制仅涉及 `FeatureGeometry` 对象。

构造 `MultiCurve` 时，构造函数获取一个或多个 `Curve`，每个 `Curve` 都可能包含多个 `CurveSegment`，则改变包含在构造的 `MultiCurve` 中的实际 `Curve` 使得每个 `Curve` 始终仅包含一个 `CurveSegment`。目前，现有 `CurveSegment` 的类型只有 `LineString`。包含多个 `LineString` `CurveSegment` 的 `Curve` 将 `LineString` 组合以形成一个大的 `LineString`。

通过调用 `EditingComplete()` 表示的编辑完成时，添加到 `MultiCurve` 中且包含多个 `CurveSegments` 的任何 `Curve` 按照与上述类似的方式改变，从而生成包含单一 `CurveSegments` 的 `Curves`。`MultiCurves` 中包含的 `Curves` 始终仅包含一个 `CurveSegment` 的限制会像引入新类型的 `CurveSegments`（例如 `EllipticalArcs`、`CircularArcs` 和 `Splines`）一样从以后版本的 `MapInfo` 中去除，且 TAB 文件格式也会改变。此外，在构造期间且编辑完成时，会从 `MultiCurve` 自动删除任何空的 `Curve`。

`Line` 对象由存在于 `MapInfo` TAB 文件中的两个点组成并成为 `MultiCurve FeatureGeometry` 对象。通过使用 `MultiCurve` 的 `IsLegacyLine` 属性，可以作为两点 `Lines` 来检测这些对象。

有关创建和编辑 `MultiCurve` 对象的代码示例，请参阅“开发人员参考”。

MultiCurves 上的度量值

`Geometry` 对象模型支持 `FeatureGeometry` 对象上的 `M` 和 `Z` 值。`M` 或度量值保存 `MultiCurve` 对象上节点处的数据，该数据可说明想用地图表示和分析的任意事物，包括实际资产、条件或事件。`M` 值在实现线性参考和动态分段上发挥重要作用。有关详细信息，请参阅 [第 21 章：线性参考](#)。

Curve 排序顺序

与编辑完成时传递到构造函数的 `Curve` 数组相比较，`MultiCurve` 中 `Curve` 的顺序可以在构造期间改变。因此，编辑完成后（即，调用 `EditingComplete()` 后），空 `Curve` 的删除，当前实现中的限制以及编辑之前和期间对包含在 `MultiCurve` 中 `Curve` 的任何引用不再有效。如果引用这些对象，则抛出 `ObjectDisposedException`。编辑完成后，部分 `FeatureGeometry` 应该重新获取以获得有效的引用。

Curve 排序顺序会在 **MultiCurve** 上调用线性参考操作时成为重要因素。**MapXtreme** 包含 **MapInfo.LinearReferencing.ILinearReferencing** 接口以处理单独曲线的排序顺序。若指定无排序顺序，**MapXtreme** 将首先返回最长曲线，同时剩余的曲线以未知的顺序返回。例如，在使用无序的 **MultiCurve** 上的 **CalculateMissingMeasures** 时，**MapXtreme** 可根据 **MultiCurve** 中的节点位置计算出节点错误的 M 值。提供正确的排序顺序将消除这个问题。

有关详细信息，请参阅[曲线顺序](#)。

LineStrings

LineString 是以线性方式连接的连续点的有方向集合。**LineString** 中任意两个连续点都由直线连接。**LineString** 可以是 **Curve** 或 **Ring** 的一部分，也可以作为独立的 **Geometry** 存在。作为 **Curve** 或 **Ring** 一部分的 **LineString** 继承其容器的坐标系。独立的 **LineStrings** 可以无内容。如果 **LineString** 包含在未处于编辑模式的 **Curve** 或 **Ring** 中，则 **LineString** 不可以无内容，且必须至少包含两个点。

有关代码示例，请参阅“[开发人员参考](#)”。

Rectangle

Rectangle Geometry 包含 2 个点，用于表示 **Rectangle** 的左下角和右上角。其它 2 个点是隐含的。**Rectangles** 始终为轴对齐，始终显示为矩形形状（与坐标系无关），且不投影。也不包含由坐标系表示的任意弯曲。

有关代码示例，请参阅“[开发人员参考](#)”。

RoundedRectangle

Rounded Rectangle 行为与 **Rectangle** 非常类似，不过在仅显示图元的阶段，这些矩形的角显示为圆形。这些角显示为四分之一圆且圆的半径由 **CornerRadius** 参数控制。

因为 **RoundedRectangle** 对象像矩形对象一样由 2 个点定义，且始终显示为轴对齐且未投影，所以主要将其设计用于外观显示。如果多项操作可以内部使用 **Rectangle** 对象（例如 **Combine**），则 **Rectangle** 的 **MultiPolygon** 副本用于这些操作。生成的 **MultiPolygon** 包含 5 个点（第一个点与最后一个点相等）且受坐标系影响。在某些实例中，转换的 **Rectangle** 可以不再显示为矩形。使用 **CreateMultiPolygon** 方法将 **RoundedRectangle** 转换为 **FeatureGeometry** 对象。

有关代码示例，请参阅“[开发人员参考](#)”。

Ellipse

Ellipse 包括在由 **DRect** 定义的轴对齐矩形中。**DRect** 由 2 个点定义，即，矩形的两个对角，其中隐含矩形的其它 2 个角。与坐标系无关，**Ellipse** 显示为不投影，还可以显示为可以由坐标系表示的任何倾斜。

因为 **Ellipse** 对象由 2 个点定义，且始终显示为轴对齐且未投影，所以主要将其设计用于外观显示。如果多项操作可以内部使用 **Ellipse** 对象，则 **Ellipse** 的 **MultiPolygon** 副本用于这些操作。生成的 **MultiPolygon** 会受到坐标系影响，且在某些情况下可能显示的不再是完美的椭圆。

有关代码示例，请参阅“[开发人员参考](#)”。

LegacyArc

LegacyArc 对象是 **Ellipse** 的一部分，并且是通过 **DRect**、开始角度和结束角度定义的。构造 **Ellipse**，使其包括在由 **DRect** 定义的矩形中。与所用的坐标系无关，包括 **Ellipse** 的矩形是轴对齐的，且始终显示为矩形。角度以度为单位度量，沿正 X 轴角度为零，而逆时针方向的角度为正。角度仅按十分之一度的分辨率存储，且角度值的范围为 0.0 到 360.0。

因为 **LegacyArc** 对象由 2 个点（构成 **DRect**）和角度定义，且始终显示为轴对齐，所以主要将其设计用于外观显示。如果多项操作可以内部使用 **LegacyArc** 对象，则 **LegacyArc** 的 **MultiCurve** 副本用于这些操作。这有时会导致意外结果。

有关代码示例，请参阅“开发人员参考”。

LegacyText

LegacyText 对象与 **MapInfo Professional** 的文本对象等价。如果给定的数据库不支持 **Text**，则使用这种格式时 **LegacyText** 对象会丢失。**LegacyText** 对象用指定的左下角标记点放置在地理尺寸的矩形内。文本的点大小由矩形中最合适放置的内容决定。

LegacyText 对象不完全符合 **Geometry** 模型。**Geometry FeatureGeometry** 类上可用的多个方法（例如 **Combine**）对 **LegacyText** 不起作用，且会抛出 **NotSupportedException**。在 **Geometry** 列的 **MapInfo** 本地 TAB 文件中不存在 **Text** 对象。**LegacyText** 类提供了访问这些对象的方法。有关 **LegacyText** 对象的特定行为请参阅联机参考。

Geometry 对象

不是 **FeatureGeometry** 对象的 **Geometry** 对象需要被转换为地图上显示的对应 **FeatureGeometry** 对象。大多数 **FeatureGeometry** 类包含了获取适当 **Geometry** 对象并创建新 **FeatureGeometry** 对象的构造函数：

```
using MapInfo.Geometry;  
  
Curve curve = new Curve(csys, lineString);  
MultiCurve multiCurve = new MultiCurve(curve.CoordSys, curve);
```

以上代码使用在 **CoordSys (csys)** 和 **LineString (lineString)** 代码之外定义的参数创建 **Curve**。然后使用 **Curve** 的 **CoordSys** 属性及 **Curve** 本身创建新的 **MultiCurve**。

在以上示例中，与从对象创建的所有 **FeatureGeometries** 一样，因为引用无法共享，所以会创建原始对象的副本。

Curve

Curve 类从 **CurveSegmentList** 类继承，表示相邻的线性 **Geometry**。**Curve** 包含必须保持相邻的 **CurveSegment** 的集合。该类包括在允许以后扩展的模型中且是 OGC 标准的一部分。

使用以下代码示例对创建 **Curve** 建模：

```
using MapInfo.Geometry;
```

```
DPoint [] points = new DPoint [4];  
  
points [0] = new DPoint (-88.135215, 43.998892);  
points [1] = new DPoint (-104.875119, 43.998892);  
points [2] = new DPoint (-120.242895, 47.048364);  
points [3] = new DPoint (-89.135215, 46.998892);  
  
LineString lineString = new LineString(csys, points);  
Curve curve = new Curve(csys, lineString);
```

CurveSegment

目前，CurveSegment 只可以是 LineString。该类设计用于以后扩展类似的产品，从而包括 Spline、CircularArc 和 EllipticalArc CurveSegment。Curve 和 Ring 由 CurveSegment 组成。

Ring

Ring 是必须保持相邻且封闭的 CurveSegment 集合。

使用以下代码示例对创建 Ring 建模：

```
using MapInfo.Geometry;  
  
dPoints = new DPoint [102];  
dPoints [0] = new DPoint (-109.171279, 49.214879);  
dPoints [1] = new DPoint (-109.169283, 49.241794);  
...  
dPoints [101] = new DPoint (-109.171279, 49.214879);  
Ring newRing = new Ring(longLatNad83, CurveSegmentType.Linear, dPoints);
```

Polygon

Polygon 是 Ring 组成的对象。多边形必须至少有一个 Ring，用于定义 Polygon 的外部边界。其它 Ring 可以包括在该环内部，然后在其中定义 Polygon 中的孔。一旦 Ring 放置在另一个 Ring 的内部，则该对象成为 MultiPolygon。

使用以下代码示例对创建 Polygon 建模：

```
using MapInfo.Geometry;  
  
DPoint [][] points = new DPoint [1] [];  
points [0] = polyPointArrays [0];  
Polygon polygon = new Polygon  
(longLatNad83, CurveSegmentType.Linear, polyPointArrays [0]);
```

将 FeatureGeometry 包括在地图中

几何体创建后，需要将几何体添加到地图中，然后才能显示、选择、标注该几何体，或在该几何体上执行任何与地图有关的其它操作。

```
Public Shared Sub MapInfo_Mapping_HowDoICreateFeatureAddToMap (ByVal
mapControl1 As MapControl, ByVal connection As MConnection, ByVal x As _
Double, ByVal y As Double)
    Dim map As Map = mapControl1.Map

    ' uses wldcty25 as a template
    Dim table As Table =
        MapInfo.Engine.Session.Current.Catalog.GetTable("wldcty25")

    ' create a temp table and add a featurelayer for it
    Dim coordSys As CoordSys = map.GetDisplayCoordSys()
    Dim tableInfo As TableInfoMemTable = New TableInfoMemTable("temp")
    tableInfo.Temporary = True

    ' add Geometry column
    Dim column As Column

    ' specify coordsys for object column
    column = New GeometryColumn(coordSys)
    column.Alias = "MI_Geometry"
    column.DataType = MIDbType.FeatureGeometry
    tableInfo.Columns.Add(column)

    ' add style column
    column = New Column
    column.Alias = "MI_Style"
    column.DataType = MIDbType.Style
    tableInfo.Columns.Add(column)

    Dim pointTable As Table =
        Session.Current.Catalog.CreateTable(tableInfo)

    ' Set the location and display style of the point
    Dim Geometry As FeatureGeometry =
        New MapInfo.Geometry.Point(coordSys, x, y)
    Dim vStyle As SimpleVectorPointStyle =
        New SimpleVectorPointStyle(37, Color.Red, 14)
    Dim cStyle As CompositeStyle =
        New MapInfo.Styles.CompositeStyle(vStyle)

    ' Update the table with the location and style of the new feature
    Dim cmd As MICommand = connection.CreateCommand()
    cmd.Parameters.Add("Geometry", MIDbType.FeatureGeometry)
    cmd.Parameters.Add("style", MIDbType.Style)
    cmd.CommandText = "Insert Into temp (MI_Geometry,MI_Style) values _"
        (Geometry,style)"
```

```
cmd.Prepare()
cmd.Parameters(0).Value = Geometry
cmd.Parameters(1).Value = cStyle
Dim nchanged As Integer = cmd.ExecuteNonQuery()
cmd.Dispose()

' add the table to the map
map.Layers.Add(New MapInfo.Mapping.FeatureLayer(pointTable))
End Sub
```

检查折线中的点

下面的代码示例显示如何确定某一个点是位于 FeatureGeometry (Multipolygon) 的边界内部、边界线上还是边界线外部。

```
Public Shared Sub MapInfoGeometryContainsPoint()
Dim coordSysFactory As CoordSysFactory = Session.Current.CoordSysFactory
Dim coordSys As CoordSys =
    coordSysFactory.CreateLongLat(MapInfo.Geometry.DatumID.NAD83)

Dim points(6) As DPoint
    points(0) = New DPoint(-0.705036, -0.122302)
    points(1) = New DPoint(-0.446043, 0.486811)
    points(2) = New DPoint(0.235012, 0.36211)
    points(3) = New DPoint(0.422062, -0.304556)
    points(4) = New DPoint(-0.244604, -0.71223)
    points(5) = New DPoint(-0.705036, -0.122302)
Dim multiCurve As MultiCurve = New _
    MultiCurve(coordSys, CurveSegmentType.Linear, points)
Dim multiPolygon As MultiPolygon = New _
    MultiPolygon(coordSys, CurveSegmentType.Linear, points)

Dim insidePoint As DPoint = New DPoint(-0.115108, 0.160671)
Dim boundaryPoint As DPoint = New DPoint(-0.446043, 0.486811)
Dim outsidePoint As DPoint = New DPoint(-1.103118, 0.021583)

If multiPolygon.ContainsPoint(insidePoint) Then
    Console.WriteLine("Points inside area inclosed by closed _
        (GeometryDimension 2) objects are contained")
End If
If Not multiCurve.ContainsPoint(insidePoint) Then
    Console.WriteLine("But this is not true for linear _
        (GeometryDimension 1) objects")
End If
If multiPolygon.ContainsPoint(boundaryPoint) Then
    Console.WriteLine("Points on the boundary of closed objects _
        are contained")
End If
If multiCurve.ContainsPoint(boundaryPoint) Then
    Console.WriteLine("Points lying on linear objects are contained")
```

```
End If
If Not multiPolygon.ContainsPoint(outsidePoint) Then _
    Console.WriteLine("Point completely outside closed objects _"
        "are not contained")
End If
If Not multiCurve.ContainsPoint(outsidePoint) Then _
    Console.WriteLine("Point completely outside linear objects _"
        "are not contained")
End If
End Sub
```

坐标系

坐标系描述了包含特定对象或对象集合的域。坐标系允许以特定的方式描绘对象或描述的对象。**CoordSys** 类包含了允许创建、操控和编辑坐标系的方法、属性和接口。

如要创建 **Geometries**，则在创建对象时指定的特定坐标系中创建。对象无法更改创建 **Geometries** 的坐标系。它们只可以复制到另一个坐标系。

CoordSys 类简化了对坐标系的创建和操控。**Coordsys** 类使用投影文件的 XML 版本 (C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x\MapInfoCoordinateSystemSet.xml)。

CoordSysFactory 对象包含注册的坐标系。通过加载一个或多个 XML 投影文件或使用 **RegisterCoordSys** 或 **RegisterCoordSysInfo** 方法，可以注册 **CoordSys** 定义。从工厂或从代码-代码空间 (EPSG, SRID)、PRJ 字符串、MapBasic 字符串和其它 Factory 创建方法来创建 **CoordSys** 对象。在 **CoordSys** 类中也有 Military Grid Reference System 转换方法。

创建 **CoordSys** 对象

以下示例代码显示了创建 **CoordSys** 对象的几种不同方式：使用 MapInfo 代码空间；通过 EPSG；作为来自 PRJ 字符串的经度/维度；来自 MapBasic 字符串；以及通过 SRID。

VB 示例：

```
Public Shared Sub MapInfoGeometryCreateCoordSys()
    Dim factory As CoordSysFactory = Session.Current.CoordSysFactory

    ' create CoordSys objects from srsName
    Dim csysWGS84 As CoordSys = factory.CreateCoordSys("EPSG:4326")
    Dim csysNAD83 As CoordSys = factory.CreateCoordSys_
        ("mapinfo:coordsys 1,74")
    Dim csysNAD27 As CoordSys = factory.CreateCoordSys("SRID:8260")

    ' create CoordSys objects from code/codeSpace
    csysWGS84 = factory.CreateCoordSys("4326", CodeSpace.Epsg)
    csysNAD83 = factory.CreateCoordSys("coordsys 1,74", CodeSpace.MapInfo)
    csysNAD27 = factory.CreateCoordSys("8260", CodeSpace.Srid)
```

```
' create CoordSys objects from user-defined parameters
Dim dat As Datum = factory.CreateDatum(DatumID.WGS84)
csysWGS84 = factory.CreateCoordSys(CoordSysType.LongLat, _
    dat, 0, 0, 0, 0, 0, 0, 0, 0, 0, nothing)
dat = factory.CreateDatum(DatumID.NAD83)
csysNAD83 = factory.CreateCoordSys(CoordSysType.LongLat, _
    dat, 0, 0, 0, 0, 0, 0, 0, 0, 0, nothing)
dat = factory.CreateDatum(DatumID.NAD27ContinentalUS)
csysNAD27 = factory.CreateCoordSys(CoordSysType.LongLat, _
    dat, 0, 0, 0, 0, 0, 0, 0, 0, 0, nothing)
' create Long/Lat coordinate system
csysWGS84 = factory.CreateLongLat(DatumID.WGS84)
csysNAD83 = factory.CreateLongLat(DatumID.NAD83)
csysNAD27 = factory.CreateLongLat(DatumID.NAD27ContinentalUS)

' create from MapBasic string
Dim csysRGF93 As CoordSys = _
    factory.CreateFromMapBasicString("CoordSys Earth Projection 3, _"
    & "33, ""m"", 3, 46.5, 44, 49, 700000, 6600000")
' create from PRJ string
csysNAD83 = factory.CreateFromPrjString("1, 74")
Sub
```

更改 Geometry 对象的坐标系

下一示例说明了如何将 Geometry 对象从一个坐标系转换到另一个坐标系。

VB 示例：

```
Public Shared Sub MapInfoGeomTryCoordSys(ByRef coordSys As _
    CoordSys, ByRef points() As DPoint, ByRef alternateCoordSys _ 
    as CoordSys
' All Geometry constructors require a CoordSys parameter
' Note that the points array is assumed to be in coordSys
    Dim lineString As LineString = New _
        LineString(coordSys,points)

' The Geometry has a reference to the CoordSys used during
' construction.Unlike the coordinate data represented by the
' points array, the CoordSys' is not copied
    If ReferenceEquals(coordSys, lineString.CoordSys) Then
        Console.WriteLine("Geometry objects hold a reference to _
            the CoordSys used during construction")
    End If

' if you want to convert the object to another coordinate
' system, you need to make a new copy using one of the copy methods
    If Not coordSys.Equals(alternateCoordSys) Then
        Dim newGeometry as MapInfo.Geometry.Geometry = _
            lineString.Copy(alternateCoordSys)
    End If
End Sub
```

确定 MapControl 中的地图坐标系

下面的代码示例说明了如何确定 Map 对象的坐标系。

VB 示例：

```
Public Shared Sub _
    MapInfo_Mapping_FeatureViewerGetDisplayCoordSys(ByVal map As Map)
    ' Load the Default Projection File so that we can get the name.

    MapInfo.Engine.Session.Current.CoordSysFactory.LoadDefault_
        ProjectionFile()

    ' Get the Coordinate System object for current map in the MapControl
    Dim mapCoordSys As MapInfo.Geometry.CoordSys = _
        map.GetDisplayCoordSys()

    ' Assign the name of the Coordinate System to a string variable.
    ' note: the CoordSysName function will return a blank string if the
    ' Coordinate System is not found in the current CoordinateSystemSet
    ' (loaded by the LoadDefaultProjectionFile above).
    Dim mapCoordSysName As String = _
        MapInfo.Engine.Session.Current.CoordSysFactory.CoordSys_
            Name(mapCoordSys)
End Sub
```

将坐标系添加到 MapXtreme

如果 MapInfoCoordinateSystemSet.xml 文件不包含符合您需要的坐标系，您可以将其添加到 MapXtreme。此功能支持添加 EPSG 代码和 SRID 代码以扩展 MapXtreme 的功能。

EPSG 代码代表一个坐标系集合（称作 codespace），这个集合是在国际石油和天然气生产商联盟 (OPG) 的支持下在 EPSG Geodetic 参数数据集中维护的。2005 年，OPG 的测量定位委员会从欧洲石油勘探组接管此任务。

SRID 代码是唯一的空间参考编号，这些编号引用 Oracle Spatial 表的 codespace。

（MapXtreme 支持名为 MapInfo 的第三个 codespace。）

MapXtreme 为您提供了许多常用的 EPSG 和 SRID 地图。如果需要将不同的 EPSG 或 SRID 代码注册到特定的坐标系，此功能将可为您提供两种方法来完成此操作。

若要扩展 MapXtreme 的功能来使用任何 EPSG 或 SRID codespace，您可以通过编程方式添加信息，这种情况下，坐标系信息保持的时间长度与 MapXtreme 会话的长度相等。或者，您可以将其添加到您的 Web 应用程序的 web.config 文件中或桌面应用程序的 app.config 文件中，用作一个更为永久的解决方案。下面将对其进行逐个讨论。

按编程方式注册 EPSG 和 SRID 代码

MapInfo.Geometry.CoordSysFactory 类包含允许您将 EPSG 和 SRID 代码注册到特定坐标系的方法。

RegisterEPSGCode() 和 RegisterSRIDCode() 分别接受两个参数：一个作为 codespace 的 EPSG 或 SRID 代码，另一个是第一个参数映射的坐标系信息。

下面的示例说明了如何将伪代码注册到 Long/Lat NAD83 坐标系。

VB 示例：

```
Public Shared Sub MapInfo_Geometry_RegisterEPSGCode()
    Dim factory As CoordSysFactory = Session.Current.CoordSysFactory
    ' create CoordSys objects from srsName
    Dim csysNAD83 As CoordSys =
        factory.CreateCoordSys("mapinfo:coordsys 1,74")
    ' 9998 is a fictional code for demonstration purposes
    Try
        factory.RegisterEPSGCode(9998, csysNAD83)
    Catch ae As ApplicationException
        'code already exists.Codes cannot be duplicated
    End Try
End Sub
```



如果 EPSG 或 SRID 代码已存在，将会抛出一个指示此事实的异常。

若要确定是否支持 MapInfo、EPSG 或 SRID codespace 的坐标系，请调用此方法：

- `MapInfo.Geometry.CoordSys.Code(codespace)` .

此方法返回匹配的或为空的第一个（或仅）codespace 的实例，但条件是该实例不存在。

类似地，要返回与输入 codespace 匹配的列表中的第一个 SRSName，请调用此方法：

- `MapInfo.Geometry.CoordSys.SRSName(codespace)` .

SRSName（空间参考系）代表用 GML（地理标记语言）编写的坐标参考系的名称。这是典型的坐标系友好名称，不是参数值列表。

若要获得映射到特定坐标系的所有代码和坐标系的列表，**MapXtreme** 提供了以下方法：

- `MapInfo.Geometry.CoordSys.Codes(codeSpace)`
- `MapInfo.Geometry.CoordSys.SrsNames(codeSpace)`

记住，您按编程方式添加的坐标系信息只能在 **MapXtreme** 会话的生命周期中维护。

将 EPSG 和 SRID 代码注册到 Web 或桌面配置文件中

将 EPSG 或 SRID 代码添加到 **MapXtreme** 的第二种更为永久的方法是将这些信息添加到您的 Web 应用程序的 `Web.config` 文件或您的桌面应用程序 `app.config`¹ 文件中。下面按粗体显示的代码显示您必须复制粘贴到 config 文件中的信息。代码的解释如下。

-
1. 如果桌面应用程序没有任何 `app.config` 文件，您可以通过从 Visual Studio 应用程序配置文件模板将该文件添加到项目中来进行创建。

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
        <section name="MapInfo.CoreEngine"
type="MapInfo.Engine.ConfigSectionHandler, MapInfo.CoreEngine,
Version=6.8.0.536, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" />
    </configSections>
    <appSettings>
        <add key="MapInfo.Engine.Session.UseHttpContext" value="false"
/>
    </appSettings>
    <MapInfo.CoreEngine>
        <EPSC_Code_Mappings>
            <EPSC_Code_Mapping>
                <srsName>My Custom CRS</srsName>
                <srsID>
                    <code>coordsys 8,74,8,-
110.083333333,47.5,0.9999375,2624666.667,328083.3333</code>
                    <codeSpace>mapinfo</codeSpace>
                    <remarks>My Custom CRS</remarks>
                </srsID>
                <EPSC_Codes>
                    <EPSC_Code>9987</EPSC_Code>
                    <EPSC_Code>9988</EPSC_Code>
                    <EPSC_Code>9989</EPSC_Code>
                </EPSC_Codes>
            </EPSC_Code_Mapping>
        </EPSC_Code_Mappings>
        <SRID_Code_Mappings>
            <SRID_Code_Mapping>
                <srsName>My Custom CRS</srsName>
                <srsID>
                    <code>coordsys 8,74,8,-
114.083333333,47.5,0.9999375,2624666.667,328083.3333</code>
                    <codeSpace>mapinfo</codeSpace>
                    <remarks>My Custom CRS</remarks>
                </srsID>
                <SRID_Codes>
                    <SRID_Code>9990</SRID_Code>
                    <SRID_Code>9991</SRID_Code>
                    <SRID_Code>9992</SRID_Code>
                </SRID_Codes>
            </SRID_Code_Mapping>
        </SRID_Code_Mappings>
    </MapInfo.CoreEngine>
</configuration>
```

上述代码显示了要添加两段信息。其中一段信息用于识别正确的 CoreEngine dll 和版本号¹，另一段信息用于为 EPSG 和 SRID 代码映射添加元素。

EPSG 代码映射包含 SRSName、SRS ID 和 EPSG 代码。SRID 是使用坐标系的参数和 codespace 进一步定义的。

SRID 代码映射类似于 EPSG 代码映射，但有一点不同是，它引用 Oracle Spatial 标识号。

有关坐标系的详细信息，请参阅[附录 H: 坐标系的元素](#)。有关 MapInfo codespace 的信息，请参阅[附录 G: 定义 MapInfo Codespace](#)。

-
1. 若要确定 MapInfo.CoreEngine 程序集的正确版本，请从“开始”菜单中，选择“运行”，然后在“运行”对话框中键入 Assembly。每个注册的 MapXtreme 程序集都在全局程序集缓存列表中列出。

运用光栅和网格

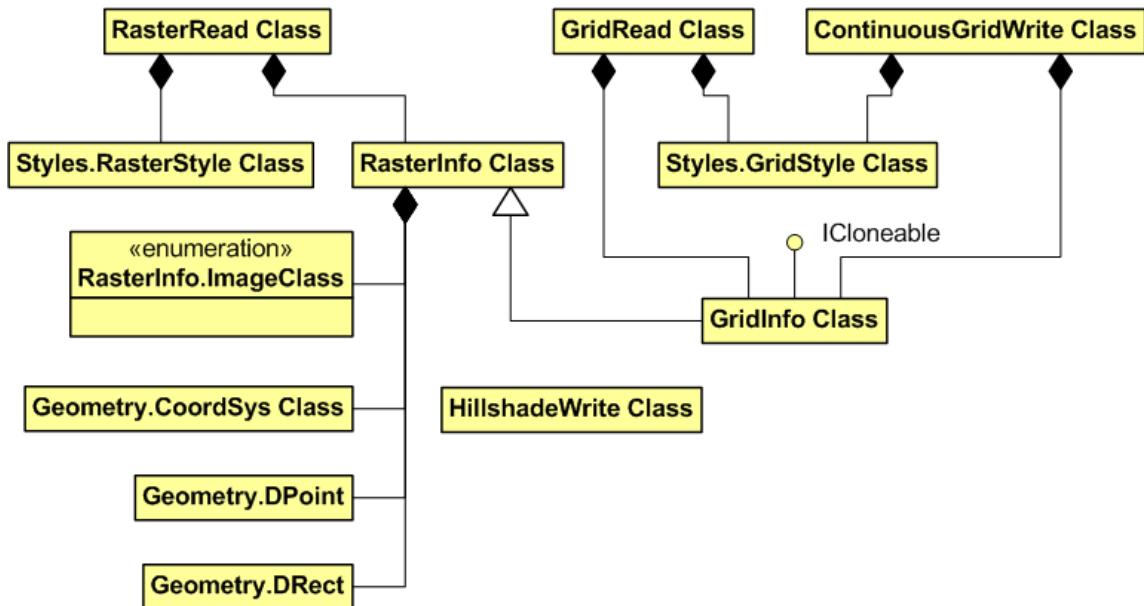
MapInfo.Raster 命名空间包含的类用于对 **MapXtreme** 中光栅和网格图像的使用和显示进行控制。光栅是由像素组成的计算机图形，由像素渲染整个图像。很多卫星图像渲染为光栅图像。网格图像是主题地图，显示了表示内插值信息的连续色阶。

在本章中：

◆ MapInfo.Raster 命名空间概述.....	300
◆ 光栅图像.....	300
◆ 光栅句柄.....	303
◆ 光栅句柄属性.....	304
◆ 配置自定义光栅处理程序.....	305
◆ 网格图像.....	306
◆ 创建网格.....	308
◆ 网格外插器.....	309
◆ 网格样式.....	310

MapInfo.Raster 命名空间概述

MapInfo.Raster 命名空间显示了 Pitney Bowes Software C/C++ Raster 和 Grid Engine API 的全部功能。光栅图像是位图，提供了地图的有用背景和参考图层。网格图像是一种主题，显示了多个图像之间的连续色阶。色阶表示对基础数据的解释。网格图像是将数据与它们关联的光栅。常见的示例为剖面地图。光栅不包含任何基础数据。



光栅图像

光栅图像为地图生成了非常好的背景。例如，反映真实世界详细信息（例如建筑物、厂房、植被）的航空照片就非常适合做地图的基本图层。扫描的纸式地图是光栅图像的另一个示例。将光栅图像用作基本图层并重叠诸如街道网络、表示客户的点位置和邮政边界等矢量数据来创建有用且在视觉上吸引人的地图。

与矢量数据一起使用的光栅图像必须注册，以便图像上已知的图形点与矢量数据上相同的图元一致。此外，要显示在 **MapXtreme** 中的公司徽标和其它美工图形即使不是真实的地理参考数据，也必须注册到地面上的某些位置。目前与注册文件一起提供了很多可用的光栅图像。例如 GeoTIFF、ADRG、ASRP、CADRG 和 CIB。要注册光栅图像，可以将其加载到 **MapInfo Professional** 中并在其中注册。注册信息存储在 **.TAB** 文件。

以下是 **MapXtreme** 中支持的光栅图像格式：

- TIFF 和 GeoTIFF (*.tif)
- MrSID (*.sid)
- ECW (*.ecw)

- Spot (*.bil)
- JPEG (*.jpg)
- JPEG2000 (*.jp2, *.j2K)
- PCX (*.pcx)
- GIF (*.gif)
- Windows Bitmap (*.bmp)
- PNG (*.png)
- Photoshop (*.psd)
- Targa (*.tga)
- Windows Metafile (*.wmf)
- Windows Enhance Metafile (*.emf)
- Wireless BMP (.WBMP)
- Vertical Mapper Continuous Grid (*.grd)
- Vertical Mapper Classified Grid (*.grc)
- ADRG - ARC Digitized Raster Graphics (*.gen)
- ASRP - ARC Standard Raster Product (多种文件扩展名)
- CADRG - Compressed ARC Digitized Raster Graphics (*.gen)
- CIB - Controlled Image Base (多种文件扩展名)
- NITF - National Imagery Transmission Format (*.ntf)

如果安装了定制光栅句柄，则系统可以支持其它的光栅格式。

光栅类

光栅图像的主要类是 `MapInfo.Raster.RasterInfo` 和 `RasterRead`。样式信息通过 `MapInfo.Style.RasterStyle` 处理。

`RasterInfo` 提供了与图像（以像素为单位）的高度和宽度有关的信息，还提供了光栅格式、颜色深度和注册信息。请参阅 `RasterInfo` 示例应用程序，该应用程序位于 `.\MapInfo\MapXtreme\7.x.x\Samples\Desktop\Features\RasterInfo`

`RasterRead` 是为要渲染的图像按顺序读取光栅图像和样式信息的类。

`RasterStyle` 与光栅的外观有关。您可以控制亮度和对比度，用灰度显示彩色图像，设置透明度和半透明度。`MapXtreme` 支持每个图像一种颜色透明。这意味着图像中该颜色所在的每个位置都是不可见的，且允许显示图像下边的图层。半透明度是整个图像上透明的程度。如果需要显示光栅图像之下的图层，则设置较高的半透明度值（100% 是透明）。

`Raster` 列是只读的，因此无法永久更改它们的样式。但是，可以通过编程方式设置并获取“图像”的属性，例如亮度、灰度以及半透明度。请参阅联机的开发人员参考中的 `RasterStyle` 类。

光栅图像和坐标系

将光栅、网格或 WMS 图像显示为地图图层时，MapXtreme 自动设置所有矢量地图图层的旋转和投影，以便使其与光栅图像的旋转和投影匹配。

如果地图包括多个光栅、网格或 WMS 图像图层，则 MapXtreme 在由可见性最好的光栅图像指定的投影中自动显示地图。如果具有不同投影的不同图像成为可见性最好的图像，则坐标系可以随地图视图的更改而更改（由于缩放或平移）。这种情况下，不可以更改地图的显示坐标系。

栅格重投影

通过光栅重投影，您可以更改光栅图层在地图中的制图投影。很多光栅图像类型都可以重投影。例如，卫星和航空照片图像、扫描的地图，以及网格和无缝光栅表。

也可以控制栅格和矢量图层的重投影。在向地图添加光栅或矢量图层时，新的图层将重投影到当前地图窗口投影中。

在更改同时包含矢量和栅格图层的地图窗口的投影时，所有图层（包括栅格和矢量）都将重投影到新的地图窗口投影中。

您可以在 MapXtreme API 中或在 Workspace 管理器的“图层控制”用户界面中访问光栅重投影设置。有关在 API 中使用光栅重投影的信息，请参阅《MapXtreme 开发人员参考》中的 **MapInfo.Mapping.RasterReprojectionMethod**。有关光栅重投影用户界面的信息，请参阅 [栅格重投影](#)。

工作空间中重投影的图像

可从工作空间中读取或写入光栅重投影属性。

MapXtreme 也可将来自 MapInfo Professional 的光栅重投影信息写入到 .mws 工作空间中。此功能允许您在 MapInfo Professional 中创建自己的 workspace，然后将其加载到 MapXtreme 中。

将工作空间载入 MapXtreme 之后，可以通过 API 或 Workspace 管理器的“图层控制”用户界面修改光栅重投影。

光栅图像的限制

- 不可以选择光栅图层的任何图元。
- 不可以搜索光栅图层的图元。

代码示例：将光栅图像添加到地图

将光栅图像添加到地图的方法和将任何其它图层添加到地图一样。

C# 示例：

```
Table MyTable = Session.Current.Catalog.OpenTable("MyRaster.tab");
FeatureLayer MyLayer = new FeatureLayer(MyTable);
MyMap.Layers.Add(MyLayer);
```

VB 示例：

```
Dim MyTable As Table = Session.Current.Catalog.OpenTable("MyRaster.tab")
Dim MyLayer As FeatureLayer = New FeatureLayer(MyTable)
MyMap.Layers.Add(MyLayer)
```

光栅句柄

MapXtreme 具有大量不同的库，可以使用其中一个库来加载光栅图像。通过 **MapXtreme** 加载光栅图像时，光栅图像搜索这些 DLL 并检查给定的文件是否可以通过该 DLL 读取。DLL/Raster 格式匹配完成之后，**MapXtreme** 就知道哪个 DLL 处理文件的格式。文件格式句柄是名为“xxxxxxxx.RHx”的 DLL。名称的基本部分取决于格式。扩展名始终以 RH 开头，但可以用任意字母 (A-Z) 结尾。搜索格式句柄时，**MapXtreme** 按字母顺序搜索格式，始于 RHA，止于 RHZ。该过程允许 **MapXtreme** 区分使用句柄的优先次序。例如，检查任何其它格式之前应该检查 SPOT 文件，因为这些文件只是原始数据，可能与其它格式混淆。SPOT 句柄的扩展名是 RHD。Halo 格式句柄称为 RHV。LEADTOOLS 格式句柄称为 RHX。

光栅句柄的默认文件夹路径如下：

\Program Files\ Common\ MapInfo\ MapXtreme\ 7.x.x\ RasterGridHandlers。

MapXtreme 包括了 LEAD Technologies, Inc. 提供的 LEADTOOLS Win32 Pro 和 Media Cybernetics 提供的 HALO Imaging 库。在 **MapXtreme** 中引用图像时，LEADTOOLS 光栅句柄会将整个光栅图像加载到内存中。这意味着图像的加载时间变长，而且如果由于内存需求极大导致图像极大的话，加载可能还会失败，但平移和缩放的速度会加快。

HALO 光栅句柄仅把需要显示的内容加载到内存中，因此这种方式加载图像的速度较快，但平移和缩放的速度较慢。根据字母顺序，HALO.RHV 在 LEADTOOL.RHX 前面，所以 HALO 句柄会先读取图像。要更改此顺序，请参阅[配置自定义光栅处理程序](#)。

下表显示了所提供的光栅句柄以及它们所支持的格式。

光栅句柄	光栅格式
SPOT.RHD	Spot (.bil)
ECW4.RHD	ECW (.ecw, .ers)
MRSID.RHE	MrSID (.sid)
ADRGASRP.RHL	ADRG, ASRP, USRP
CADRGCIB.RHL	CADRG, CIB, NITF
TIFF.RHL	TIF (.tif) [*]
VMGRID.RHL	Vertical Mapper (.grd, .grc) [†]

光栅句柄	光栅格式
HALO.RHV	BMP, TIF, GIF, TGA, JPEG, PCX
LEADTOOL.RHX	BMP, GIF, JPEG, JPEG2000, PNG, TIF, PSD, WMF, EMF

* 仅限支持 TIF 地理参考的句柄。如果在此 TIFF 句柄之前尝试使用过支持 TIFF 文件的其它句柄，则可能不再支持地理参考的 TIFF 文件。因此，要支持地理参考的 TIFF 文件，应先尝试使用此 TIFF 句柄，之后再使用支持 TIFF 的其它句柄。

† 可以显示为网格或光栅。.TAB 文件确定图像应使用网格句柄还是光栅句柄来绘制。

在多线程应用程序（例如 ASP.NET 应用程序）中有几种光栅格式不被支持。这些格式如下所示：

读取
Vertical Mapper GRD、GRC
Halo 的 JPEG (LeadTools JPEG 句柄是线程安全的)
Halo 中带 jpeg 压缩的 TIFF (LeadTools TIFF 句柄是线程安全的)
导出
LeadTools 的 JPEG 2000 (仅针对 Win2k)
LeadTools 的 TIFF CMYK

光栅句柄属性

MapXtreme 会基于句柄的文件扩展名中的最后一个字符按字母顺序调用光栅句柄。现在，您可以使 MapInfo.Data.TableRaster 类中的两个属性以编程方式控制顺序。

TableInfoRaster.PreferredHandler 属性可指定使用哪个光栅句柄来打开光栅图像。光栅引擎尝试使用指定的句柄打开光栅图像。如果无法打开，则会遵循正常流程，即按字母顺序调用句柄。如果成功打开，则会在新属性 **TableInfoRaster.ActualHandlerUsed** 中报告所使用的实际句柄。

注意，该值存储在表格的 ClientMetadata 中，且会从中读取该值。如果 .TAB 文件的介质数据中存在某个值，则会读入和识别该值。如果调用了 WriteTabFile，该值将在 .TAB 文件的 begin_metadata 部分中永久化存在。此属性等同于 TableInfo.ClientMetadata["\\PreferredRasterHandler"]。

配置自定义光栅处理程序

可以配置 MapXtreme 应用程序以使用其它光栅句柄（这些光栅句柄不同于 MapXtreme 分发中所包含的光栅句柄）或支持全新的光栅类型。还可以更改光栅句柄所用的优先级。

在 MapXtreme 的默认安装中，所有光栅句柄均放在以下位置：‘<program files>\Common Files\MapInfo\MapXtreme\7.x.x\RasterGridHandlers。这也是文件 mirasteru.dll 的位置。这是应用程序中所用的任何其它光栅句柄的推荐安装位置。若使用默认位置，则不需要任何其它的配置步骤。

若要将自定义光栅句柄放置到非默认位置，则必须在桌面应用程序的应用程序配置文件中或 web 应用程序的 web.config 文件中指定自定义光栅句柄的位置。要执行该操作，请在 <ApplicationDataPaths> 下面定义 <Path> 或 <SpecialPath> 元素，然后将所选光栅句柄复制到那个文件夹。

例如，要在非默认的 MyAppData 目录中配置桌面应用程序自定义光栅句柄，则可以使用下面的 .config 文件。

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
        <section name="MapInfo.CoreEngine"
            type="MapInfo.Engine.ConfigSectionHandler, MapInfo.CoreEngine,
            Version=6.8.0.536, Culture=neutral,
            PublicKeyToken=93e298a0f6b95eb1" />
    </configSections>
    <MapInfo.CoreEngine>
        <ApplicationDataPaths>
            <SpecialPath>
                <Personal>MyAppData</Personal>
            </SpecialPath>
            <Path>c:\MyAppData</Path>
        </ApplicationDataPaths>
    </MapInfo.CoreEngine>
</configuration>
```

在该示例中，<Personal> 标记是在用户的“我的文档”中定义的特殊位置。这将会引用“我的文档”中名为 MyAppData 的文件夹。在 <ApplicationDataPaths> 标记中，使用 <SpecialPath> 或 <Path> 标记。同时使用这两个标记意味着光栅句柄可能会放置在任一 "MyAppData" 文件夹中。虽然这种情况不能算是错误，但也可能不是您想要实现的操作。由于这是非默认的配置，还必须将 mirasteru.dll 放置到同一目录。

如果应用程序数据存储在相对于 .NET Framework 特殊系统文件夹的位置，则使用 <SpecialPath> 语法。例如，如果应用程序数据存储在“我的文档”目录下名为 MyAppData 的目录中，则配置文件中的条目可能为：

```
<Personal>MyAppData</Personal>
```

其中 "Personal" 是表示“我的文档”目录的 .NET Framework 枚举 Environment.SpecialFolder 的值。

该配置方法还可以用来更改可由多个光栅句柄管理的文件类型的首选光栅句柄。例如，JPEG 文件可由 Halo 或 LEADTOOLS（这两者都已捆绑在 MapXtreme 中）处理。通常，Halo 具有优先权，因为它的 *.rvh 文件扩展名按字母排序位于 LEADTOOLS *.rnx 扩展名的前面。但是，由于 MapXtreme 会先查看在配置文件中定义的任何 <ApplicationDataPaths>，因此，在默认 [CommonFiles] 文件夹中查找句柄之前，会先定位指定的光栅句柄。因此，例如，可能会将 LEADTOOLS *.RNX 句柄复制到定义的文件夹中并配置 MapXtreme 首先查找并使用那个句柄。

或者，可以重命名 [CommonFiles] 文件夹中的文件扩展名，以便所需的光栅句柄先出现在字母列表中。但是，这将会影响所有使用 MapXtreme 开发的应用程序并可能产生意外的影响。此外，若按此方式更改了文件扩展名，则在卸载 MapXtreme 时将不会删除这些重命名的光栅处理程序。出于上述这些原因，可能想要使用配置方法来更改光栅句柄的位置和优先级。

网格图像

网格图像显示了区域中数据值的内插值。网格图像从数据文件中创建，该数据文件中的数据以平均间距的点度量。整个地图区域转换为网格，其中每个网格单元表示一个值。尽管数据集合点需要均匀间隔，不过网格值不必以生成网格的内插值替换。MapXtreme 和 MapInfo Professional 创建网格时，需使用网格句柄来内插值。

除色阶外，网格图像还可以显示山脉或辅助底纹。辅助底纹允许网格表面根据虚拟光源加上底纹。每个网格单元的亮度与照射在表面上的光线一致，且亮度可以根据对光源的方向来调整。这非常适合于剖面网格地图，在这种地图中可以相对于光线的方向考虑表面倾斜和方向。最大亮度分配在太阳光线与表面垂直的点上。随着斜面转动远离光源，亮度值降低。

支持的网格格式包括：

- MapInfo Grid (*.mig)
- USGS DEM (*.dem)
- GTOPO30 (*.dem)
- DTED (*.dt0, *.dt1, *.dt2)
- Vertical Mapper Continuous Grids^{1 2}(*.grd, *.grc)

如果安装了定制网格句柄，则系统可以支持其它的网格格式。

网格类

网格图像的主类有 MapInfo.Raster.GridInfo、GridRead、GridCreatorFromFeatures、GridInflectionCalculator 和 HillShadeWrite。GridInfo 和 GridRead 类似于 RasterInfo 和 RasterRead，这种情况下可以获得与网格文件有关的信息。要针对数据点的图层创建网格，需使用

-
1. 可以显示为网格或光栅。.TAB 文件确定图像应使用网格句柄还是光栅句柄来绘制。
 2. 不是多线程安全的。

`GridCreatorFromFeatures`、`GridInflectionCalculator` 可用来设置变化值和变化色。`HillshadeWrite` 允许将辅助底纹添加到或改变为网格。网格的很多属性从 `RasterInfo` 继承，包括图像类、坐标系、光栅控制点和最小边界矩形 (MBR)。

代码示例：将网格图像添加到地图

将网格图像添加到地图的方法和将任何其它图层添加到地图一样。

C# 示例：

```
Table MyTable = Session.Current.Catalog.OpenTable("MyGrid.tab");
FeatureLayer MyLayer = new FeatureLayer(MyTable);
MyMap.Layers.Add(MyLayer);
```

VB 示例：

```
Dim MyTable As Table = _ Session.Current.Catalog.OpenTable("MyGrid.tab")
Dim MyLayer As FeatureLayer = New FeatureLayer(MyTable)
MyMap.Layers.Add(MyLayer)
```

代码示例：从网格地图检索数据

以下示例说明了如何打开并读取网格文件的信息。

VB 示例：

```
Public Shared Sub MapInfo_Raster_GridRead(ByVal strGridFilename _
As String)
    Dim strHillshadeFilename As String = _
        MapInfo.Raster.GridRead.DefaultHillshadeFilename(strGridFilename)
    Dim session As ISession

    session = MapInfo.Engine.Session.Current
    Dim gridread As GridRead = New GridRead(strGridFilename, _
        strHillshadeFilename)
    Console.WriteLine(gridread)
End Sub
```

以下示例说明了如何打开并读取网格文件的单元格值。

VB 示例：

```
Public Shared Sub MapInfo_Raster_GridReadStartRead(ByVal _
gridread As GridRead, ByVal strGridFilename As String)
    Dim strHillshadeFilename As String = _
        gridread.DefaultHillshadeFilename(strGridFilename)

    If gridread.StartRead() Then
        Dim x As Integer = 0      ' TODO - set to a pixel column value
        Dim y As Integer = 0      ' TODO - set to a pixel row value
        Dim bIsNull As Boolean
        Dim dValue As Double
```

```
If gridread.GetValue(x, y, bIsNull, dValue) Then
    If bIsNull Then
        ' read a null cell
        Console.WriteLine("{0,20}", "NULL")
    Else
        ' read a non-null cell, with value == dValue
        Console.WriteLine("{0,20}", dValue)
    End If
End If
gridread.EndRead()
End If
End Sub
```

创建网格

MapXtreme 提供使用可写网格句柄和内插器以创建连续网格的功能。如要以编程方式创建这些网格，请使用 **MapInfo.Raster.GridCreatorFromFeatures** 类、**Mig.ghl** 网格句柄和提供的两个内插器中的任何一个。网格可根据数据点的表或根据选择创建。

在 **MapXtreme** 中创建的网格可以和 **MapInfo Professional v 10.0** 兼容。

连续网格是一种划分为矩形网格单元格的地图，其中每个单元格都会包含一个数据值，它表示测量的数据点或基于周围数据点的内插值。连续网格通过在地图上使用连续色阶，以显示数据值的变化情况。

在以前，**MapXtreme** 的功能只限于读取网格和获取其网格单元格值的列表。它也可以读取有关网格的信息，例如山体阴影和样式。**ContinuousGridWrite** 类虽然可用，但没有使用内插器的功能，所以创建的网格无法更好地反映其数据点。

MapInfo.Raster.GridCreatorFromFeatures 是创建连续网格时可以调用的主类。它使用 **MapInfo** 可写网格句柄和内插器，以生成连续的网格地图。

内插器 **IDW** 和 **TIN** 根据其特定的公式，提供确定这些网格单元值的算法。**IDW** 最适合用于人口数据，而 **TIN** 则最适合用于地形数据。请参阅**反距离加权 (IDW) 内插器**和**不规则三角网络 (TIN) 内插器**。

如果提供的任何一个内插器无法满足您的要求，您可从新的 **IInterpolator** 接口派生一个，以创建自己的内插器。请参阅《开发人员参考》中的 **IInterpolator 接口**和 **MapInfo.Raster.Interpolators** 命名空间。

如果数据包含的重复数据点在同一个单元格位置中，网格 API 将提供合计类，以便将这些点相加或进行平均操作。其他可用的合计器方法有 **count**、**min** 和 **max**。也可以创建或使用执行 **IGridCellAggregator** 的任何合计器。

其他 Grid API 的增强功能还包括根据地图边界轮廓剪切网格。创建网格时，需参照源数据的最小边界矩形。如需根据地图边界轮廓设置网格边界，请将 **GridCreatorFromFeatures.ClippingGeometry** 属性设置为 **Geometry**，使其为网格单元格的剪切形状。

从数据点创建网格后，可以立即使用新的“网格样式”对话框和“Workspace 管理器”修改特征，例如山体阴影、样式和变化点。有关详细信息，请参阅[网格样式](#)。

网格内插器

MapXtreme 提供两种用来创建连续网格的网格内插器：**IDW** 和 **TIN**。**IDW** 内插器和 **TIN** 内插器位于单独的命名空间中，该空间名为 **MapInfo.Raster.Interpolators**。

此外，**MapXtreme** 还提供创建您自己的内插器的接口。

反距离加权 (IDW) 内插器

MapInfo.Raster.Interpolators.InverseDistanceWeighted 内插器类是两种带有 **MapXtreme** 的网格内插器中的一种。**IDW** 内插器最适合用于数据值，例如人口或在网格上产生任意值的任何数据，它不适合与邻近值相关的值或受邻近值影响的值。这种内插方法也很适合稀疏数据。

IDW 内插器计算覆盖网格的单元格中所包含的值。计算中的每个数据点值都会被其从单元格中心的距离进行加权。由于内插是一种反距离加权计算方法，所以点距离单元格越远，其值对所得单元格值的影响就越小。

在 **IDW** 中，指数决定每个点对得数的影响程度。指数越高，距离越近的点对单元格值的影响就越大。指数的范围为 1 至 10。

也可以针对同一网格单元格中的源数据点值选择合计方法。可选择：**average**、**count**、**sum**、**min** 或 **max**。

有关使用 **IDW** 内插器创建网格的代码示例，请参阅《开发人员参考》中的 **MapInfo.Raster.GridCreatorFromFeatures** 类。

不规则三角网络 (TIN) 内插器

第二种内插器称为“不规则三角网络”或 **TIN**。**TIN** 最适合用于地形数据和具有线性进程或在网格上有相互关系的数据点，例如温度。

TIN 内插器可从网点绘制三角形，它重绘的原始地图地形比 **IDW** 内插器重绘的更细致。它可在点之间绘制线条，将它们分割成三角形并尽量连接所有的点。它创建一个连通网，使网格点能够插入。此内插方法不受相邻原始数据值的影响，因此您不会发生数据的“错误碰撞”，而使用 **IDW** 内插器则可能发生这种情况。

TIN 参数（对象属性）可以更改，以显示地图地形上更多或更少的细节。这些属性包括 **Tolerance**（控制是否丢弃距离过近的点）、**Distance**（控制输出）和 **FeatureAngle**（控制网格边缘的锐度）。

有关更多信息，请参阅《开发人员参考》中的 **MapInfo.Raster.Interpolators.TriangulatedIrregularNetwork** 类。

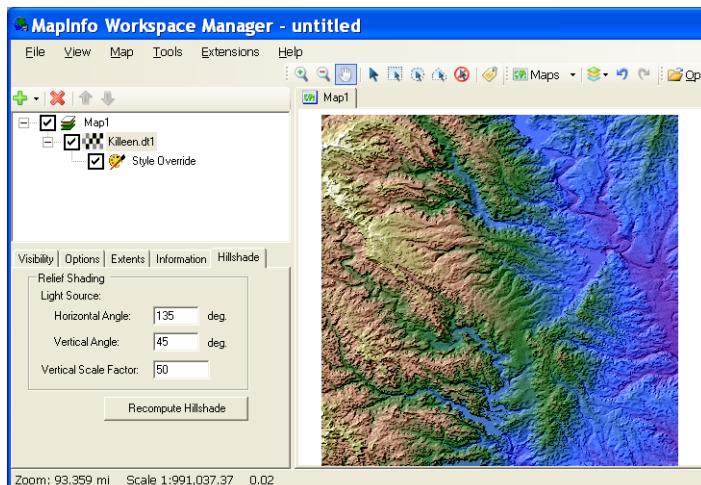
IInterpolator 接口

我们还支持您通过 IInterpolator 接口创建您自己的内插器。从该接口派生您自己的内插器，即可根据您的算法创建网格单元格值。请参阅《开发人员参考》中的 MapInfo.Raster.IInterpolator 接口。

网格样式

MapXtreme 提供针对连续网格图像修改变化值和颜色的功能。旧版本的 MapXtreme 只能从网格读取信息。

网格样式支持通过 GridStyleControl> MapInfo.Styles.GridStyle 类以编程方式提供；该样式支持可以集成到 Workspace 管理器中。有一种名为 GridForm 的示例应用程序也可提供此功能。变化值和颜色更改后会保存到工作空间，以便以后加载。



网格图像和变化

连续的网格图像可显示区域上数据值的内插值。网格由网格单元格构成，其中每个单元格都表示一个值。这些值在地图上的表现形式为受数据值范围限制的连续色阶。

由于网格值变化或百分比变化导致颜色改变的点被称为“拐点”。MapXtreme 创建的 GridInflectionCalculator 对象可根据所支持的任何一种方法计算拐点值。将网格的拐点数量与数据值的最小范围和最大范围相结合，可以用来确定地图上的色阶。表示变化的颜色可以作为 FeatureOverrideStyleModifier 应用于网格。

变化方法

MapXtreme 支持以下计算方法：

- 单元格数量相等 (*Equal Cell Count*)—计算变化值，使数量相等的单元格值处于所计算的比例范围内。
- 值范围相等 (*Equal Value Ranges*)—计算变化值，使每个变化值的范围都大小相同。
- 自定义单元格数量 (*Custom Cell Count*)— 和“单元格数量相同”类似，但它使用的是自定义百分比范围。
- 自定义值范围 (*Custom Value Ranges*)— 和“值范围相等”类似，但它使用的是自定义变化值。

变化值和变化颜色可以通过 `MapInfo.Raster.GridInflectionCalculator` 类使用编程方式修改。

`GridStyleControl` 位于 Visual Studio 的 `MapXtreme` 工具箱中。

该过程也可集成到 `Workspace` 管理器的“网格样式”对话框中。请参阅[网格样式](#)。



变化方法设置为“值范围相等”或“单元格数量相等”时，如要更改值或百分比，变化方法则将分别变为“自定义值范围”或“自定义单元格数量”。

变化值和颜色更改后会保存到工作空间，以便以后加载。

计算网格图层的变化值和颜色

计算变化值和颜色时可通过编程方式，也可以通过 `Workspace` 管理器中的“网格样式”对话框。`GridForm` 示例应用程序也提供此功能。

GridInflectionCalculator 类

参阅以下代码示例，可了解如何创建初始的值集和颜色集并修改它们。此代码示例位于 `MapInfo.Raster.GridInflectionCalculator` 类下面的《开发人员参考》API 文档中。

```
public static void MapInfo_Raster_GridInflectionCalculator(TableInfoGrid
tableInfoGrid)
{
    // Create 10 grid inflection values, by using the EqualRangeValues
    method, using colors from orange to red.
    GridInflectionCalculator gridInflectionCalculator = new
    GridInflectionCalculator(tableInfoGrid,
    InflectionMethod.EqualRangeValues, 10, Color.Orange, Color.Red);

    // Modify the colors to go from green to brown.
    gridInflectionCalculator.CalculateStyles(Color.Green, Color.Brown);

    // Calculate 5 grid inflection values, by using the EqualRangeValues
    method.

    gridInflectionCalculator.CalculateValues(InflectionMethod.EqualRangeValue
s, 5);
}
```

辅助底纹

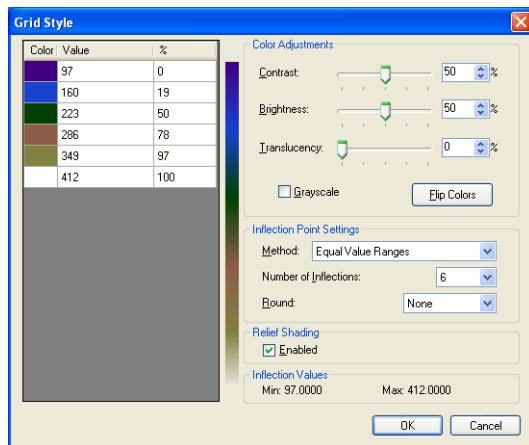
除色阶外，网格图像还可以显示山脉或辅助底纹。辅助底纹允许网格表面根据虚拟光源加上底纹。每个网格单元的亮度与照射在表面上的光线一致，且亮度可以根据对光源的方向来调整。这非常适合于剖面网格地图，在这种地图中可以相对于光线的方向考虑表面倾斜和方向。最大亮度分配在太阳光线与表面垂直的点上。随着斜面转动远离光源，亮度值降低。

可通过“网格样式”对话框中的复选框启用山体阴影。光源的水平和垂直角度以及比例因子可在“图层控制”中设置。可使用编程方式通过 `MapInfo.Raster.HillshadeWrite` 类将山体阴影添加到网格。

“网格样式”对话框

在“网格样式”对话框中，可以通过图形用户接口 (GUI) 修改网格的样式设置，其中包括颜色、变化值、对比度、亮度、半透明度等样式的设置。样式可作为 `FeatureOverrideStyleModifier` 应用于网格图层。

要访问“网格样式”对话框，请打开 `Workspace` 工作空间中的网格图像。选择“网格图层”，然后右键单击“添加样式覆盖”菜单项。在“可视性”选项卡中，单击“网格图像样式”按钮以显示对话框。有关“网格样式”对话框中的每项组件详情，请参见以下内容。



Color/Value/% (Percentage)

“Color/Value/% (Percent)” 网格框显示变化颜色、值和百分比的当前设置。

要更改颜色，请双击颜色色板并从“颜色”对话调色板中选择新颜色。

值和 % 的设置来源于网格中的数据。在此对话框中修改的“变化点设置”组的设置时，这些值会随之更改。这会使变化值重新计算并显示新值。例如，选择“自定义值范围”时可以直接编辑值；而选择“自定义单元格数量”时则可以直接编辑百分比。

- ①** 变化方法设置为“值范围相等”或“单元格数量相等”时，如要编辑值或百分比，变化方法则将分别变为“自定义值范围”或“自定义单元格数量”。

颜色调整

可进行调整，以更改网格图像中的颜色。这些更改会均匀地应用于整个图层。

对比度

使用“对比度”滚动条可调整图像中的对比度。在 0 和 100% 之间滑动滚动条，以设置网格的对比度。可以使用光标键进行微调。

亮度

使用“亮度”滚动条可调整图像中的亮度。在 0 和 100% 之间滑动滚动条，以设置网格的亮度。可以使用光标键进行微调。

半透明

使用“半透明度”滚动条可调整图像的半透明度。半透明度的设置范围为 0 至 100%。半透明度为 0% 的图像是完全不透明的（无法透视）。半透明度为 100% 的图像是完全透明的（完全不可见）。

灰度

勾选以仅显示网格中的灰度颜色。

“替换颜色”按钮

单击此按钮可使变化颜色反向。此时只会影响起点颜色和终点颜色。

变化点设置

变化点设置可用来设置计算变化点的方法，以显示数据在区域上的分布方式。“单元格数量相等”和“自定义单元格数量”基于百分比。“值范围相等”和“自定义值范围”基于值。

单元格数量相等

设置变化值，使数量大致相等的网格单元格处于计算的百分比范围内。

值范围相等

在源数据的最小范围和最大范围之间分布变化值，使每个变化值范围都相等。

自定义单元格数量

如不想使用均匀分布的百分比，而要使用自己的百分比，可选择此方法。

自定义值范围

如不想使用均匀分布的值，而要使用自己的值，可选择此方法。

变化值的数量：

从列表中选择 2 至 16 个变化值，或者也可键入 2 至 255 之间的任意数字。

舍入：

从列表中选择要应用于变化值的舍入因子。如果分布方法基于单元格数量，那么只有在真正计算变化值时才能看到这种舍入的效果。

辅助底纹

辅助底纹可根据虚拟光源的方向给网格表面加阴影。这样就可以根据光的方向考虑表面斜度和方向。“网格样式”对话框提供启用或禁用辅助底纹的设置。

要调整辅助底纹的设置，首先必须突出显示“图层控制”中的网格图层，然后选择“山体阴影”选项卡。“辅助底纹”不可作为样式覆盖进行应用，因为它只是一种变化设置。

光源

光源设置在“图层控制”对话框中的“山体阴影”下面，它可以设置水平角、垂直角和垂直比例。该选项卡仅在选定网格图层或启用山体阴影时才会显示。

水平角

旋转水平平面中的光源。零度表示东边有光源闪耀。正值角逆时针方向旋转光源，所以，90 度会将光源放在北边。

垂直角

旋转垂直平面中的光源。零度会将光源放于水平面，而 90 度则会将光源直接架空。用 180 至 360 度之间的数字指定角度，以指定光源位于平面之下。

垂直比例因子

用 0 至 100 之间的数字指定比例因子。增加此比例因子会使表面垂直增大，以增强阴影效果。需要根据平面增强细节时，此功能很有用。

重新计算山体阴影

在调整光源设置后，单击此按钮可重新计算山体阴影。

变化值

这些值显示网格表中的最小值和最大值。这些值不可编辑。

Min:

显示源表中最小的数据点的值。

Max:

显示源表中最大的数据点的值。

GridInfoForm 示例应用程序

MapXtreme 提供可集成 GridInflectionCalculator 的 C# 示例应用程序。这些示例位于 \\Samples\\VisualStudio20xx 文件夹下，具体路径如下：\\Desktop\\Features\\GridinfoForm。

使用图片服务器地图

MapXtreme 支持显示由 MapInfo Professional 创建的图片服务器 TAB 文件。允许您从 MapXtreme 桌面或 Web 应用程序打开和查看地图瓦块，例如 Bing 地图和 Google。

在本章中：

◆ 图片服务器图像	316
◆ 瓦块缓存.....	316
◆ 图片服务器图层的地图行为	316
◆ 使用图片服务器图像.....	317
◆ 图片服务器设置	318
◆ 使用 <code>TableInfoTileServer</code> 类	319
◆ 图片服务器示例应用程序	320

图片服务器图像

瓦块是地图的组成部分，存储在图片服务器中，可根据需要作为图像传输到客户端应用程序进行显示。MapXtreme 以支持栅格或 WMS 图像的同样方式来打开和显示瓦块图像。图像可以作为单独的图层或者 .MWS 工作空间的组成部分加载到 MapXtreme 中。作为图层打开的瓦块添加为地图控件的最底层图层。

来自图片服务器的图像可为地图生成出色的背景。例如，反映真实世界详细信息（例如建筑物、厂房、植被）的航空照片就非常适合做地图的基本图层。在 MapXtreme web 或桌面应用程序中将图片服务器图像用作基本图层。重叠诸如街道网络、表示客户的点位置和邮政边界等矢量数据来创建有用且在视觉上吸引人的地图。

来自 MapInfo Professional 的图片服务器文件可构成 TAB 文件以及包含指向该图片服务器的 URL 的相关 XML 文件。TAB 文件提供了坐标系信息，MapXtreme 需要用来精确显示图层并且与您很可能在地图中包含的矢量数据相符合。

MapXtreme 支持两类图片服务器：QuadKey 和 LevelRowColumn。请参阅[第 317 页中的使用图片服务器图像](#)。

MapXtreme 拥有针对瓦块的内置缓存支持，可以更快速地访问瓦块图像。在打开桌面或 Web 应用程序的会话过程中，缓存保持不变。请参阅[第 316 页中的瓦块缓存](#)。

您还可以对图片服务器图层应用样式覆盖，以控制图像的透明度、亮度、对比度和灰度，与处理栅格图层的方式相同。请参阅[第 301 页中的光栅类](#)。

瓦块缓存

MapXtreme 可将引用的图片服务器的瓦块缓存到磁盘中并执行渲染。如果需要在地图控件中绘制瓦块，通常是由于缩放和平移，以及在缓存中不可用，它们是从服务器中请求并且添加到缓存中的。在关闭 MapXtreme 桌面或 Web 应用程序之前，图片服务器缓存将一直保留在磁盘中。

图片服务器缓存默认存储在 TEMP 或 TMP 目录下的 MapXtreme 文件夹内。

图片服务器图层的地图行为

图片服务器图层的坐标系在 TAB 文件中定义，且在渲染时不可更改。在将图片服务器图像显示为地图基本图层时，MapXtreme 会强制以图片服务器图层的同样投影方式绘制已经可见的图层（矢量、栅格、WMS 等）。地图的坐标系也会更改以适合图片服务器图层。如果在地图中存在其它栅格图层，则直到图片服务器图层变为可见，才允许栅格重投影。要启用其它栅格图层的栅格重投影，必须删除可见的图片服务器图层。

如果地图包括多个图片服务器图层，则 MapXtreme 在由可见性最好的图片服务器图像指定的投影中自动显示地图。新添加的图片服务器图层将添加到现有图片服务器图层的上面。

为了获得更佳的可见性和顺畅的文本渲染，不论添加任何图片服务器图层，地图的抗锯齿和半透明属性都默认设置为 true。这些设置会一直存在，直到您手动禁用这些属性。

使用图片服务器图像

图片服务器图像是向地图中加载更多信息的非常常用方法。由于仅返回覆盖处于当前缩放级别和中心的地图视图的瓦块，所以它们比 WMS 更高效。

MapXtreme 支持两类 Tile 服务器：QuadKey 和 LevelRowColumn。

QuadKey

QuadKey 是 Bing Maps Tile System 用来唯一标识其特定级别的瓦块的一种机制。来自 Bing 的图像在 URL 中包含 QuadKey 属性。要访问 Bing 地图，必须从 Microsoft 获取 Bing Key。请参阅 <http://msdn.microsoft.com/en-us/library/ff428642.aspx>。有关如何设置该密钥的详细信息，请参阅第 319 页中的使用 TableInfoTileServer 类。

MapXtreme 还支持已经本地化的 Bing 地图。

MapXtreme 为以下三种类型的 Bing 地图提供 TAB /XML 文件示例：航空图、公路图和混合图。

LevelRowColumn

LevelRowColumn 是唯一标识瓦块的另一种方式。Google 地图以及其它地图使用此种机制。级别、行和列都作为 URL 的一部分给出。

MapXtreme 通过由 MapInfo Professional 创建的图片服务器 TAB 文件支持下列图片服务器类型：

- Spectrum Spatial
- ArcGIS
- OSM
- MapQuest

向图片服务器验证

根据 TAB 文件所引用的图片服务器（Quadkey 或 LevelRowColumn），MapXtreme 可能需要向图片服务器提供凭据。对于如 Bing 瓦块地图（航空图、公路图和混合图）的 QuadKey 类型图片服务器，需要 Bing 密钥来设置身份验证。Bing 密钥可以指定基于 MapXtreme 应用程序，如第 318 页中的图片服务器设置中所述。

如果 TAB 文件引用的任何图片服务器需要用户凭据（用户名和密码），那么对于桌面应用程序，将向用户显示身份验证对话框。

图片服务器设置

在向地图控件中加载图片服务器 TAB 或工作空间之前，必须在系统级别或者通过 `MapInfo.Engine.TileServerSettings` 类以编程方式配置一些设置。将为整个 `MapXtreme` 应用程序实例设置图片服务器设置。

Bing 地图的许可密钥

使用注册表键变量 "SOFTWARE\MapInfo\MapXtreme\7.3.0\TileServerKey" 来设置工作空间中的 TileServer TAB 文件所需的图片服务器密钥。

使用系统环境路径变量 `MI_TILE_SERVER_KEY` 来设置工作空间中的 TileServer TAB 文件所需的图片服务器密钥。

可以从 Microsoft 获取 Bing 图片服务器的试用密钥。有关详细信息，请参阅 <http://msdn.microsoft.com/en-us/library/ff428642.aspx>。

通过 Web 或桌面配置文件

对于 `MapXtreme` web 应用程序和/或独立应用程序，您可以在 `Web.Config` 和/或 `app.config` 文件中设置图片服务器设置，如下所示：

```
<add key="MapInfo.Engine.Session.Workspace.TileServerSettings"
      value="ReadTimeOut:60; RequestTimeOut:60;
      TileServerKey:XXXXXXXXXXXXXXXXXXXX" />
```

通过 `MapInfo.Engine.TileServerSettings` 类

`MapXtreme` 的 API 提供了 `MapInfo.Engine.TileServerSettings` 类来设置图片服务器的设置（请参阅第 319 页中的使用 `TableInfoTileServer` 类）。默认情况下，`MapXtreme` 有如下图片服务器设置值：

<code>MapInfo.Engine.TileServerSettings.ReadTimeOut</code>	60 seconds
<code>MapInfo.Engine.TileServerSettings.RequestTimeOut</code>	60 seconds
<code>MapInfo.Engine.TileServerSettings.Key</code>	"" empty string



注：这些设置会覆盖已有设置并且在被更改之前一直使用。

TileServerSettings 类的示例代码

`MapInfo.Engine.TileServerSettings` 类可用于设置如下代码片段的值

C# 示例：

```
public static void MapInfoSetTileServerSettings()
{
    MapInfo.Engine.TileServerSettings tssetting = new
    TileServerSettings();
    tssetting.Key = "";
    tssetting.ReadTimeout = 70;
    tssetting.RequestTimeout = 70;
    Session.Current.TileServerSettings = tssetting;
    //Table and/or workspace opening code, for example, here.
}
```

VB 示例：

```
Public Shared Sub MapInfoSetTileServerSettings()
    Dim tssetting As MapInfo.Engine.TileServerSettings = New
    TileServerSettings()
    tssetting.Key = ""
    tssetting.ReadTimeout = 70
    tssetting.RequestTimeout = 70
    Session.Current.TileServerSettings = tssetting
    'Table and/or workspace opening code, for example, here.
End Sub
```

使用 **TableInfoTileServer** 类

可以使用 **MapInfo.Data.TableInfoTileServer** 类来打开图片服务器 tab 文件。

C# 示例：

```
public static void MapInfoDataTableInfoTileServer(Map map)
{
    TableInfo tableInfo = TableInfo.CreateFromFile("BingAerial.tab");
    TableInfoTileServer tableInfoTileServer = tableInfo as
    TableInfoTileServer;
    tableInfoTileServer.PreferredHandler = "leadtool.rhx";
    Table table =
    Session.Current.Catalog.OpenTable(tableInfoTileServer);
    try {
        map.Load(new MapTableLoader(table));
    }catch (Exception ex){
        String str = ex.ToString();
        // user code
    }
}
```

VB 示例：

```
Public Shared Sub MapInfoDataTableInfoTileServer(ByVal map As Map)
    Dim tableInfo As TableInfo =
    tableInfo.CreateFromFile("BingAerial.tab")
    Dim tableInfoTileServer As TableInfoTileServer = CType(tableInfo,
    TableInfoTileServer)
    tableInfoTileServer.PreferredHandler = "leadtool.rhx"
    Dim table As Table =
    Session.Current.Catalog.OpenTable(tableInfoTileServer)
    Try
        map.Load(New MapTableLoader(table))
    Catch e As Exception
        Console.WriteLine("Exception caught : {0}", e)
    End Try
    'user code
End Sub
```

图片服务器示例应用程序

图片服务器 C# 桌面示例应用程序可以帮助您更深入地了解如何使用瓦块功能。它位于 MapXtreme 安装目录中，对于 32 位和 64 位安装分别如下所述：

C:\Program Files\MapInfo\MapXtreme\7.3.0\Samples\VisualStudio2012\Desktop\Features\TileServer
C:\Program Files (x86)\MapInfo\MapXtreme\7.3.0\Samples\VisualStudio2012\Desktop\Features\TileServer
C:\Program Files\MapInfo\MapXtreme\7.3.0\Samples\VisualStudio2013\Desktop\Features\TileServer
C:\Program Files (x86)\MapInfo\MapXtreme\7.3.0\Samples\VisualStudio2013\Desktop\Features\TileServer

地理编码

本章介绍用于进行地理编码的 **MapXtreme** 命名空间，并对访问地理编码服务器或服务的应用程序的编写进行说明并提供相关示例。

在本章中：

- ◆ **MapInfo.Geocoding** 命名空间概述 322
- ◆ 主要的地理编码类 322
- ◆ 了解地理编码模型 324
- ◆ 对位置进行地理编码 326
- ◆ 使用约束条件以实现精确地理编码 329
- ◆ 了解近似匹配项的精确度 332

MapInfo.Geocoding 命名空间概述

MapInfo.Geocoding 命名空间提供用于对地址记录进行地理编码的接口和类。地理编码是指确定街道地址、十字路口、场所或邮政编码对应的地理位置以便在地图上精确定位相关位置的过程。地理编码客户端通过 HTTP 向 Pitney Bowes Software 的地理编码服务器或地理编码 Web 服务产品发送请求。购买 **MapXtreme** 并不附送服务器或 Web 服务。

比如说，您正在构建一个查找最近位置的应用程序，用户以地址的形式向该应用程序提供他们的地理位置，之后会接收到标明他们所在位置以及最近的 ATM 机的地图。为了实现这一功能，仅有地址不足以在地图上显示位置。这些地理位置（地址、建筑物、相关地点）会转换为地球表面坐标。您需要经度/纬度值或坐标值来精确定位并在地图上确定位置。**MapXtreme** 地理编码客户端可以通过场所名称、街道地址、邮政编码或两个街道之间的十字路口来确定上述这些值并提供给您。

除了返回坐标值之外，地理编码客户端可以返回特定位置的完整描述，如果对于该位置不是完全了解，这些信息会很有帮助。根据请求参数选择、所使用的方法、指定的参数选择以及匹配条件，地理编码客户端可能会针对请求返回零个、一个或多个响应。

MapXtreme 地理编码客户端也具有可对不完整地址进行地理编码的 **Gazetteer** 类型功能（世界/国家/城市地理编码组件）。例如，仅包含城市、州/省、位置名称、路标或机场的地址。

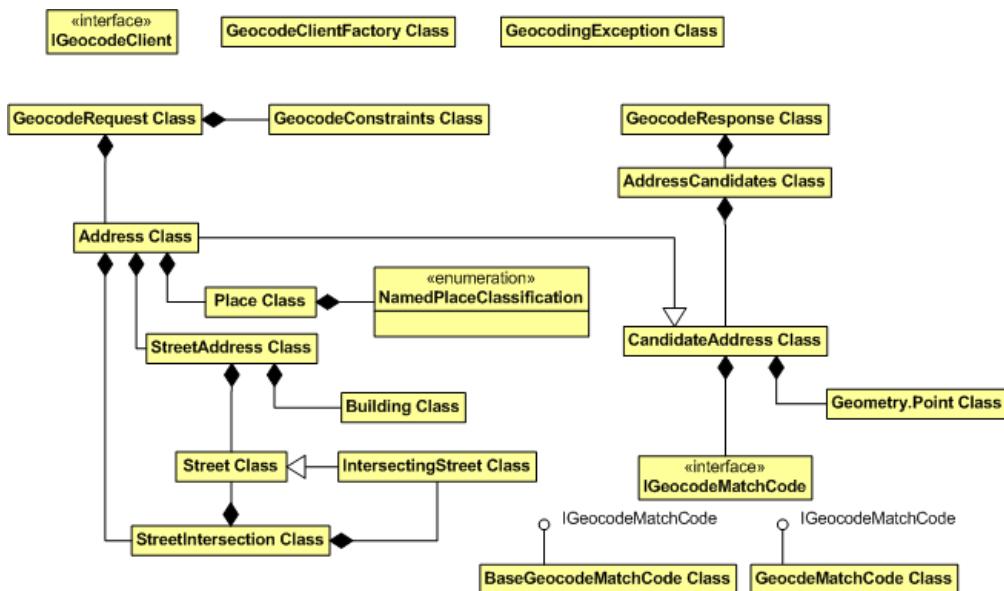
使用地理编码功能，可以通过将输入地址与具有坐标值的位置相匹配来获取这些坐标值。您可以在单个请求文本中提供一个或多个地址。它也可以处理不完整的地址。服务器会返回完整的地址，您可以使用它来代替您的输入地址。根据数据质量和请求参数选择，服务器会针对服务请求返回零个、一个或多个响应。返回的响应是一个列表，通常会包括一个或多个可能的匹配项、调用的候选项以及指示候选项的定位精确度的结果代码。然后，您的应用程序需要通过选择合适的地址候选项来解析响应结果，从而获取坐标值。

主要的地理编码类

地理编码功能是通过 **MapInfo** 地理编码服务器或 **Envinsa Location Utility** 服务来支持实现的。其中一个资源需要部署，并使其可通过 URL 获得。要创建合适的地理编码客户端，需要使用 **GeocodeClientFactory** 类。这两种客户端的接口是类似地，因为它们对于地理编码请求、参数选择和响应都使用相同的类。

地理编码客户端的主接口称为 **MapInfo.Geocoding.IGeocodeClient**。此接口定义地理编码客户端对象。它只有一个方法 **Geocode**，该方法以请求作为输入参数，返回包含候选地址（输入地址的可能匹配项）的响应。其他类包括：**GeocodeRequest**、**GeocodeResponse**、**GeocodeClientFactory**、**GeocodingConstraints**、**AddressCandidates**、**GeocodeMatchCode** 和 **CandidateAddress**。

下图说明了构成 **Geocoding** 命名空间的接口和类：



GeocodeRequest

`GeocodeRequest` 类向地理编码服务器或服务发送请求。它的属性包括 `AddressList`（要进行地理编码处理的地址列表）和 `Length`（地址列表中的项目数）。输入地址可以包括各种信息，比如街道地址或十字路口、主要邮政编码和辅助邮政编码以及国家/地区代码，等等。

GeocodeResponse

`GeocodeResponse` 类包含对 `GeocodeRequest` 中的每个地址的响应。响应中会包含地址候选项，即那些被认为可能与输入地址匹配的地址。请注意，如果 `GeocodeRequest` 包含多个地址，则所产生的 `GeocodeResponse` 对象中，每一个输入地址都对应有一个候选地址列表。

GeocodeClientFactory

此类返回一个 `IGeocodeClient`，您可以使用它来发送 `GeocodeRequest` 和接收 `GeocodeResponse`。方法 `GetMmjHttpClient` 使用地理编码客户端，后者与 `MapInfo MapMarker Java servlet` 的实例进行通信。方法 `GetEnvinsaGeocodeClient` 用于向 `MapInfo Envinsa 4.0` 中的 `Location Utility Service` 发送请求。

GeocodingConstraints

此类包含可在进行地理编码时设置的参数选择。所有的参数选择都要进行设置，并且所有参数选择会获取返回 `true` 或 `false`（默认值）的属性。

AddressCandidates

此类是一个包含 GeocodeResponse 中返回的可能的地址匹配项的列表。

BaseGeocodeMatchCode 和 GeocodeMatchCode

这些是实现 `IGeocodeMatchCode` 接口的类。该接口会公开一个 `ResultCode` 属性，此属性是一个描述匹配良好程度的字符串。将请求发送到 Envinsa Location Utility Service 后，会返回 `BaseGeocodeMatchCode`。`GeocodeMatchCode` 是对 `BaseGeocodeMatchCode` 的扩展。在将地理编码请求发送到 MapMarker 服务器后会返回 `GeocodeMatchCode`。`GeocodeMatchCode` 包括其他一些有用的属性，可确定地址的哪些部分被匹配。在这些属性中，如果候选项与街道匹配，则 `StreetAddressMatch` 返回 `true` 值，如果候选项与自治区匹配，则 `MunicipalityMatch` 返回 `true` 值。有关结果代码的详细信息，请参阅[了解近似匹配项的精确度](#)。

CandidateAddress

这个类定义进行地理编码后的地址。它的属性包括 `Address`（地理编码地址所对应的地址）、`GeocodeMatchCode`（说明地址的地理编码匹配程度）和 `Point`（代表候选地址的几何对象）。

了解地理编码模型

地理编码客户端的基础是一个相对匹配模型，相对匹配由一组权重来控制，而权重分值通过对每一个地址部分与数据中的候选记录（可能的匹配）进行比较来获得。将得出的分值进行汇总，然后使用候选项的总得分来确定一项或多项最佳匹配。如果有一个候选项的得分高于任何其他候选项，即获得了一个近似匹配。此外，匹配例程使用一组地理编码参数选择来确定特定匹配条件是必要的还是宽松的。例如，默认参数选择对邮政编码是否匹配不作要求，而要求住宅号码和街道名称必须匹配。这样返回的命中率效果最好，错误匹配最少（误报），性能最佳。

有关地理编码的一些权衡考虑

对于如地理编码客户端这类相对匹配系统，在如何使用地理编码数据方面有几个必须考虑的权衡关系。请考虑如下一些问题：

- 您需要的匹配精确度（唯一地址匹配、近似匹配）？
- 对于您的地理编码点（街道级别、邮政编码中心位置）所需的地理精确度？
- 您的目标是要对尽可能多的记录进行地理编码吗？

对于这些问题的回答取决于您想要如何使用地理编码记录。例如，您可能要确定新的零售店的位置，需要知道当前和潜在顾客的分布。在这个例子中，您需要对尽可能多的顾客进行地理编码，而不需要对每位顾客找到完全匹配的街道地址。对邮政编码中心位置进行地理编码更适合于您的分析。

另一方面，如果您是一名公共设施服务机构的协调人员，您需要知道您的客户相对于附近天然气管线的位置，每个客户的定位精确度都非常重要。使用严格的匹配参数选择进行街道级别的地理编码是您的最佳策略。

有关地址的说明

地址数据的质量在地理编码中是最重要。与不完整的或格式不符合要求的地址相比，一个遵循标准的场所地址约定的规范的输入地址会产生更好的结果。

本节概括介绍输入地址和匹配的参考地址的地址质量。要充分利用地理编码，重要的是了解您的数据以及要匹配的数据的地址结构。

输入地址

输入地址是指那些您希望对其进行地理编码的地址。这些地址不包含地理坐标，因此无法在地图上定位。

输入地址由几部分组成，地理编码服务器会检查这些部分以确定匹配项。输入地址中可以包含地址号码、街道名称以及街道前缀和后缀部分。前缀通常标识方向，比如“北”或“南”，并且只有某些地址存在前缀。后缀通常是街道类型、街道、道路、巷道。输入街道中可能还会出现公寓或道路号码等其他信息。

输入地址可以包含完整的街道信息或不完整的街道信息，也可以包含不同的拼写形式。例如，如果输入的地址是 LaSalle Street，则会匹配 LaSalle St。然而，如果 LaSalle 不带街道后缀，则可能无法匹配，因为地址也可能是 LaSalle Ave。

MapXtreme 地理编码客户端可以使用所谓的“不规范”数据和不完整的地址。此客户端同时包含美国和美国以外的地址，只要服务器包含相应的参考地址。

参考地址

参考地址是指服务器用来与输入地址进行匹配的地址。这些记录包含在地图上定位地址所需要的地理坐标。参考地址保留在服务器上，并且会定期通过维护程序进行更新，以处理新的地址。它们按照区域名称约定进行了标准化。例如，在北美，街道号码在街道名称之前。而在世界其他地方，街道号码有可能是在街道名称之后。

通常，参考地址包含基本组件：地址号码、街道名称和前缀（如果有的话）及后缀。对于非美国地区参考地址中可能不会包含公寓号码之类的具体信息，因此，会忽略输入地址中的这些内容。

什么是自定义用户字典？

可能会出现的情况是，服务器或服务所提供的 **MapInfo** 地址字典不会像期望的那样包含您感兴趣的区域或地区。此外，用户可能需要拥有他们愿意使用的自定义数据（例如，有关国内所有公司专营店的数据）。为了达到这些目的，用户可以使用他们自己的包含此类数据的字典。

地理编码客户端支持使用自定义字典。自定义字典是一个包含街道和地址范围的表，通过对照该表来匹配未进行地理编码的记录。在进行地理编码时可以根据需要使用任何数量的自定义字典。有关创建自定义用户字典的详细信息，请参阅服务器或服务的相关文档。

什么是 World Geocoding？

地理编码客户端允许通过在输入地址中指定国家/地区代码来按不同的国家进行地理编码。**Geocoding World** 组件与基于 **MapMarker** 的组件不同，后者每个组件提供一个国家/地区或区域。**Geocoding World** 组件提供 238 个国家或区域的城市和邮政编码数据以及 9 个国家的街道一级的数据。请注意，对于这两个选件，访问数据都需要经过授权，需要拥有有效的数据许可文件。有关 **Geocoding World** 的信息，请联系您的 **Pitney Bowes Software** 销售代表。

对位置进行地理编码

位置的地理编码可以帮助对您的数据关系进行可视化处理。例如，地址记录一经进行地理编码，路径规划客户端就可以利用并显示这些记录来指示两个地址（位置）之间的驾驶方向。一旦数据有了地理位置方面的参考信息，就可以执行空间搜索来回答如“找到此位置 10 英里范围内的所有顾客”这类问题。所有地理编码请求都可以：

- 对不完整的地址进行地理编码，并返回一整套地址信息（规范化地址）。
- 指示对地理编码请求中提供的特定地址的响应中所包括的完全匹配项和近似匹配项的数量。
- 在一个地理编码请求中可以处理一个或多个地址。
- 使用匹配代码提供有关匹配结果的匹配质量的信息。

使用 **GeocodeRequest** 类可以向地理编码服务器或服务发送请求。它的属性包括 **AddressList**（要进行地理编码处理的地址列表）、**Length**（地址列表中的项目数）和 **GeocodeConstraints**（有关详细信息，请参阅[使用约束条件以实现精确地理编码](#)）。可以在输入地址中包括各种信息，例如街道地址、十字路口、主要邮政编码和辅助邮政编码以及国家/地区代码。输入地址中包括的信息级别将确定地理编码结果候选项的级别和精确度。

使用地理编码客户端，您可以执行不同级别的地理编码操作：

街道地址地理编码

可以使用地理编码街道地址信息在地图上显示街道地址或者执行空间搜索和查询。例如，地理编码信息可用于在地图上显示商店位置以及属于商店忠诚度计划范围内的顾客，以此确定市场区域。对于街道地址地理编码，使用 **Address** 类为输入地址指定可用信息。

如何对地址进行地理编码？

以下示例说明如何利用地理编码客户端使用 C# 对街道地址进行地理编码。地理编码请求有以下三部分：定义参数、创建街道和地址对象以及创建地理编码请求。

另请参见 ..\MapInfo\MapXtreme\7.x.x\Samples\Desktop\Features\Geocode\cs 路径下提供的 Geocode 应用程序示例。

定义参数

要填充地址对象，必须定义地址参数。除了地址参数之外，还必须定义服务器或服务的 URL。

```
//Define the server URL
String MMJUrl = "";

//Define the address parameters
String streetName = "One Global View";
String cityName = "Troy";
String stateName = "NY";
String zipCode = "12180";
String countryCode = "USA";
//The following are optional address parameters
String directionalPrefix = "";
String typePrefix = "";
String typeSuffix = "";
String directionalSuffix = "";
```

创建地址

通过创建 Street、StreetAddress 和 Address 对象来填充地址。

```
//Create a Street object
MapInfo.Geocoding.Street street =
    new MapInfo.Geocoding.Street(
        directionalPrefix,
        typePrefix,
        streetName,
        typeSuffix,
        directionalSuffix);

//Create a StreetAddress object
MapInfo.Geocoding.StreetAddress streetAddress =
    new MapInfo.Geocoding.StreetAddress(street);

//Create an Address object
MapInfo.Geocoding.Address address =
    new MapInfo.Geocoding.Address(streetAddress, countryCode);
address.PrimaryPostalCode = zipCode;
address.PlaceList =
    new MapInfo.Geocoding.Place[]
    {new MapInfo.Geocoding.Place(
        cityName,
        MapInfo.Geocoding.NamedPlaceClassification.Municipality),
     new MapInfo.Geocoding.Place(
        stateName,
        MapInfo.Geocoding.NamedPlaceClassification.CountrySubdivision)};
```

创建地理编码请求

一旦创建地址对象，就可以将地址添加到一个列表并使用该列表来创建地理编码请求。使用 **GeocodeClientFactory** 可以定义是将请求发送到地理编码服务器还是发送到服务。对于要发送到地理编码服务器的请求，使用 **GetMmjHttpClient** 方法；对于要发送到地理编码服务的请求，则使用 **GetEnvinsaLocationUtilityService** 方法。

```
//Add the address to a list of addresses
MapInfo.Geocoding.Address[] addressList = {address};

//Create the geocode request
MapInfo.Geocoding.GeocodeRequest geoRequest =
    new MapInfo.Geocoding.GeocodeRequest(addressList);

//Create the geocoding client
MapInfo.Geocoding.IGeocodeClient geoClient =
    MapInfo.Geocoding.GeocodeClientFactory.GetMmjHttpClient(MMJUrl);

//Send the request and get the response
MapInfo.Geocoding.GeocodeResponse geoResponse =
    geoClient.Geocode(geoRequest);
```

街道十字路口地理编码

街道十字路口可以由不同的地址类型或地址元素构成，并且请求可以具有不同级别的精确度或参数选择。例如，移动用户可以使用街道十字路口地理编码功能输入最近的街道十字路口，以在移动设备上查看他们所在位置的地图。对于街道十字路口地理编码，需要使用 **StreetIntersection** 类并指定 **Street** 和 **IntersectingStreet** 属性。

邮政编码的地理编码

邮政编码中心位置代表邮政编码对应区域的中心。利用这种在地图上定位中心位置的功能可以执行人口统计分析来找到市场营销区域，以便有针对性地进行广告宣传或直接通过邮件发送广告。对于邮政编码级别的地理编码，使用 **Address** 类仅指定 **PrimaryPostalCode** 信息即可。

Gazetteer 类型地理编码

地理编码客户端的功能之一就是，可以执行世界、国家/地区或城市范围内的地理编码操作，以针对仅包括城市、州/省或地点名称的不完整地址找到所对应的位置。支持以下操作：

- 基于国家/地区、国内地区、城市、市内区域、路标或机场进行搜索以返回位置。
- 模式搜索功能支持在一个或多个字段中使用通配符 (*) 作为搜索值来返回特定字段的所有可用值。

批量地理编码

对批量地理编码功能的支持是通过 `IGeocodeClient.BatchRequests` 属性来实现的。此属性设置在一次操作中发送到服务器的地址数量。默认的地址数目为 25。`BatchRequest` 属性必须大于 0 且小于 500。向 `IGeocodeClient.Geocode()` 方法提交地址时，可以超出 `BatchRequest` 属性中指定的数目。如果出现这种情况，则会将包含这些地址的地理编码操作细分为多个操作，每个操作所包含的地址数量不超过在 `BatchRequest` 属性中指定的数值。

i 如果 `BatchRequest` 的数值较低（例如 2），则会降低性能。如果 `BatchRequest` 数值较高（例如 500），则可能会引发内存问题。批量地理编码的执行性能取决于系统配置。

使用约束条件以实现精确地理编码

地理编码客户端允许您在对输入地址进行地理编码时使用 `GeocodeConstraints` 类来设置匹配限制、后备设置以及多重匹配设置。在此可以根据需要设置严格或宽松的参数选择。在匹配率、精确度和性能之间达到最佳平衡效果的设置具备以下特点：

- 对于住宅号码和街道名称要达到完全匹配。
- 不转而依赖邮政编码中心位置来定位。
- 只返回近似匹配项。

最严格的匹配条件要求对住宅号码、街道名称、邮政编码进行完全的匹配，并且不依赖于邮政编码中心位置。实际上，服务器或服务在您的输入地址中指定的邮政编码范围内查找街道地址的完全匹配项。

什么是匹配约束条件？

匹配约束条件控制如何匹配给定的地址、为每个地址返回的匹配项或候选项数量以及返回哪些辅助信息。输入地址可能会由于提供的信息不完整或不正确而无法精确匹配任何实际的地址。

`GeocodeConstraints` 类包含一系列属性，可确定地址的哪些部分最为重要，以及在无法找到完全匹配项的情况下如何操作。

`IGeocodeClient.DefaultGeocodeConstraints` 属性可用于从服务器或服务（如果使用）检索默认约束条件。如果没有提供 `GeocodeConstraints`，地理编码客户端将使用从服务器或服务检索到的默认约束条件。

下面的任何一个参数选择都会影响地理编码服务器或服务所使用的记录匹配级别：

- **CloseMatchesOnly** - 只返回那些属于近似匹配候选项的地理编码结果。地址候选项按输入地址与 `MustMatchXXX` 参数选择的近似匹配程度进行分级。只有那些标记为近似匹配项 (`true`) 的候选项才会被返回。默认为 `false`。
- **MustMatchAddressNumber** - 只将那些与地址号码匹配的候选项视为近似匹配。默认为 `true`。

- **MustMatchMainAddress** - 只将那些与街道名称匹配的候选项视为近似匹配。这并不会妨碍考虑那些通过“探索方式”获得的匹配项。（例如，对于“Muller Strasse”，仍旧会认为它是“Mueller Strasse”的近似匹配。）不过，如果输入的街道没有生成近似匹配，它确实会阻止对街道名称进行扩展性处理（拼写错误），如并列字母的情况（例如，“Muleler Strasse”之于“Mueller Strasse”）。默认为 true。
- **MustMatchCountry** - 只将那些匹配国家/地区的候选项视为近似匹配。默认为 true。建议不要更改此参数选择。
- **MustMatchCountrySubdivision** - 只将那些匹配国内地区的候选项视为近似匹配。默认为 true。CountrySubdivision 在各个国家可能会代表不同的区域。例如，在美国，它表示州，而在加拿大，它表示省。
- **MustMatchCountrySecondarySubdivision** - 只将那些匹配二级国家地区的候选项视为近似匹配。默认为 false。CountrySecondarySubdivision 在各个国家代表不同的区域。例如，在美国，它表示郡，而在加拿大，它表示按人口统计划分的地区。
- **MustMatchMunicipality** - 只将那些匹配独立行政区的候选项视为近似匹配。默认为 true。比如镇或城市都属于独立行政区。
- **MustMatchMunicipalitySubdivision** - 只将那些匹配独立行政区下级地区的候选项视为近似匹配。默认为 false。比如 MSA（城市统计区）或城市的自治行政区（如纽约的布朗克斯或布鲁克林）都属于 MunicipalitySubdivision。
- **MustMatchPostalCode** - 只将那些匹配邮政编码的候选项视为近似匹配。默认为 false。
- **MustMatchInput** - 只将那些匹配所有输入参数选择的候选项视为近似匹配，而不管特定参数选择如何设置。默认为 false。
- **FallbackToGeographic** - 确定如果对于街道地理编码请求没有找到候选项，是否在地理区域的中心位置执行地理编码。生成的位置点位于区域的地理中心位置，在该区域中可能获得最高级别的精确度。默认值设置为 false。
- **FallbackToPostal** - 确定如果对于街道地理编码请求没有找到近似候选项，是否在邮政编码中心位置执行地理编码。默认值设置为 false。
- **OffsetFromStreet** - 确定到地址所定位到的街区的距离。该值用来防止将分布在一个街道上的不同地址定位为同样的地点。例如，如果某个街区的住宅号码范围为 40 到 60，那么 50 号住宅就会确定在该街区的中心点位置。OffsetFromStreet 垂直于道路来定位地点位置，这样，街道左侧的住宅会出现在左侧，街道右侧的住宅会出现在右侧。OffsetFromStreet 的默认值是 25 米。OffsetFromStreet 使用 Distance 类来指定，需要定义该值和度量单位。
- **OffsetFromCorner** - 确定到地址所定位到的街区的终点（街道拐角）的距离。该值用来防止将位于街道拐角的地址与十字路口按照同一个位置点给出。OffsetFromCorner 会使用沿街区与（地址所在）街道平行的方向上的一段距离值来定位街道拐角处的点。例如，如果某个街区的住宅号码范围为 40 到 60，那么 50 号住宅就不会定位在街道十字路口处。OffsetFromCorner 的默认值是 25 米。OffsetFromCorner 使用 Distance 类来指定，需要定义该值和度量单位。
- **MaxCandidates** - 定义响应中返回的候选项的最大数量。所返回的实际候选项数目可能会小于这个最大值。值 -1 用来表示返回所有满足其他约束条件的候选项。使用 MapInfo 地理编码服务器时的默认值为 3。使用 Envinsa Location Utility 服务时的默认值为 -1（所有）。

- **MaxRanges** - 此参数选择控制地理编码操作返回确切的街道地址号码还是返回一定数量的号码。如果将 **MaxRanges** 设置为 0，则不会返回一定数量范围的号码，而只返回确切的地址号码。如果将 **MaxRanges** 设置为大于 0 的值，则会返回该数量的街道地址号码。**MaxRanges** 的默认值是 0。例如，如果 **MaxRanges** 设置为 5，则该服务至多返回 5 个位于该地址范围内的匹配项。如果值为 -1，则返回所有合适的匹配项。
- **MaxRangeUnits** - 定义每个候选项范围所返回的候选项范围单位的最大数。**MaxRangeUnits** 的默认值为零 (0)。
- **GeocodeType** - 确定在可通过服务器使用地理编码时的地理编码类型。有关所支持的地理编码类型，请参阅服务器或服务文档。
- **LocalGeocodeConstraints** - 允许设置对特定国家/地区的约束条件，这些条件专门用于特定国家和该地理编码组件。这些值始终为 **Key/Value** 对的形式。有关可用的本地约束条件的信息，请参阅服务器或服务文档。
- **DictionaryUsage** - 指定是仅使用服务器地址词典、仅使用配置的用户字典还是同时使用二者。若要同时使用这两种字典，则需要指定优先使用哪个字典。默认情况下，会使用 **AD_AND_UD** 选项。此选项指示应使用服务器地址字典和任何已配置的用户字典。有关创建和访问用户字典的信息，请参阅服务器或服务文档。
- **UseCASSRules** - 指定服务器是否使用 CASS 级别地理编码（仅适用于美国用户）。CASS 认证是将邮寄地址表标准化以满足 U.S.Postal Service® (USPS) 大宗邮件折扣要求的过程。地理编码客户端在对记录进行地理编码时严格依照 U.S.Postal Service 设定的匹配条件来执行地址标准化。必须使用支持 CASS 的美国地理编码数据组件版本才能使用 CASS 地理编码功能。请联系您的 **MapInfo** 销售代表来获取最新版本，以遵循 CASS 地址标准化要求。默认为 **false**。

发送匹配约束条件的影响

放松约束条件可以扩大匹配项的搜索区域。例如，通过放松邮政编码条件，可以在输入地址所指定的邮政编码范围之外但在所指定的城市范围之内来搜索候选项。在确定地理编码设置时请考虑以下事项：

放松邮政编码条件

如果放松了邮政编码条件，则会在更宽泛的区域内搜索匹配项。虽然这样会降低性能，但匹配率会更高，因为在针对请求比较匹配候选项时不需要完全匹配。

放松下级地区和独立行政区条件

如果放松了这些条件，则会对与特定邮政编码匹配的街道地址进行搜索，搜索时会考虑与指定的城市或下级地区名称不匹配但与邮政编码匹配的其他城市或下级地区。

放松街道名称

在执行搜索时，会考虑名称与输入地址类似或名称由于拼写不规范而存在差异的所有候选项。这会降低性能。但同时，由于可以对更多的候选项进行检查，因此会提供匹配率。如果您的数据表进行了索引，则会减小由性能和匹配率而产生的时间差别。

放松住宅号码条件

放松住宅号码设置对性能不会有很大的影响。但如果候选地址所对应的街区不包含任何范围，则确实会影响到匹配类型。在候选项的住宅号码范围不包含输入的住宅号码时，匹配类型也可能会受到影响。如果放松了住宅号码条件，建议将返回的最大范围设置为大于 0 的值。

了解近似匹配项的精确度

对结果的排名在请求的近似匹配项基础上进行。对于 **GeocodeMatchCode** 类，对于按照请求中的 **GeocodeConstraints** 以及服务器或服务配置一同确定的约束条件来尝试匹配的每一个记录，地理编码服务器或服务会返回一个结果代码。结果代码表示地理编码操作是成功还是失败，并包含反映匹配质量的信息。代码分为 5 个主要类别：

- 单个近似匹配（**S** 类别）
- 来自多个候选项的最佳匹配（**M** 类别）
- 邮政编码中心位置匹配（**Z** 类别）
- 地理中心匹配（**G** 类别）
- 无匹配代码

对于 **S** 或 **M** 类别结果代码，代码的每个字符表示地理编码操作对每个地址部分的匹配准确程度。

类别	描述	示例
H	住宅号码	115
P	街道的方向前缀	北
无	街道名称	主要
T	街道类型	地点
S	街道的方向后缀	SE
C	城市名称	纽约
Z	邮政编码	80302
A	地址字典	A

下面是一个近似匹配候选项的示例：

S5HPNTSCZA

下面是一个非近似匹配候选项的示例：

S5-PNTSC-A

结果代码中的破折号表明存在冲突。如果请求中的邮政编码与地理编码操作在数据中找到的邮政编码冲突（输入为 28031，而找到的却是 28013），则结果字符串中的 Z 字符由一个破折号来代替（例如 S5HPNTSC-A）。

单个近似匹配（S 类别）

S 类别的匹配表示记录仅与一个地址候选项匹配。第一个字符 (S) 表示地理编码服务器找到了一个与记录匹配的街道地址。代码中的第二个字符表示针对进行地理编码处理的记录所获得的位置点的定位精确度。

- S1 - 单个近似匹配，位置点在邮政编码对应的中心位置。
- S2 - 单个近似匹配，位置点在 ZIP+2 中心位置（美国中心）
- S3 - 单个近似匹配，位置点在 ZIP+4 中心位置（美国中心）
- S4 - 位置点在形状点路径（形状点定义街道的折线形状）的中心
- S5 - 位置点在街道地址所在位置
- S6 - 位置点在邮政编码对应的几何范围的中心。（例如，大建筑物有自己的代码）。这代表可达到的最高精确度。
- SX - 位置点在街道十字路口
- SO - 无可用坐标（极少出现）。

来自多个候选项的最佳匹配（M 类别）

M 类别的匹配表示记录有多个近似匹配候选项，服务器或服务已经从中选出了最佳匹配项。与 S 类别一样，M 类别代码中的第二个位置代表所获得的位置点对象的定位精确度。

- M1 - 多个近似匹配，位置点在邮政编码对应的中心位置。
- M2 - 多个近似匹配，位置点在 ZIP+2 中心位置（美国中心）
- M3 - 多个近似匹配，位置点在 ZIP+4 中心位置（美国中心）
- M4 - 多个近似匹配，位置点在形状点路径（形状点定义街道的折线形状）的中心
- M5 - 多个近似匹配，位置点在街道地址所在的位置（可获得的最高精确度）
- M6 - 多个近似匹配，位置点在邮政编码对应的点位置。
- MX - 多个近似匹配，位置点在街道十字路口
- M0 - 多个近似匹配，没有可用的候选项

邮政编码中心位置匹配（Z 类别）

Z 类别中的匹配表示没有建立任何街道匹配条件，因为：1) 没有近似匹配且您允许服务器或服务落在邮政编码中心位置；2) 地址为邮政信箱或乡村地址；或 3) 您将服务器或服务设置为匹配邮政编码中心位置。生成的位置点在邮政编码对应的中心位置，可能有以下 4 种精确度。

- Z1 - 邮政编码中心匹配
- Z2 - ZIP+2 中心匹配（美国中心）

- Z3 - ZIP+4 中心匹配（以美国为中心，是可获得的最高精确度）
- Z6 - 对于位置点 ZIP 的邮政编码中心匹配
- Z0 - 邮政编码匹配，没有可用坐标（很少出现）

i 位置点 ZIP 是指不对应于区域的 ZIP 代码。这包括 邮政信箱和唯一 ZIP（单个场所、建筑物或组织）。在使用非美国地址字典时，此类别的结果仍旧表示为 Z。

地理中心匹配（G 类别）

G 类别匹配表示没有进行街道匹配。如果没有找到近似匹配，并且服务器或服务配置为转而使用地理中心位置，就会出现这种情况。生成的位置点在地理中心位置，可能的精确度如下：

- G0 - areaName0 中心位置（国家/地区）
- G1 - areaName1 中心位置（国内地区）
- G2 - areaName2 中心位置（国内二级地区）
- G3 - areaName3 中心位置（独立行政区）
- G4 - areaName4 中心位置（独立行政区下级地区）

无匹配代码

下面的结果代码表示没有找到任何匹配：

N - 无近似匹配。这些记录可以使用 MapInfo 地理编码产品或随后在按不同的匹配条件自动通过匹配的过程中重新进行地理编码。

NX - 对于街道十字路口没有近似匹配。

ND - 服务器或服务无法针对给定的邮政编码或独立行政区 / 国内地区找到地址字典。一旦找到地址字典，这些记录也可以重新进行地理编码。

NG - 用户在地理编码期间将这些记录标注为不可进行地理编码。服务器或服务不会再次尝试匹配这些记录，直到将该代码移除。

路径规划

本章介绍了 **MapXtreme** 路径规划命名空间并提供编写应用程序以访问路径规划服务器或服务的说明和示例。

在本章中：

- ◆ **MapInfo.Routing** 命名空间概述 336
- ◆ 路径计算 337
- ◆ 高级路径选项 340
- ◆ **Iso** 路径规划（驾驶时间和驾驶距离） 343
- ◆ 使用路径规划数据更新请求 347

MapInfo.Routing 命名空间概述

MapInfo.Routing 命名空间提供了一个接口和多个类，用于将 **MapXtreme** 路径规划客户端集成到应用程序中，其方法与地理编码客户端相似。路径规划客户端通过 HTTP 向 **Pitney Bowes Software** 的路径规划服务器或路径规划 Web 服务产品发送请求。购买 **MapXtreme** 并不附送服务器或 Web 服务。

使用路径规划功能，可以确定两点之间的最快或最短路线（路径），并可以添加各种高级路径规划选项。此功能可以返回路径几何体（用于地图显示的路径布局）、逐圈方向以及总距离和总时间路径摘要。另外，还可以创建包括一些中间点和参数选择的路径，以避免特定的路况，如高速公路和某些特定的地点。

路径规划功能包括重要的分析工具，用于计算驾驶时间和驾驶距离（**isoChrono** 和 **isoDistance**），并且可以使用瞬态更新来避免某些道路类型或重新计算路径。路径规划客户端的功能可分为以下四个逻辑组：

- 路径计算
- 高级路径选项
- Iso 路径规划（驾驶时间和驾驶距离）
- 使用路径规划数据更新请求

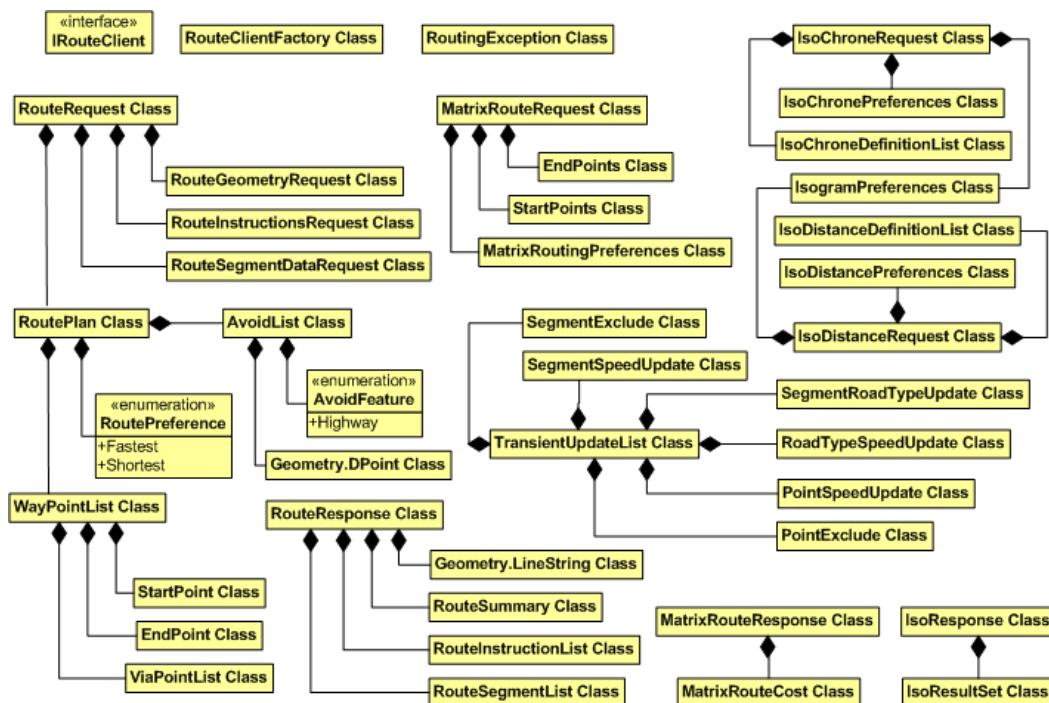
主要的路径规划类

Routing 支持使用 **MapInfo** 路径规划服务器或 **Envinsa Route** 服务。其中一个资源需要部署，并使其可通过 URL 获得。要创建适当的路径规划客户端，请使用 **RouteClientFactory** 类。两个客户端的接口是类似地，因为它们对于路径规划请求、参数选择和响应都使用相同的类。

使用路径规划客户端，可以确定两个位置之间的最短或最远路线（路径），增加中间点位置（**ViaPoint** 类）、避免位置或道路类型（**AvoidList**、**PointExclude**、**SegmentExclude** 类），获得驾驶导航（**RouteInstructionsRequest** 类）、获得路径几何体（**RouteGeometryRequest** 类）以及使用在更多有意义的分析中所生成的众多参数选择来自定义路径。路径规划客户端包括重要的分析工具，允许计算驾驶时间和驾驶距离（**isoChrono** 和 **isoDistance**），并且可以使用瞬态更新来避免某些道路类型或重新计算路径。路径规划客户端的功能可分为以下三个逻辑组：

1. 计算路径，**RouteRequest** 类用于计算点到点和多个点之间的路径，而 **MatrixRouteRequest** 类用于计算矩阵路径。
2. 计算驾驶时间和驾驶距离，**IsoChronoRequest** 类用于计算驾驶时间，而 **IsoDistanceRequest** 类用于计算驾驶距离。
3. 使用瞬态更新来计算路径，通过使用 **TransientUpdate** 类以生成更加精确和有意义的路径。

下图的示意图阐述了组成 Routing 命名空间的接口和类：



路径计算

可以执行三种类型的路径计算（确定最短或最快路径）：简单路径（点到点）、多点路径和矩阵路径请求。

点到点路径规划

最常见的路径规划要求之一是简单的驾驶导航，即用户输入两个地址、两点或任意组合，就可以获得一条路径。这条路径根据指定的起点和指定的终点计算获得。

在点到点路径规划的最简单级别，响应信息包括一个路径摘要。这个摘要包括路径的距离和所需要的时间。可以为点到点路径添加大量的选项，以返回其他的相关信息。可以添加：

- 路径规划参数选择
- 驾驶导航
- 路径几何体
- 避免点、图元和分段
- 基于时间的路径规划
- 使用路径规划数据更新请求

使用 **RouteRequest** 类创建点到点路径。有关 API 级的语法和说明，请参阅《开发人员参考》。

下面的代码示例是一个简单的点到点路径示例：

```
Public Shared Sub New_RouteRequest()
    ' Create the start point and end point
    Dim coordSys As MapInfo.Geometry.CoordSys = _
        Session.Current.CoordSysFactory.CreateFromPrjString("1, 104")
    Dim dpt1 As MapInfo.Geometry.DPoint = _
        New MapInfo.Geometry.DPoint(-74, 42)
    Dim dpt2 As MapInfo.Geometry.DPoint = _
        New MapInfo.Geometry.DPoint(-74, 41)
    Dim startPoint As MapInfo.Geometry.Point = _
        New MapInfo.Geometry.Point(coordSys, dpt1)
    Dim endPoint As MapInfo.Geometry.Point = _
        New MapInfo.Geometry.Point(coordSys, dpt2)
    ' Create the point list
    Dim pointList As WayPointList = _
        New WayPointList(startPoint, endPoint)
    ' Create the route plan object
    Dim plan As RoutePlan = _
        New RoutePlan(pointList)
    ' Create the request object
    Dim request As RouteRequest = _
        New RouteRequest(plan, DistanceUnit.Mile)
End Sub
```

多点路径规划

多点路径规划是指通过多个点进行路径规划的能力。路径规划服务器将查找通过所有点的最短时间或距离。这可以用于复杂的导航。对于想要规划每日走访或顾客拜访（如配送和修理公司）或一般现场销售的客户，他们需要这些导航功能。一般来说，应定义一个起点和终点并加入一系列的停止点，然后进行计算。除了起点和终端点之间有经过点之外，多点路径与点到点路径没有任何区别。

与点到点路径相似，也可以为多点路径添加大量的选项，以返回其他的相关信息。可以添加：

- 路径规划参数选择
- 驾驶导航
- 路径几何体
- 避免点、图元和分段
- 基于时间的路径规划

使用 **RouteRequest** 类创建多点路径。要想为一个路径添加多个中间点，**WayPointList** 必须至少包含一个 **ViaPoint**。有关 API 级的语法和说明，请参阅《开发人员参考》。

下面的代码示例介绍了如何创建由一个 **startPoint**、**endPoint** 和 **ViaPoint** 组成的 **WayPointList**：

```
Public Shared Sub New_WayPointList()
    ' Create a start point and end point
    Dim coordSys As MapInfo.Geometry.CoordSys = _
        Session.Current.CoordSysFactory.CreateFromPrjString("1, 104")
```

```
Dim dpt1 As MapInfo.Geometry.DPoint = _
    New MapInfo.Geometry.DPoint(-74, 42)
Dim dpt2 As MapInfo.Geometry.DPoint = _
    New MapInfo.Geometry.DPoint(-73, 42)
Dim dpt3 As MapInfo.Geometry.DPoint = _
    New MapInfo.Geometry.DPoint(-73.562, 42)

Dim startPoint As MapInfo.Geometry.Point = _
    New MapInfo.Geometry.Point(coordSys, dpt1)
Dim endPoint As MapInfo.Geometry.Point = _
    New MapInfo.Geometry.Point(coordSys, dpt2)
Dim intermediatePoint As MapInfo.Geometry.Point = _
    New MapInfo.Geometry.Point(coordSys, dpt3)

' Create the intermediate point object
Dim point As ViaPoint = New ViaPoint(intermediatePoint)
' Create the collection
Dim pointList As ViaPointList = New ViaPointList
' Append the object to the collection.
pointList.Append(point)
' Create the WayPointList object
Dim list As WayPointList = _
    New WayPointList(startPoint, endPoint, pointList)
End Sub
```

矩阵路径规划

矩阵路径规划允许查找大量起点和大量终点之间的最短或最远路径，并返回路径代价。这些代价是各个独立路径的总时间和距离。如果有三个起点和 50 个目标点，要想查找所有起点与所有目标点之间的最短路径，并且返回路径的代价，这时矩阵路径功能特别有用。

响应反馈的点顺序和序列数量由请求中输入的起点和终端的数量确定。例如，如果在请求中指定了两个起点和两个终点，响应信息将包含以下序列（S 是起点，E 是终点）：S1 - E1、S1 - E2、S2 - E1、S2 - E2。

一般来说，此计算功能可以用于确定一些特定服务（如消防或警局）的服务响应时间和覆盖范围。客户可能需要这些计算功能，以确保他们能够提供符合服务级别协议的完全覆盖能力，例如，配备一名或更多的人员，以便在接到通知的 20 分钟内即可响应突发事件。在矩阵路径规划的最简单级别，响应信息将包括一个序列列表。每个序列都包括起点和终端，以及距离和时间。可以为矩阵路径添加大量的选项，以返回其他的相关信息。可以添加：

- 路径规划参数选择
- 避免点、图元和分段
- 使用路径规划数据更新请求

使用 **MatrixRouteRequest** 类创建矩阵路径。矩阵路径必须包括 **MatrixRouteCost**，而且可能包括 **MatrixRoutingPreferences**。有关 API 级的语法和说明，请参阅《开发人员参考》。

高级路径选项

可以为路径规划应用程序添加大量的高级功能。本节介绍这些用于自定义点到点、多点以及矩阵路径请求的重要功能。

路径规划参数选择

可以为想要计算的每个路径类型指定各种参数选择。所有这些参数选择都是可选的，但是它们对于确定路径的准确性和帮助创建更有意义的路径有着十分重要的作用。可以返回带有从起点到终点最快时间或从起点到终点最短距离的路径。如果指定，这些参数选择将覆盖在服务器或服务配置中定义的路径优化设置。如果服务器参数选择的限制作用更强，参数选择可能被服务器参数选择覆盖。

路径计算的类型（点到点、多点或矩阵）决定哪些参数选择可用。下面的列表列出了可用的路径参数选择：

- **DistanceUnit** - 可以为所有的路径计算指定距离单位。最常用的单位包括千米、米、英里、码和英尺。若未指定值，将会使用默认的设置（英里）。
- **RouteMethod** - 确定要执行的路径参数选择的常规类型。路径客户端可以执行最快或最短路径计算。若未指定值，将会使用默认的设置（最快）。
- **OptimizeIntermediatePoints** (仅用于多点路径) - 确定是否要按在请求中指定的顺序遍历经过点或通过使用经过点来使路径客户端查找最佳路径。默认情况下 (`false`)，路径客户端将按请求中的顺序遍历指定的点。当设置为 `true` 时，则使用经过点计算最佳路径。
- **StopThreshold** (仅用于多点路径) - 设置路径计算算法停止阈值。当算法中剩余路径候选之间的时间或距离差距达到停止阈值时，该路径算法将会停止计算并返回当前“最佳”的路径。有效值是任意正数。为了获得最佳的性能，建议使用一个小于 1 的正数。设置停止阈值必须在准确度与速度之间取得平衡。阈值比平均值越低，结果的准确度越高，但是路径计算所需要的时间越长。对于经过点很少（小于 10）的路径，停止阈值的影响很小。默认为 0.01 (1%)。
- **TimeOut** - 设置路径计算算法超时值。超时值的单位为秒。默认时间为 600 秒。当达到超时值后，该路径算法将会停止计算。
- **TimeUnit** - 设置所有路径持续的时间单位。默认时间单位为分钟。
- **TravelPreferences** - 支持指定独特的路径规划功能，以修改旅行的道路类型优先级。要添加旅行参数选择，可以指定道路类型和道路类型参数选择。通过将 `RoadTypePreference` 设置为“高”、“中”、“低”或“避免”，可以根据优先级执行路径规划。进行路径计算时，将优先选择那些高优先级的道路类型。还可以使用 `AvoidRoadTypePreference` 避免特定的道路类型。一种更为常用的旅行参数选择方法是将 `FERRY RoadType` 设置为“低”，从而仅在渡口路径是唯一的目的地到达方式时，才选择渡口路径。
- **TravelTime** (点到点和多点) - 为路径请求设置开始或结束旅行时间。有关如何定义旅行时间和其他基于时间的路径规划选项的详细介绍，请参阅[基于时间的路径规划](#)。
- **Updates** - 允许在临时修改道路或分段速度和道路类型的路径请求中包括瞬时更新。有关如何包括更新的信息介绍，请参阅[瞬态更新](#)。

定义路径参数选择时，应从 `RoutingPreferences` 和 `MatrixRoutingPreferences` 类开始。有关语法和说明的详细介绍，请参阅《开发人员参考》。

驾驶导航

可以获得点到点或多点路径的驾驶导航。这些指示列出了路径的一步步的程序。路径的每个分段都返回需要执行的步骤（例如，起动、右转等）、街道名称、方向、距离和时间等。驾驶导航有大量的选项，包括特定路段重点导航、更改导航的用语和格式，更改导航语言等。

可以在两个地方配置/请求驾驶导航。第一个地方是服务器或服务设置。第二个地方是使用 **RouteInstructionsRequest** 类将 **ReturnDirections** 属性设置为 **true**。

修改方向参数选择

下面的列表介绍了用于修改驾驶方向的可用参数选择：

- **DirectionBreakTurnAngle** - 该类设置将街道细分为新的方向字符串时所决定的旋转角度值。有时，为了遵循路径，街道会进行很大的方向转变，同时保持同一名称。通过使用该值，可以指定应开始新方向的旋转角度。有效的值是 0 至 180 度。默认为 45 度。
- **DirectionsGeneratorName** - 指定一个用于创建路径方向的用户、服务器端方向生成器。如果为服务器或服务配置了一个或多个方向生成器，此参数选择可以用于强制由一个特定的生成器处理多个方向生成器。例如，设置“myGenerator”将强制服务器或服务使用名称被配置为“myGenerator”的生成器。仅当已使用自定义方向生成器配置服务器或服务后才使用该参数选择。请参阅服务器或服务的文档以获取有关如何创建并使用方向生成器的指示。
- **Focus** - 指定路径的焦点。焦点路径是整个路径的子集，它集中于路径的起始或结束位置。在开始处集中的路径将用户从其原点（包括原点）安排到第一个主要高速公路的路径。在结束处集中的路径将用户从该路径中的最后一个高速公路（包括该高速公路）安排到目的地的路径。若在该路径中没有任何主要高速公路，则关注的路径将与非关注的路径相同。**NONE** 表示将返回整个路径的方向，**START** 表示将返回路径的起始位置，**END** 表示将返回路径的结束位置。默认为 **NONE**。
- **ShowDistance** - 表示是否返回驾驶导航方向的距离。默认为 **true**。
- **ShowPrimaryNameOnly** - 指定是否仅显示主要街道名称。经常会出现一个街道包含多个名称的情况。这用来指示在方向中是显示该街道的所有名称还是仅显示主要名称。如果设置为 **true**，则仅显示将在导航中使用的街道的主名称。如果设置为 **false**，将使用主名称和备用名称。默认为 **false**。
- **ShowTime** - 表示是否返回按驾驶导航方向驾驶所需的时间。默认为 **true**。
- **Style** - 指定要返回的导航的类型。客户端可以返回两种类型的导航，即正常和简明。正常导航是标准驾驶导航指示。例如，“在 Yonge Blvd 左转并向东南行驶 (1 s)”。简明导航是较短的指示，更适用于无线服务。例如，“在 Yonge Blvd 左转”。可以设置为“正常”或“简明”。默认设置是“正常”。

设置导航语言

可以通过更改 **UserLocale** 属性来操控路径指示的输出语言。路径客户端使用 ISO 标准本地代码 **<language>_<country>**（例如，French 是 fr_FR、而 German 是 de_DE 等）。目前，路径客户端支持 EN、FR、DA、DE、FI、NL、NO、ES、PT、IT、SV。English (en_US) 是默认的语言。

路径几何体

创建地图和进行路径分析时，创建路径几何体是一项十分重要的功能。路径几何体是路径的可视化表示。默认情况下，不返回路径几何体。但是，可以将 `ReturnGeometry` 属性设置为 `true`，从而在响应中返回路径几何体。

避免点、图元和分段

可以避免或排除一些类型的道路、点或路径分段。确定路径之前，用户可能已获知相关的交通流量、事故、节假日、高峰时间或其他相关信息，因而已经知道应避免哪些道路和地点。

避免或排除的类型有以下三种：点、图元和分段。

避免点

目前有两种方法可以避免路径上的点。可以使用 `AvoidList` 类的 `Points` 属性，或者使用 `PointExclude` 类排除路径请求中的多个点。

 推荐使用 `PointExclude` 类来避免点。

避免图元

目前有两种方法可以避免路径上的图元（路径类型）。可以使用 `RoutingPreferences` 类的 `TravelPreferences` 属性避免任何路径类型。有关使用 `TravelPreference` 属性的详细信息，请参阅[路径规划参数选择](#)。

路径客户端还提供了另外一种方法，用于在进行路径计算时避免高速公路。要想将路径限制为本地道路时，此功能十分有用。可以使用 `AvoidList` 类的 `Features` 属性，将高速公路定义为要避免的图元。

避免分段

可以使用 `SegmentExclude` 类从路径请求中排除多个路径分段 (ID)。分段 ID 是分配给每条线路、道路或区段路径数据的标识符。可以通过在前一路径响应中返回它们来确定特定路径区段的分段 ID。有关如何在路径响应中返回分段 ID 的详细介绍，请参阅[返回分段信息](#)。

 如果无法计算路径，或者无法使用避免或排除来获得目的地，将不会返回路径方向。

基于时间的路径规划

对于运输系统、移动工作客户和计算准确驾驶时间和路径，基于时间的路径规划是一个关键的组件。可以为路径指定起点或终点，或者为多点路径中的每个中间点指定停止时间。例如，可以指定路径从位置 A 开始，经过位置 B、C 和 D，在后面的三个经过点，每个点需要停留五分钟的时间用来与客户交谈或装货。路径在位置 E 结束。

开始和结束时间

基于时间的路径规划的功能之一是指定路径的开始或结束时间。使用其他日期时间参数选择时，路径规划中的开始和结束时间十分重要。例如，如果为服务器配置了表示高峰时间交通流量模式的数据，那么路径的开始和结束时间对于分析有重要的意义。可以使用 **TravelTime** 类设置开始和结束时间。下面的属性用于控制旅行时间：

- **DateTime** - 指定开始或结束路径计算的日期和时间。
- **TimeZone** - 指定旅行时间中使用的时区。时区使用与格林威治标准时间 (GMT) 的时差值来定义。
- **TravelTimePreference** - 指定定义的旅行时间是否为开始或结束时间。使用 **StartTravel** 或 **EndTravel** 定义两个成员。

下面的代码示例介绍如何设置路径的旅行时间。这个示例包括 **StartTravel** 参数选择，定义了开始时间（年、月、日、时（24 小时制）、分、秒）和时区：

```
Public Shared Sub New_TravelTime()
    Dim startTime As DateTime = New DateTime(2005, 5, 1, 14, 0, 0)
    Dim timeZone As MapInfo.Routing.TimeZone = _
        New MapInfo.Routing.TimeZone(-4)
    Dim travelTime As MapInfo.Routing.TravelTime = _
        New MapInfo.Routing.TravelTime(startTime, _
        TravelTimePreference.StartTravel, timeZone)
End Sub
```

停止时间

可以在路径的任意中间点指定停止时间。此时间将增加路径的整体时间。如果用户需要在多点路径的特定位置消耗一定的时间（如运输卡车装货或卸货），则使用停止时间来定义这些消耗时间尤为有用。通过指定 **Stop** 和 **StopTime** 属性，可以为 **ViaPoint** 类添加停止。

- **Stop** - 可以指定是否在一个经过点停止。默认情况下，在经过点处不计算停止 (**false**)。通过将 **Stop** 定义为 **true**，可以指定是否想要在经过点停止。
- **StopTime** - 将停止时间增加到沿该路径的任意经过点。该停止时间由 **TimeSpan** 定义，并将增加到该路径的总时间。默认停止时间为 0 秒。**TimeSpan** 可以用字符串格式 “[**-**]d.hh:mm:ss.**ff**” 表示，其中 “-” 是可选的符号，用于负数的 **TimeSpan** 值，“**d**” 表示天数，“**hh**” 表示小时、“**mm**” 表示分钟，“**ss**” 表示秒，“**ff**” 表示秒的小数部分。例如，**TimeSpan** 定义为 “11.13:46:40” 表示 11 天 13 小时 46 分 40 秒。

Iso 路径规划（驾驶时间和驾驶距离）

IsoChrones 和 **isoDistances** 对于决策制定而言是十分有价值的信息。**isoChrone** 是一个由点构成的多边形或集合，表示可以在给定的时间内从起点遍历的网络区域。**isoDistance** 是一个由点构成的多边形或集合，表示与起点相距一定距离的区域。它们可以用于确定驾驶时间或与一个位置的驾驶距离边界。零售、银行和保险行业的用户可以使用此服务确定潜在的市场或任何给定资产的风险。这些边界可以用于进一步的分析，例如，确定邮件列表中的哪些客户位于 **iso** 边界之内，从而向这些客户通知新的服务或新店开张。

创建 IsoChrone（驾驶时间）

创建 IsoChrone 时，您希望获得一个或多个多边形/节点集，表示在给定的时间量（驾驶时间代价）内从起始位置可以旅行的区域。

代价指定计算 IsoChrone 所使用的时间。代价表示从起点旅行到道路网络上的计算点所需的时间。可以指定多个代价，以生成同心带（多 IsoChrone），可视化地表示可以在道路网络上到达的不同距离。一个标记与多个代价相关联。可以为请求中的每个代价指定一个标记或 ID，以便在响应中标识相应的 isogram 结果（几何体）。

可以在 IsoChronePreferences 和 IsogramPreferences 类中指定大量的参数选择。利用这些参数选择，可以获得想要的分析输出结果。

IsoChronePreferences

IsoChronePreferences 类还提供以下参数选择：

- **DefaultAmbientSpeed** - 确定未指定的野外网络速度。未在网络中标识的道路可能是车道和/或通道。例如，如果用户位于野外网络上的一个 IsoChrone 上只剩五分钟车程的位置，边界点将位于根据周围速度和剩余时间得出的距离位置。因此，如果这时的周围速度为每小时 15 英里，边界点将位于 1.25 英里的距离处。
- **AmbientSpeedOverride** - 重载指定道路类型的周围速度。可以重载所有道路类型的周围速度。例如，由于交通繁忙，可在周末将城市主要道路的速度设置为 30 Mph。

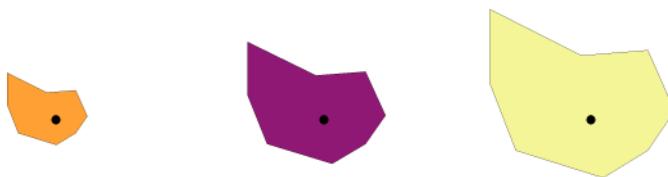
IsogramPreferences

IsogramPreferences 类还提供以下参数选择：

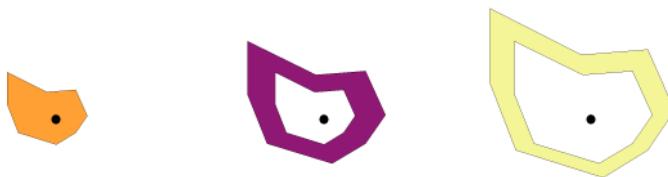
- **BandingStyle** - 指定在结果中使用的条带样式。条带样式是多个时间或距离条带的样式，这些时间或距离条带可基于多个代价显示。样式包括 Donut（每个边界都通过减去下一个最小边界确定）或 Encompassing（每个边界都独立于其它所有边界）。可通过指定多个代价因子来请求多个 IsoChrone 条带，例如要求 IsoChrone 从同一起点 5 分钟和 15 分钟离开。当接近同心条带时结束这些操作。用户可以选择为 5 分钟和 15 分钟离开之间的内容显示完整的数据集（显示任何内容的 Encompassing 样式）或仅显示这两者之间的条带（Donut 样式）。

两个边界（时间）几乎相同时，Donut 条带可能产生路径规划服务器问题，可以使用三个请求设置来避免这种情况：

- a. 如果使用 Donut 条带，则不使用最大野外距离设置。如果必须使用此设置，应将它设置为尽可能大的值。
- b. 使用 Donut 条带时，不使用低周围速度。
- c. 不要进行代价增幅相对代价而言比较小的请求。例如，请求 4、5 和 6 分钟的代价（从 4 分钟开始递增 1 分钟）一般不会产生问题，但是 120、121 和 122 分钟则可能产生问题。代价越大，代价的增幅也应越大。

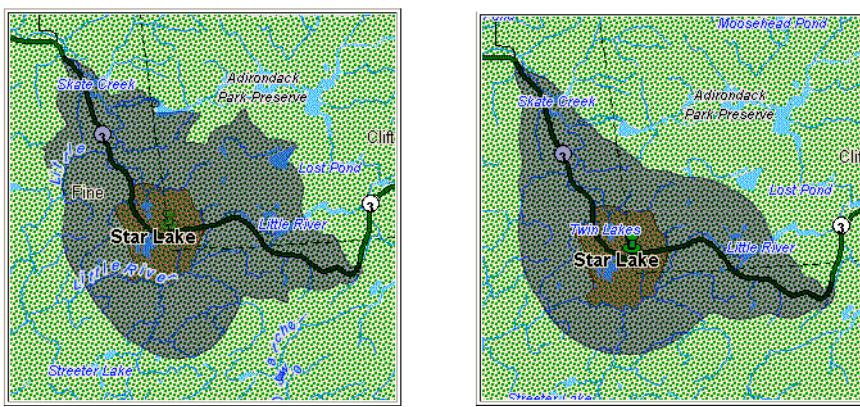


上面的示例介绍了为距离起点 5、10 和 15 分钟代价的 *Encompassing* 样式条带返回的几何体。请注意每个几何体如何包括前一 *isoChrone* 的区域。



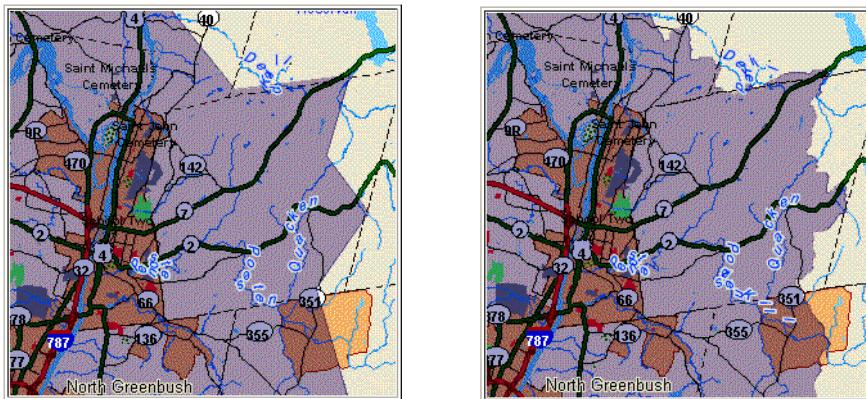
上面的示例介绍了为距离起点 5、10 和 15 分钟代价的 *Donut* 样式条带返回的几何体。注意每个几何体如何从前一几何体的终点开始，从而排除前一 *isoChrone* 的区域。

- **MajorRoadsOnly** - 确定要将何种道路网络用在计算中。该网络可以仅包括主要道路或所有道路。主要道路是主干道或高速公路。若选择使用主要道路，将会提高性能，但会降低精确性。下面的图像描述了主要道路选项的行为。左边的图像显示 **MajorRoadsOnly** 被设置为 `true`，而右边的图像显示 **MajorRoadsOnly** 被设置为 `false`。注意，如果 **MajorRoadsOnly** 被设置为 `false`，计算 *iso* 响应时，服务使用侧街道和其他类型的辅道。



- **MaxOffRoadDistance** - 指定使用周围速度时驶离道路网络所需要行驶的最大距离。此属性的默认设置为无限制，允许野外旅行到 *isoChrone* 的最大范围。如果最大野外距离设置为很小的值，服务器可能无法生成响应。

- **ReturnAccessibleNodes** - 将 isoChrono 请求返回的 isogram 或图元的类型指定为节点集 (MultiPoint)。默认情况下，ReturnAccessibleNodes 为 False。此属性为 true 时，将返回根据 isoChrono 计算可以到达的道路网络上的所有点。
- **ReturnGeometry** - 将 isoChrono 请求返回的 isogram 或图元的类型指定为几何体 (MultiPolygon)。默认情况下，ReturnGeometry 为 true。此属性为 false 时，将返回根据 isoChrono 计算可以到达的道路网络上的所有点。
- **ReturnHoles** - 指定是否在路径响应中返回洞。基于道路网络，洞是位于无法在所要求时间或距离内到达的较大边界内的区域。这些袋状区域经常位于那些不易遍历的地方道路附近。洞可按其原样返回，或整个删除。只有 ReturnGeometry 设置为 true (默认) 时，才会应用该设置。若未设置该属性，则使用默认设置 false。
- **ReturnIslands** - 指示是否在路径响应中返回岛。岛是位于可在所要求时间或距离内到达的主要边界外部的小块区域。这些区域经常位于主要高速公路的出口坡道外。岛可按其原样返回，或整个删除。只有 ReturnGeometry 设置为 true (默认) 时，才会应用该设置。若未设置该属性，则使用默认设置 false。
- **SimplificationFactor** - 指定多边形复杂度简化因子。简化因子包括应返回原始点的百分比或生成的多边形应基于的内容。多边形或点集可以包含多个点。简化因子是介于 0 到 1.0 之间的小数 (1 为 100%，0.01 为 1%)。数字越小表示存储空间越少且传输时间越小。只有结果类型为几何体时，才会应用该设置。若未指定此属性，将使用默认设置 0.05 (5%)。下面的图像介绍了简化因子选项的行为。左边的图像显示简化因子被设置为 0.01，右边的图像显示简化因子被设置为 1。



- **TimeOut** - 指定服务器创建等值线所允许的时间量。该属性的默认值是 600 秒 (10 分钟)。

创建 IsoDistance（驾驶距离）

创建 isoDistance 时，您希望获得一个或多个多边形/节点集，表示与起点的距离（驾驶距离代价）已给定的情况下可以旅行的道路网络区域。

为 `isoDistance` 请求指定的代价与 `isoChrone` 相似，但是它的代价值是用于计算 `isogram` 的距离值。代价表示从起点旅行到道路网络上的计算点所需行驶的距离。可以指定多个代价，以生成同心带（多 `isoDistances`），可视化地表示可以在道路网络上到达的不同距离。标记与多个代价相关联。可以为请求中的每个代价指定一个标记或 ID，以便在响应中标识相应的 `iso` 结果（几何体）。

以下代码示例说明如何创建一个 `IsoDistanceDefinition`。响应将创建一个 `isogram`（几何体），这个 `isogram` 使用点作为中心，并且使用以英里为单位的代价 (`pt, DistanceUnit.Mile`) 定义边界：

```
Public Shared Sub New_IsoDistanceDefinition()
    ' Create a point
    Dim coordSys As MapInfo.Geometry.CoordSys = _
        Session.Current.CoordSysFactory.CreateFromPrjString("1, 104")
    Dim dpt As MapInfo.Geometry.DPoint = _
        New MapInfo.Geometry.DPoint(-74, 42)
    Dim pt As MapInfo.Geometry.Point = _
        New MapInfo.Geometry.Point(coordSys, dpt)
    ' Create a definition
    Dim def As IsoDistanceDefinition = _
        New IsoDistanceDefinition(pt, DistanceUnit.Mile)
End Sub
```

可以在 `IsoDistancePreferences` 和 `IsogramPreferences` 类中指定大量的参数选择。利用这些参数选择，可以获得想要的分析输出结果。`IsogramPreferences` 类中指定的参数选择与 `isoDistance` 和 `isoChrone` 请求中指定的参数选择相同。有关 `IsogramPreferences` 的说明，请参阅 [IsogramPreferences](#)。

IsoDistancePreferences

`IsoDistancePreferences` 类还提供以下参数选择：

- **DefaultPropagationFactor** - 当查找 `isoDistance` 边界时，为所允许的野外网络确定剩余代价（距离）的野外网络百分比。未在网络中标识的道路可能是车道和/或通道。传播因子是用来计算起点和 `isoDistance` 之间的距离的代价百分比。例如，如果用户位于野外网络上的一个 `isoDistance` 上只剩五分钟车程的位置，边界点将位于根据传播因子和剩余距离得出的距离位置。因此，如果传播因子为 0.15，边界点将位于 1.8 英里的距离位置。
- **PropagationFactorOverrides** - 重载指定道路类型的传播因子。可以重载所有道路类型的传播因子。例如，对于主要城市道路，可以将它设置为 0.24：

使用路径规划数据更新请求

具备访问路径数据的能力并且使用这些数据创建更准确和有意义的路径是一项十分有用的功能。创建路径并且获得路径响应之后，可能发现结果包含不想要经过的位置或分段和可能需要减速或加速的道路类型或分段。`MapXtreme` 可以根据路径数据返回分段信息，并且使用路径数据中的信息修改/更新路径请求（瞬态更新）。这意味着可以更新路径规划网络而不需要重新处理基本数据。

瞬态更新过程包括两个步骤：

1. 在路径响应中返回路径数据分段信息。确定用于道路类型或速度更新的分段时需要这些信息。有关详细信息请参阅[返回分段信息](#)。
2. 根据响应中返回的信息，可以对路径数据进行临时更新。这称为瞬态更新。这些更改通过路径请求提交，而且只有服务器或服务正在处理这种特定的请求时才会有效。此功能可用于包括一些动态流量 / 偶发数据，或者为特定的道路类型设置参数选择。有关详细信息请参阅[瞬态更新](#)。

返回分段信息

为了避免或更新道路分段的道路类型或速度数据，需要知道分段 ID 和其他道路信息。**MapXtreme** 能够返回道路分段数据，包括诸如分段 ID、名称、距离、道路类型、速度、旅行时间几何体和其他的重要数据。

默认情况下，路径请求中不返回分段信息。使用类，将 `ReturnSegmentData` 设置为 `true`。还可以使用选项为返回信息定义速度和角度的度量单位。默认情况下，速度单位为 `Mph`（英里每小时），角度单位为度。分段数据是创建驾驶导航所必需的数据，而且包含路径中的每个分段的详细信息。返回的分段信息包括：

- 街道名称（包括备用名称）
- 街道名称所使用的语言
- 罗盘方向
- 分段 ID
- 单行道布尔值
- 环形路布尔值
- 收费道路布尔值
- 道路类型
- 限速
- 旋转角度
- 遍历分段所需要的时间
- 分段距离

瞬态更新

为了创建更准确和更有意义的路径计算，可以临时更改路径数据。这些更改通过每个路径请求提交，而且只有服务器或服务正在处理这种特定的请求时才会有效。可以使用此功能避开正处于高峰时段的特定高速公路或者在请求中降低道路分段的速度，以表示实际的交通流量模式，在路径规划的基础上获得更精确的时间。瞬态更新可以包含在任何类型的路径请求（点到点、多点或矩阵路径）之中。通过进行这些类型的修改，可以：

- 设置点、分段或道路类型的速度。
- 按数值更改（提高或降低）点、分段或道路类型的速度。
- 按百分比更改（提高或降低）点、分段或道路类型的速度。
- 设置分段的道路类型。

在瞬态更新中定义速度时，有三种定义类型，每种都定义不同的速度值：

- **Speed** - 指定分段的新速度。
- **Relative** - 指定速度的增加或减小相对值。这些值表示速度的变化。例如，如果值为 10，默认速度将提高 10，值 10 MPH 将默认速度提高 10 MPH。
- **Percentage** - 指定速度的增加或减小百分比。值介于 -100 和 100 之间。例如，如果值为 50，将把默认速度提高 50%（速度 30 MPH 将提高至 45 MPH），如果值为 -50，将把默认速度降低 50%（速度 30 MPH 将降低至 15 MPH）。

使用瞬态更新修改分段。

获得路径响应后，可能发现结果包含交通流量过快或过慢的特定路径分段或（道路）分段组，或者想要在请求中更新道路类型。可以使用瞬态更新设置道路分段或道路类型的速度，重新计算路径以获得更准确的结果。使用下面的类执行各种分段瞬态更新：

- **SegmentRoadTypeUpdate** - 更新分段的道路类型。下面的代码示例介绍如何为已定义的分段 ID “S1256”，RoadType.MAJOR_ROAD_URBAN 定义新的道路类型。
- **SegmentSpeedUpdate** - 使用新速度更新分段的速度。

下面的示例介绍如何为指定的路径分段 (S1256) 定义新速度 50 英里每小时 (Velocity(50, VelocityUnit.Mph))：

```
Public Shared Sub New_SegmentSpeedUpdate()
    ' Create the velocity object
    Dim velocity As Velocity = New Velocity(50, VelocityUnit.Mph)
    ' Create the update object
    Dim update As SegmentSpeedUpdate = New SegmentSpeedUpdate("S1256",
velocity)
End Sub
```

- **SegmentRelativeSpeedUpdate** - 使用速度更改来更新分段的速度。

下面的示例介绍如何将指定路径分段 (T1256) 的速度提高 5 英里每小时 (Velocity(5, VelocityUnit.Mph))：

```
Public Shared Sub New_SegmentRelativeSpeedUpdate()
    ' Create the velocity object
    Dim velocity As Velocity = New Velocity(5, VelocityUnit.Mph)
    ' Increase the speed by 5 mph.
    Dim update As SegmentRelativeSpeedUpdate = New
SegmentRelativeSpeedUpdate("T1256", velocity)
End Sub
```

- **SegmentPercentageSpeedUpdate** - SegmentPercentageSpeedUpdate - 使用默认速度的百分比更新分段的速度。

下面的示例介绍如何将指定路径分段的速度提高 20% ("S1256", 20)：

```
Public Shared Sub New_SegmentPercentageSpeedUpdate()
```

```
' Increase the speed by 20 percent.  
Dim update As SegmentPercentageSpeedUpdate = New  
SegmentPercentageSpeedUpdate("S1256", 20)  
End Sub
```

使用瞬态更新修改点

获得路径响应后，可能发现结果包含一些交通流量过慢或过快的位置（点）。可以使用瞬态更新设置距离点最近的道路分段的速度，重新计算路径以获得更准确的结果。使用下面的类执行各种点瞬态更新：

- **PointSpeedUpdate** - 使用新速度更新距离点最近的分段的速度。

下面的示例介绍如何为距离定义的点 (pt) 最近的道路定义新速度 50 英里每小时 (Velocity(50, VelocityUnit.Mph))：

```
Public Shared Sub New_PointSpeedUpdate()  
    ' Create a point  
    Dim coordSys As MapInfo.Geometry.CoordSys =  
Session.Current.CoordSysFactory.CreateFromPrjString("1, 104")  
    Dim dpt As MapInfo.Geometry.DPoint = New MapInfo.Geometry.DPoint(-74,  
42)  
    Dim pt As MapInfo.Geometry.Point = New  
MapInfo.Geometry.Point(coordSys, dpt)  
    ' Create the velocity object  
    Dim velocity As Velocity = New Velocity(50, VelocityUnit.Mph)  
    ' Create the update object  
    Dim update As PointSpeedUpdate = New PointSpeedUpdate(pt, velocity)  
End Sub
```

- **PointRelativeSpeedUpdate** - 使用速度更改来更新距离点最近的分段的速度。

下面的示例介绍如何将距离定义的点 (pt) 最近的道路的速度提高 5 英里每小时 (Velocity(5, VelocityUnit.Mph))：

```
Public Shared Sub New_PointRelativeSpeedUpdate()  
    ' Create a point  
    Dim coordSys As MapInfo.Geometry.CoordSys =  
Session.Current.CoordSysFactory.CreateFromPrjString("1, 104")  
    Dim dpt As MapInfo.Geometry.DPoint = New MapInfo.Geometry.DPoint(-74,  
42)  
    Dim pt As MapInfo.Geometry.Point = New  
MapInfo.Geometry.Point(coordSys, dpt)  
    ' Create the velocity object  
    Dim velocity As Velocity = New Velocity(5, VelocityUnit.Mph)  
    ' Increase the speed by 5 mph.  
    Dim update As PointRelativeSpeedUpdate = New  
PointRelativeSpeedUpdate(pt, velocity)  
End Sub
```

- **PointPercentageSpeedUpdate** - 使用默认速度的百分比更新距离指定位置最近的点的速度。

下面的示例介绍如何将距离定义点最近的道路的速度提高 20% (pt, 20):

```
Public Shared Sub New_PointPercentageSpeedUpdate()
    ' Create a point
    Dim coordSys As MapInfo.Geometry.CoordSys =
Session.Current.CoordSysFactory.CreateFromPrjString("1, 104")
    Dim dpt As MapInfo.Geometry.DPoint = New MapInfo.Geometry.DPoint(-74,
42)
    Dim pt As MapInfo.Geometry.Point = New
MapInfo.Geometry.Point(coordSys, dpt)
    ' Increase the speed by 20 percent.
    Dim update As PointPercentageSpeedUpdate = New
PointPercentageSpeedUpdate(pt, 20)
End Sub
```

使用瞬态更新修改道路类型速度

在确定路径之前，您可能已经知道某些旅行信息（例如施工、事故或高峰时间），需要降低或提高特定道路类型的速度。例如，考虑到节假日和周末旅行的因素，可以降低所有高速公路的速度。可以通过选项降低路径计算中使用特定道路类型的机会。确定最快的路径时，可以降低特定道路类型的速度（将这些道路设为较低的优先级），并且降低在计算中使用它们的机会。使用下面的类执行各种道路类型瞬态更新：

- **RoadTypeSpeedUpdate** - 使用新速度更新特定道路类型的速度。

下面的示例介绍如何为所有的城市主要道路 (RoadType.MAJOR_ROAD_URBAN) 定义新速度 50 英里每小时 (Velocity(50, VelocityUnit.Mph)):

```
Public Shared Sub New_RoadTypeSpeedUpdate()
    ' Create the velocity object
    Dim velocity As Velocity = New Velocity(50, VelocityUnit.Mph)
    ' Create the update object
    Dim update As RoadTypeSpeedUpdate = New
RoadTypeSpeedUpdate(RoadType.MAJOR_ROAD_URBAN, velocity)
End Sub
```

- **RoadTypeRelativeSpeedUpdate** - 使用速度更改来更新道路类型的速度。

下面的示例介绍如何将所有城市主要道路 (RoadType.MAJOR_ROAD_URBAN) 的速度提高 5 英里每小时 (Velocity(5, VelocityUnit.Mph)):

```
Public Shared Sub New_RoadTypeRelativeSpeedUpdate()
    ' Create the velocity object
    Dim velocity As Velocity = New Velocity(5, VelocityUnit.Mph)
    ' Increase the speed by 5 mph.
```

```
Dim update As RoadTypeRelativeSpeedUpdate = New
RoadTypeRelativeSpeedUpdate(RoadType.MAJOR_ROAD_URBAN, velocity)
End Sub
```

- **RoadTypePercentageSpeedUpdate** - 使用默认速度的百分比更新道路类型的速度。

下面的示例介绍如何将所有城市主要道路的速度提高 20% (RoadType.MAJOR_ROAD_URBAN, 20):

```
Public Shared Sub New_RoadTypePercentageSpeedUpdate()
    ' Increase the speed by 20 percent.
    Dim update As RoadTypePercentageSpeedUpdate = New
    RoadTypePercentageSpeedUpdate(RoadType.MAJOR_ROAD_URBAN, 20)
End Sub
```

线性参考

本章介绍使用和 **MultiCurve** 图元几何体相关的 M（测量）值以绘制地图和分析线性网络的功能。

在本章中：

- ◆ 什么是线性参考 354
- ◆ 使用线性参考的 M 值 354
- ◆ 曲线顺序 356
- ◆ 线性参考示例应用程序 357

什么是线性参考

线性参考是相对于传统坐标参考系的另一种参考系，它可以将线性图元的位置和地球上的点相联系。线性参考可用于多种领域，包括水资源管理、交通和石油/天然气勘探。作为线性网络的部分，任何可用地图表示的实际资产都可保存为数据，该数据说明资产或与资产相关的情况或事件。数据在 MultiCurve 对象上和该位置的 X、Y 坐标一起保存为 M（或度量值）。为更好的资源管理，可进一步地图化和分析 M 值。

M 值是线性参考的基础。M 值可以保存任何度量。线性图元上的点可以从相对于其他点的已知点进行参考。高速公路上的英里标志柱就是一个典型示例。英里标志柱的 ID 是某种 M 值，它表示从一个已知位置（一般为高速公路十字路口或国家边境）开始的某段距离。

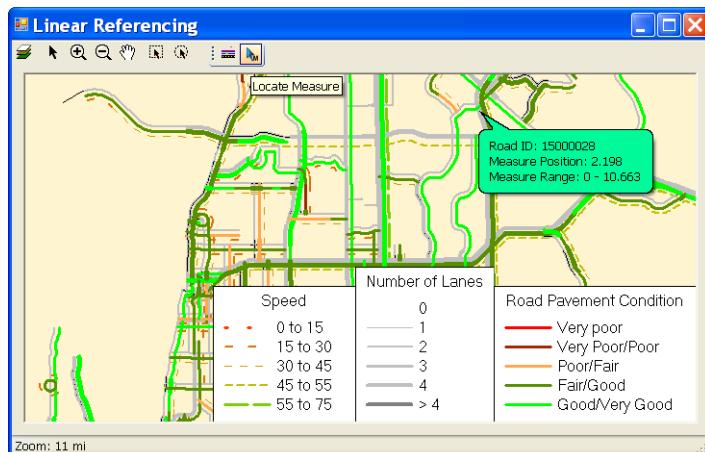
M 值一般用于基础设施数据管理器和交通数据管理器，以便更好地显示、查询、监控和分析线上的资产、状况和事件。例如，开车的人如果迷路致电紧急救援电话中心，只需向接线员提供英里标志柱信息，接线员就立即可以确定位置。因为救援电话中心已经知道了英里标志柱的位置。只要有了这些信息，救援电话中心就可以向相应的位置指派人员。

有关支持 MultiCurves 和 M 值的 Geometry 模型概述，请参阅[第 16 章：空间对象和坐标系](#)。

使用线性参考的 M 值

MapXtreme 提供多种运算，均可以使用 MultiCurve 几何体的度量 (M) 值，以提供线性参考和动态分段功能。根据定义，此方法是一种可实例化的 MapInfo.Geometry.LinearReferencingOperations 类的一部分，它可根据需要灵活方便地添加属性或扩展行为。

这些操作可以同时显示线性网络中的多个属性，它们也可以称为动态分段，显示时无需存储同一个几何体的多个副本。例如，高速公路可以表示为单独的线性图元，各个图元均可以表示道路状况、车道类型和道路材料。它们均表示相同的真实几何体，但你可以更轻松地查看多种类型的信息。



MapXtreme 中提供的线性参考示例应用程序

线性参考方法在应用于任何一个维 MultiCurve 几何体时须遵循 OGC Simple Feature 规范。MultiCurves 可包含几条不相交的曲线。换而言之，两条曲线可包含在 MultiCurve 内，其中一条曲线的尾与另一条曲线的头不匹配。但是，程序可能会给运算一个开始量度和一个结束量度，以便使量度能应用于整个 MultiCurve。在此方案和类似的方案中，我们假定：即使实际位置数据不同于一条曲线的头和另一条曲线的尾之间的 (X, Y), MapXtreme 仍假定量度值相同。例如，在高速公路网络中，几条高速公路形成十字形，但并不相交，因为一条高速公路在另一条公路上方的高架桥上。在这些情况下，根据线性表示方法，下面的高速公路可能会在另一条高速公路前后实际显示为断裂。

LinearReferencingOperations 类提供的方法可分为三类：

- 度量值确定法
- 线性参考运算
- 动态分段运算 (PerpendicularOffset)

以下内容将简述这些运算。请参阅 `MapInfo.LinearReferencing` 命名空间以了解全部详情。

度量值确定法

形成此类的运算是一种辅助方法，它可帮助设置和管理 MultiCurve 上的度量值。如果至少有两个 M 值，请使用它们在 MultiCurve 上填充丢失的节点；如果没有 M 值，则使用相应的开始节点和结束节点设置 M 值。对于一两个特定的位置来说，这类值是往往已知的；但对于其他的位置来说，这类值需要根据距离按比例计算。

- `CalculateMissingMeasures(MultiCurve)`
- `SetMeasures (MultiCurve, double startMeasure, double endMeasure)`
- `SetMeasuresAsDistance(MultiCurve, double startDistance)`
- `DropMeasures (MultiCurve)`
- `ScaleMeasures(MultiCurve, double scale)`
- `TranslateMeasures(MultiCurve, double offset)`
- `TranslateMeasures(MultiCurve)`
- `Reverse(MultiCurve)`

线性参考运算

第二套运算沿着 MultiCurve 几何体使用线性参考系统，以定位位于特定度量值的点或提取两个度量值之间的分支曲线。虽然 MapXtreme 可存储位于 MultiCurve 节点并且已输入这些方法的引用度量值，但实际上并不是所有用户数据都会有存储在几何体中的引用系统（或者它们使用的存储技术无法保存此信息）。因此，这些方法一般有两种重载形式，在其中一种形式中，引用系统在输入 MultiCurve 中进行管理；而在另一种形式中，引用系统则是一种根据输入 MultiCurve 几何体起始节点的起始度量。

- `LocateAlong(MultiCurve, double Measure)`
- `LocateAlong(MultiCurve, double startMeasure, double endMeasure, double Measure)`
- `LocateMeasure (MultiCurve, Point)`

- LocateMeasure (MultiCurve, double startMeasure, double endMeasure, Point)
- LocateBetween(MultiCurve, double subCurveStartMeasure, double subCurveEndMeasure)
- LocateBetween(MultiCurve, double startMeasure, double endMeasure, double subCurveStartMeasure, double subCurveEndMeasure)

FeatureGeometry.Distance 方法

MapInfo.Geometry.FeatureGeometry 类包括 Distance 方法的重载，输入点没有在曲线上定位时，MapXtreme 可使用它沿着 MultiCurve 定位度量值。此方法返回 MultiCurve，它表示曲线外的输入点和曲线上的投射点之间的最短或最长距离。要指定最短或最长的距离，必须将 minimumDistance 作为布尔值传入。最短的距离为 true。返回的 MultiCurve 是 LocateMeasure 方法的输入，它在 MultiCurve 上返回的点离输入点最近（或最远）。

如果传入的 MultiCurve 和 Point 位于不同的坐标系，则 Distance 运算将在 MultiCurve 实例的坐标系中执行。生成的 FeatureGeometry 始终与 MultiCurve 实例处于相同的坐标系中。

动态分段运算 (PerpendicularOffset)

垂向偏移是从现有 MultiCurve 中生成新 MultiCurve 的运算。它不同于以上方法，以上方法仅提取底层 X、Y 坐标相同的分支曲线。此运算创建的平行偏移 MultiCurve 是从原始 MultiCurve 垂直测量的某段距离。然后，可以进一步操作新的 MultiCurve，例如查询其属性并分析结果。

PerpendicularOffset 可用于线性网络的动态分段，您可根据 M 值将 MultiCurve 再分段。例如，在道路施工的应用中，可以使用道路网络的 M 值，以描述高速公路路段状态是在建设中还是在修整中。每个分段都可以显示在地图上，偏离实际道路显示，上面有醒目的颜色代码或标记，以明确表示其状态。如需使用垂向偏移进行动态分段的示例，请参阅[线性参考示例应用程序](#)。

以下是 PerpendicularOffset() 方法的语法：

- PerpendicularOffset(MultiCurve, double distance)
- PerpendicularOffset(MultiCurve, double distance, DistanceUnit, int resolution)

正距离值表示按节点顺序（例如：节点 i 至节点 i+1）遍历时线的右边的垂直方向。负距离值则表示左边。距离值不得为零。

分辨率可用来沿着圆角的弧线内插点。支持 0 或大于 0 的值，最大值为 36。默认值为 1。

此运算保存原始 MultiCurve 的度量值（如有）。但不保留原始 MultiCurve 中的 Elevation (Z) 值。

曲线顺序

MapInfo.LinearReferencing 命名空间包括 ICurveSorter 界面，以处理构成 MultiCurve 的单个曲线的排序顺序。在调用以下任何线性参考运算时，排序顺序至关重要：

- CalculateMissingMeasures

- SetMeasures
- SetMeasuresAsDistance
- LocateAlong
- LocateMeasure
- LocateBetween

如未指定排序顺序，**MapXtreme** 将首先返回最长的曲线，剩下的曲线将以未知顺序返回。例如，在使用无序的 **MultiCurve** 上的 **CalculateMissingMeasures** 时，**MapXtreme** 可根据 **MultiCurve** 中的节点位置计算出节点错误的 M 值。提供正确的排序顺序将消除这个问题。

ICurveSorter 界面提供 **SortCurves** 方法，它可将曲线顺序返回为整数数组。然后，您需要使用该数组获取曲线列表。有关代码示例，请参阅“开发人员参考”。

i 曲线排序后不会更改节点的 X、Y 位置。

接口还提供 **DefaultCurveSorter** 类，它在执行时仅将组件曲线返回为 **MultiCurve** 本身通知的形式。不得通过其 **SortCurve** 方法执行排序。

线性参考示例应用程序

MapXtreme 安装的 **Samples** 文件夹中所提供的内容为线性参考示例应用程序，它可生成多个平行线段，表示道路的不同特征。在此示例中，表示路况、最大速度区域、车道数量的道路数据显示时作为实际道路上的 **MultiCurves** 偏移量，以表示特定路段的状况或特征。每种特征或状况其显示的线条样式都不同。

在常规的 **MapXtreme** 表中，属性作为一个整体应用于图元。在线性网络中，图元的属性称为 **Measure** 值，它应用于其线性几何体表现形式的分段。例如，路面一般有路面情况质量好的路段或质量差的路段，或者还有正在修理的路段。这些状况一般不会和整个路段一致，也不会和 **MultiCurve** 上的节点完全一致。但是，线性参考系统可以使用相对于图元（而不是相对于地球位置）的度量值，通过这种线性参考系统，您可以采用有用的新方法捕获、查询、显示和分析分段数据。

通过其线性参考运算，**MapXtreme** 可根据这些 M 值对 **MultiCurves** 进行动态分段。请参阅[使用线性参考的 M 值](#)以了解所支持运算的描述。

Web 图元服务

MapXtreme 提供通过 Internet 或内联网 Web 图元服务 (WFS) 托管和/或访问地图图元数据的功能。

在本章中：

- ◆ **Web 图元服务** 359
- ◆ 理解 WFS 服务器的操作 359
- ◆ 配置 WFS 服务器 363
- ◆ 通过编程方式使用 MapXtreme WFS 客户端 369
- ◆ 根据 WFS 响应创建地图图层 372

Web 图元服务

MapXtreme 提供 Web 图元服务 (WFS) 实现，通过 Internet 或内联网发送请求以便检索以地理标记语言编码的地理空间数据。在 [Web 地图服务](#)生成地图图像时，对 WFS 服务器的请求会生成 GML，GML 是一种可以捕获地理数据的 XML。

基本 WFS 客户端可以向 WFS 服务器发送三类请求。“获取功能列表” (GetCapabilities) 请求要求服务器列出所能提供的地理数据以及可以对该数据执行的操作。“描述图元类型” (DescribeFeature) 请求要求服务器描述某个地理图元的数据。最后，可以要求服务器提供实际数据。

来自 WFS 的数据包含对该数据的描述。此时，检索到的数据尚不可作为 MapXtreme 中的地图图层进行查看。但可使用 MapXtreme 的 WfsClient 将 GML 输出转换为 MapXtreme 可以处理的格式: MultiFeatureCollections。随后，可以像 MapXtreme 中的任何其他 FeatureCollection 一样处理这些图元集合，从而应用主题、标注并执行各种分析操作。

作为 WFS Basic 配置文件，MapXtreme WFS 服务器实现符合 1.0.0 OpenGIS® Web 图元服务实现规范。在 www.opengis.org 中可以找到指向该文档的链接。目前，MapXtreme 的 WFS 实现不支持 Transaction WFS 规范。

通过 MapXtreme 的 WFS 实现对图元的请求可利用 HTTP GET 或 HTTP POST 请求建立。此响应默认的返回格式为 GML2；然而，请求可以明确指定以 GML3 格式返回响应。

MapXtreme 对 WFS 的支持包括两部分 — WFS 服务器和 WFS 客户端。本章介绍如何配置 WFS 服务器使之托管数据，方便其他用户访问。请参阅[配置 WFS 服务器](#)。

如有兴趣了解如何在 Internet 上或从内联网中访问来自其他 WFS 服务器的数据，请参阅[通过编程方式使用 MapXtreme WFS 客户端](#)。

下一节介绍可通过 MapXtreme WFS 执行的 WFS 服务器操作。

理解 WFS 服务器的操作

有三种 WFS 服务器操作构成 MapXtreme WFS 服务器实现的基础，它们分别是：GetCapabilities、DescribeFeatureType 和 GetFeature。

GetCapabilities

GetCapabilities 请求是针对 WFS 服务器的查询，查询的目的是了解服务器提供哪些地理数据以及可对这些数据执行哪些操作。对 GetCapabilities 请求的响应是一份 XML 文档，该文档描述 WFS 支持的操作以及可以服务的所有图元类型列表。在首次访问 WFS 服务器时，也许希望请求服务功能列表。

GetCapabilities 通过 HTTP GET 和 HTTP POST 支持。

GetCapabilities 请求的 XML 输出如下。部分重要元素以粗体显示，包括支持的功能、可用的图元类型以及各种筛选操作（用于请求某图元类型数据的子集）。

```
<?xml version="1.0" encoding="utf-8" ?>
- <WFS_Capabilities xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:miwfs="http://www.mapinfo.com/wfs"
  xsi:schemaLocation="http://schemas.opengis.net/wfs/1.0.0/WFS-
  capabilities.xsd" version="1.0.0" updateSequence="0"
  xmlns="http://www.opengis.net/wfs">
  - <Service>
    <Name>Sample WFS Server</Name>
    <Title>Sample WFS Server</Title>
    <OnlineResource>http://localhost/Wfs/GetFeature.ashx</OnlineResource>
  </Service>
  - <Capability>
    - <Request>
      - <GetCapabilities>
        - <DCPType>
          - <HTTP>
            <Get onlineResource="http://localhost/Wfs/GetFeature.ashx" />
            <Post onlineResource="http://localhost/Wfs/GetFeature.ashx" />
          </HTTP>
        </DCPType>
      </GetCapabilities>
    - <DescribeFeatureType>
      - <SchemaDescriptionLanguage>
        <XMLSCHEMA />
      </SchemaDescriptionLanguage>
    - <DCPType>
      - <HTTP>
        <Get onlineResource="http://localhost/Wfs/GetFeature.ashx" />
        <Post onlineResource="http://localhost/Wfs/GetFeature.ashx" />
      </HTTP>
    </DCPType>
    </DescribeFeatureType>
  - <GetFeature>
    - <ResultFormat>
      <GML2 />
    </ResultFormat>
    - <DCPType>
      - <HTTP>
        <Post onlineResource="http://localhost/Wfs/GetFeature.ashx" />
      </HTTP>
    </DCPType>
    </GetFeature>
  </Request>
  </Capability>
  - <FeatureTypeList>
    - <Operations>
      <Query />
    </Operations>
    - <FeatureType>
      <Name>miwfs:USA</Name>
      <Title>Title for usa</Title>
      <Abstract>Abstract for USA</Abstract>
    </FeatureType>
  </FeatureTypeList>

```

```
<Keywords>Keywords for USA</Keywords>
<SRS>epsg:4326</SRS>
<LatLongBoundingBox minx="-179.62816" miny="18.925255" maxx="-166.951403" maxy="71.42856" />
</FeatureType>
</FeatureTypeList>
- <ogc:Filter_Capabilities>
- <ogc:Spatial_Capabilities>
- <ogc:Spatial_Operators>
  <ogc:BBOX />
  <ogc:Equals />
  <ogc:Disjoint />
  <ogc:Intersect />
  <ogc:Within />
  <ogc:Contains />
</ogc:Spatial_Operators>
</ogc:Spatial_Capabilities>
- <ogc:Scalar_Capabilities>
  <ogc:Logical_Operators />
- <ogc:Comparison_Operators>
  <ogc:Simple_Comparisons />
  <ogc:NullCheck />
</ogc:Comparison_Operators>
- <ogc:Arithmetic_Operators>
  <ogc:Simple_Arithmetic />
</ogc:Arithmetic_Operators>
</ogc:Scalar_Capabilities>
</ogc:Filter_Capabilities>
</WFS_Capabilities>
```

DescribeFeatureType

在使用 GetCapabilities 请求查找可用的图元类型之后，可以使用 DescribeFeatureType 请求要求服务器提供有关其中一个或多个可用图元类型的详细信息。在 MapXtreme 中，WFS 图元类型由表表示，一个 WFS 图元等价于表中的一行信息。对 DescribeFeatureType 请求的响应包括图元类型的名称（表的名称）和表中各个属性的名称和类型（列的名称和类型）。结果以 GML 格式的 XML 架构文档返回。

DescribeFeatureType 通过 HTTP GET 和 HTTP POST 支持。图元类型中的每个属性即构成返回架构的一个元素。

MapXtreme 不返回以下列类型作为属性：

- MIDbType.Binary
- MIDbType.CoordSys
- MIDbType.Grid
- MIDbType.Key
- MIDbType.Raster
- MIDbType.Style

下面显示了 **DescribeFeatureType** 响应文档的一部分。本例中，请求是针对名为 USA 的图元类型。USA 图元类型带有几个分别由几何体列 OBJ、State 和 State_Name 数据列表表示的属性。

```
<?xml version="1.0" encoding="utf-8" ?>
- <schema targetNamespace="http://www.mapinfo.com/wfs"
  xmlns:miwfs="http://www.mapinfo.com/wfs"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified"
  xmlns="http://www.w3.org/2001/XMLSchema">
    <xss:import namespace="http://www.opengis.net/gml"
    schemaLocation="http://schemas.opengis.net/gml/2.1.2/feature.xsd" />
    <xss:import namespace="http://www.opengis.net/wfs"
    schemaLocation="http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd" />
    <xss:element name="USA" type="miwfs:USA_Type"
    substitutionGroup="gml:_Feature" />
    - <xss:complexType name="USA_Type">
    - <xss:complexContent>
    - <xss:extension base="gml:AbstractFeatureType">
    - <xss:sequence>
    - <xss:element name="Obj" minOccurs="0" maxOccurs="1">
    - <xss:complexType>
    - <xss:sequence>
      <xss:element ref="gml:_Geometry" />
    </xss:sequence>
    </xss:complexType>
    </xss:element>
    <xss:element name="State" nillable="false" type="xs:string" minOccurs="0"
    maxOccurs="1" />
    <xss:element name="State_Name" nillable="false" type="xs:string"
    minOccurs="0" maxOccurs="1" />
    </xss:sequence>
    </xss:extension>
    </xss:complexContent>
  </xss:complexType>
</schema>
```

GetFeature

知道可用的图元类型及其属性之后，检索 WFS 图元信息的最后一步便是发送 **GetFeature** 请求。该请求指定要获取哪些图元和属性。为了检索图元的某个子集，请使用筛选操作以便从空间和非空间的角度限制查询。请参阅 [代码示例：使用筛选器请求图元](#)。

GetFeature 支持通过 HTTP POST 执行。

在 OGC 支持的参数中，**Request** 和 **Typename** 是必需的。**Propertyname**、**Filter**、**FeatureID** 及其他参数为可选。指定 **FeatureID** 时，**Typename** 为可选。有关详细信息，请参阅 OGC Web 图元服务实现规范。

以下是针对名为 USA 的图元类型的 **GetFeatureResponse** 样例的一部分。

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
- <wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml" xmlns="http://www.mapinfo.com/wfs"
  xsi:schemaLocation="http://www.opengis.net/wfs
  http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd
  http://www.mapinfo.com/wfs
  http://localhost/Wfs/GetFeature.ashx?REQUEST=DescribeFeatureType&SERVICE=
  WFS&VERSION=1.0.0&TYPENAME=miwfs:USA">
  - <gml:boundedBy>
    <gml:null>inapplicable</gml:null>
    </gml:boundedBy>
  - <gml:featureMember>
    - <USA>
      <State>AK</State>
      </USA>
    </gml:featureMember>
  - <gml:featureMember>
    - <USA>
      <State>AL</State>
      </USA>
    </gml:featureMember>
  - <gml:featureMember>
    - <USA>
      <State>AR</State>
      </USA>
    </gml:featureMember>
  - <gml:featureMember>
    - <USA>
      <State>AZ</State>
      </USA>
    </gml:featureMember>
  - <gml:featureMember>
    - <USA>
      <State>CA</State>
      </USA>
    </gml:featureMember>
  - <gml:featureMember>
    - <USA>
      <State>CO</State>
      </USA>
    </gml:featureMember>
  ...
</wfs:FeatureCollection>
```

配置 WFS 服务器

如果有空间数据，并且希望其他用户也可以访问该数据，则必须首先配置 WFS 服务器，使之描述您所提供的数据及其功能。为此，您需要提供两个配置文件。这里假定您有架构和 Web 服务方面的实践经验。

如果您只想访问其他用户的 WFS 服务器以检索图元，则无需配置 WFS 服务器。请参阅通过编程方式使用 MapXtreme WFS 客户端。

本节介绍主要配置步骤，如下所示：

步骤 1：创建或修改 Web.config 文件以包含 MapXtreme 特定的 WFS 信息以及 IIS 经典或集成管线模式的正确句柄。

步骤 2：创建有效的 WFS 服务器配置文件，该文件包含与待托管数据有关的信息。为避免在运行 WFS 服务器时出错，必须根据 WFS 架构文件 (MXP_WFSConfiguration_1_0.xsd) 对该文件进行校验。

步骤 3：配置并测试 WFS 服务器的设置。提供了 IIS7 和 IIS6 配置说明。

在 MapXtreme 产品介质上，我们提供 Web.config 和 WFS 配置文件示例，您可以参考它们来创建自己的配置文件。Web.config 文件定义 ASP 进程的处理方式。WFSSample.xml 定义希望 WFS 服务器提供的数据源和图元定义。

MapXtreme Workspace 和 WFS 服务器的架构也位于产品介质中。

步骤 1：创建 Web.config 文件

Web.config 是一个标准的 Web 应用程序配置文件。要使用它来配置 MapXtreme WFS 服务器，您必须对其进行编辑，以便提供 MapXtreme 特定的 WFS 信息并定义 ASP.NET 进程的处理方式。

1. 创建一个用于存放 Web.config 和 WFSSample.xml 的文件夹。本例中，文件夹的位置为 c:\wfs。
2. 从 MapXtreme 产品介质中将 Web.config 和 WFSSample.xml 复制到该文件夹。
3. 在文本编辑器中打开 Web.config 并修改 <appSettings> 行使之指向该 WFS 配置文件。

```
<configuration>
<appSettings>
<add key="configFile" value="C:\wfs\WFSSample.xml" />
```

4. 对于 IIS 7 经典模式，请更新系统（下面的粗体类型）上安装的 MapInfo.Wfs.Server 和 MapInfo.CoreEngine 程序集的版本号和 PublicKeyToken（如果必要）。

程序集位于以下位置处：C:\Windows\Microsoft.NET\assembly\GAC_32 或 GAC_64。

```
<system.web>
  <httpHandlers>
    <add verb="GET, POST" path="*.ashx"
      type="MapInfo.Wfs.WfsHttpHandler, MapInfo.Wfs.Server, Version=7.3.0.140, Culture=neutral, PublicKeyToken=4ac3224575145b20" />
  </httpHandlers>
  <httpModules>
    <add type="MapInfo.Engine.WebSessionActivator,
MapInfo.CoreEngine, Version=7.3.0.140, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator" />
  </httpModules>
```

- 对于 IIS 7 集成管线模式, 请将以下部分复制到 web.config 中。您不必注释掉 <system.web> 部分以在集成管线模式下运行。不过, 如果您需要在 IIS 7 经典模式下运行, 必须注释掉此 <system.webServer> 部分。

遵循步骤 4 中的注释以更新 MapInfo.CoreEngine 和 MapInfo.Wfs.Server 的程序集版本。

```
<system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <directoryBrowse enabled="true" />
    <modules>
        <add type="MapInfo.Engine.WebSessionActivator,
MapInfo.CoreEngine, Version=7.3.0.140, Culture=neutral,
PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator"/>
    </modules>
    <handlers>
        <add name="WFSHandler" verb="GET, POST" path="*.ashx"
type="MapInfo.Wfs.WfsHttpHandler, MapInfo.Wfs.Server, Version=7.3.0.140,
Culture=neutral, PublicKeyToken=4ac3224575145b20" />
    </handlers>
</system.webServer>
```

- 保存该文件并将其复制到在步骤 1 中创建的位置。

步骤 2: 为待托管的图元创建有效的 WFS 配置文件

WFSSample.xml 是 MapXtreme 附带提供的 WFS 配置文件。该文件定义与 WFS 服务器有关的信息, 包括服务器名称、标题、摘要、指向该 WFS 服务器的 URL 以及希望托管的数据。

- 在文本编辑器中打开 WFSSample.xml 并修改 OnlineResource 行使之包含 WFS 服务器的 URL。将 localhost 更改为适合您的 WFS 服务器的设置。

```
<mxp-wfs:Service>
    <mxp-wfs:Name>Sample WFS Server</mxp-wfs:Name>
    <mxp-wfs:Title>Sample WFS Server</mxp-wfs:Title>
    <!-- The following is the URL of the WFS server -->
    <mxp-wfs:OnlineResource>http://localhost/wfs/GetFeature.ashx</mxp-
wfs:OnlineResource>
</mxp-wfs:Service>
```

您可以根据需要修改其他元素, 包括服务器名称、标题、摘要、费用、访问限制, 等等。

- 通过为每个表创建一个 Table 元素注册该 WFS 服务的表。<mxp-wfs:DataSourceDefinition> id 的值必须与 <mxp-wfs:Name> 匹配。可以按任意顺序包含各个表。
- 在 <DataSourceDefinitionSet> 中, 修改 MYPATH 变量使之反映指向您的数据的实际路径。

以下是标识两个图元表的 WFSSample.xml 的一部分。

```
<mxp-wfs:Table>
    <mxp-wfs:Name>USA</mxp-wfs:Name>
    <mxp-wfs:Title>Title for usa</mxp-wfs:Title>
    <mxp-wfs:Abstract>Abstract for USA</mxp-wfs:Abstract>
    <mxp-wfs:Keywords>Keywords for USA</mxp-wfs:Keywords>
<mxp:DataSourceDefinitionSet>
    <mxp:TABFileDataSourceDefinition id="USA">
```

```
<mxp:DataSourceName>USA</mxp:DataSourceName>
<mxp:FileName>MYPATH\USA.TAB</mxp:FileName>
</mxp:TABFileDataSourceDefinition >
</mxp:DataSourceDefinitionSet>
</mxp-wfs:Table>
<mxp-wfs:Table>
<mxp-wfs:Name>US_HIWAY</mxp-wfs:Name>
<mxp-wfs:Title>Title for US_HIWAY</mxp-wfs:Title>
<mxp-wfs:Abstract>Abstract for US_HIWAY</mxp-wfs:Abstract>
<mxp-wfs:Keywords>Keywords for US_HIWAY</mxp-wfs:Keywords>
<mxp:DataSourceDefinitionSet>
    <mxp:TABFileDataSourceDefinition id="US_HIWAY">
        <mxp:DataSourceName>US_HIWAY</mxp:DataSourceName>
        <mxp:FileName>MYPATH\US_HIWAY.TAB</mxp:FileName>
    </mxp:TABFileDataSourceDefinition >
</mxp:DataSourceDefinitionSet>
</mxp-wfs:Table>
```

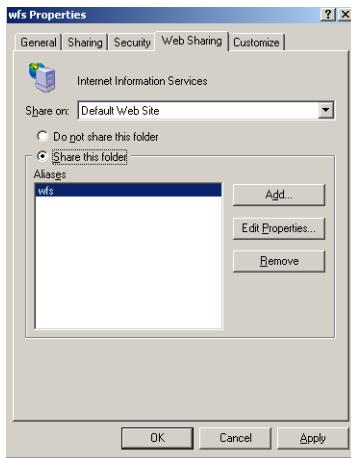
4. 如果由 WFS 托管 RDBMS 上存储的数据, 请按照以下示例指定 `<ConnectionSet>` 和 `<ConnectionMember>` 元素:

```
<mxp:ConnectionSet>
    <mxp:DBConnection dbType="sqlserver">
        <mxp:ConnectionName>sqlserver2k</mxp:ConnectionName>
        <mxp:ODBCConnectionString>DRIVER={ SQL
Server } ;DATABASE=YOURDB;Server=YOURSERVER;UID=YOURUSER;PWD=YOURPASSWORD;Q
uotedID=No;Trusted_Connection=No;Network=DBMSSOCN;Address=YOURSERVER, YOUR
SERVERPORT</mxp:ODBCConnectionString>
    </mxp:DBConnection
</mxp:ConnectionSet>
...
<mxp-wfs:Table>
    <mxp-wfs:Name>MySQLServerTable</mxp-wfs:Name>
    <mxp-wfs:Title>Title for MySQLServerTable</mxp-wfs:Title>
    <mxp-wfs:Abstract>Abstract for MySQLServerTable</mxp-
wfs:Abstract>
    <mxp-wfs:Keywords>Keywords for MySQLServerTable</mxp-
wfs:Keywords>
    <mxp:DataSourceDefinitionSet>
        <mxp:DBDataSourceDefinition id="MySQLServerTable">
            <mxp:DataSourceName>MySQLServerTable</mxp:DataSourceName>
            <mxp:ConnectionMember>
                <mxp:ConnectionName>my_sqlserver2000_advserver</mxp:ConnectionName>
                </mxp:ConnectionMember>
                <mxp:DBQuery>
                    <mxp:Query>select * from MySQLServerTable</mxp:Query>
                </mxp:DBQuery>
                <mxp:DBDataSourceMetadata/>
            </mxp:DBDataSourceDefinition>
        </mxp:DataSourceDefinitionSet>
    </mxp-wfs:Table>
```

步骤 3：配置和测试 WFS 服务器

在编辑 Web.config 和 WFSSample.xml 文件之后，必须使用 Internet 信息服务注册 WFS 服务器，最后还必须测试 WFS 服务器的设置。

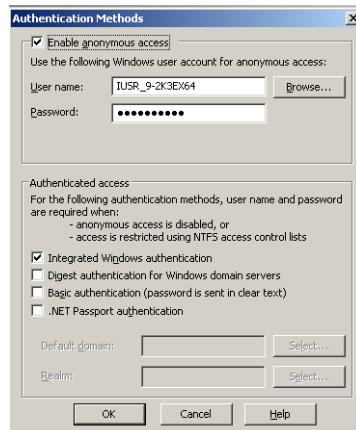
1. 右键单击 WFS 文件夹（例如 c:\wfs）并选择“共享和安全”。在“Web 共享”选项卡上，选择“共享这个文件夹”单选按钮。如果希望为 Web 服务器设置别名，请单击“添加”按钮。



2. 打开 IIS（“开始”菜单 > 控制面板 > 管理工具 > Internet 信息服务）。展开默认的 Web 站点并按文件夹名称或按别名（若已设置别名）定位您的 WFS 服务器。
3. 右键单击该 Web 站点并选择“属性”。在 ASP.NET 选项卡下，从 ASP.NET 版本下拉列表中选择 4.0.30319（MapXtreme 程序集在 4.0 Framework 下编译）。



4. 仍在刚才的“属性”对话框中，在“目录安全性”选项卡下，单击右上方的“编辑”按钮。在“身份验证方法”对话框中，选中“匿名访问”复选框。这样，WFS 服务的用户可以跳过用户名/密码身份验证过程。单击两次“确定”，关闭 IIS 窗口。



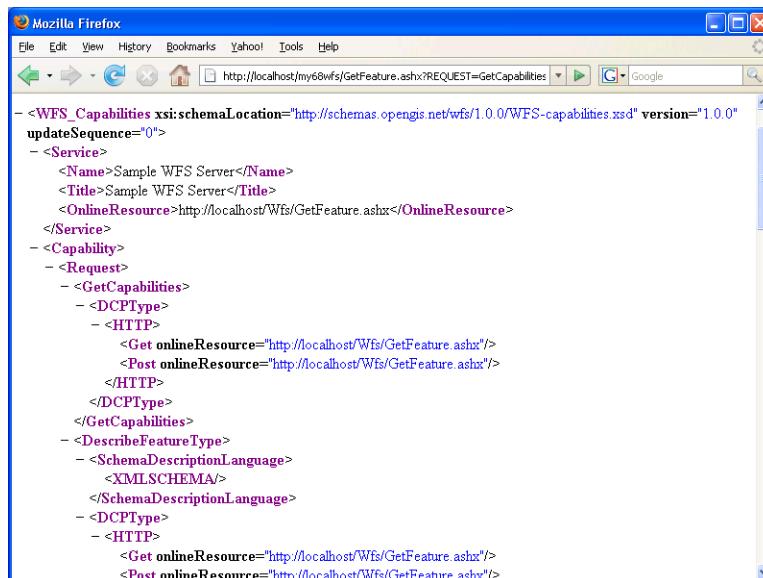
5. 通过从 Web 浏览器发送 **GetCapabilities** 请求，测试您的设置。在地址框中键入：

`http://localhost/wfs/GetFeature.ashx?REQUEST=GetCapabilities&SERVICE=WFS&VERSION=1.0.0`

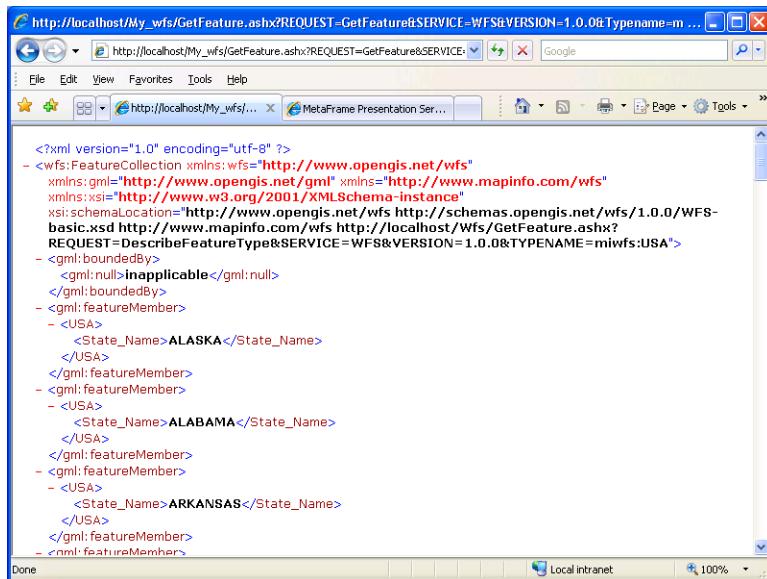
使用您的 Web 服务器代替 localhost。如果已为 Web 服务器设置别名，请务必在 URL 中包含该别名。例如：

`http://localhost/My_WFS/GetFeature.ashx?REQUEST=GetCapabilities&SERVICE=WFS&VERSION=1.0.0`

测试成功将返回类似下图的网页。若未返回功能列表，请检查您的配置文件以确保您的输入正确无误。既然是手动创建配置文件，自然很容易出现打字错误或丢失标记。



6. 对于返回的图元类型，要了解与其属性有关的信息，请发送 **DescribeFeatureType** 请求：
`http://localhost/My_wfs/GetFeature.ashx?REQUEST=DescribeFeatureType&SERVICE=WFS&VERSION=1.0.0&Typename=miwfs:USA`
该请求返回对图元类型 USA 的属性的描述，包括两个列名：State 和 State_Name。
7. 要请求查询 USA 表的图元信息，请发送 **GetFeature** 请求。
`http://localhost/My_wfs/GetFeature.ashx?REQUEST=GetFeature&SERVICE=WFS&VERSION=1.0.0&Typename=miwfs:USA&propertyname=miwfs:State_Name`
该请求以 **FeatureCollection** 的形式返回 USA 表的图元。注意：在该 URL 中，我们只请求了 USA 表两个列属性中的一个：State_Name。如果未指定任何属性名称，则返回表中所有属性。



通过编程方式使用 MapXtreme WFS 客户端

用户可以通过编程方式使用 MapXtreme 附带的 WFS 客户端，访问符合 OGC 标准的 Web 图元服务的数据。MapXtreme WfsClient 提供与任何符合 WFS 1.0.0 标准的服务器进行 API 级交互的能力。WfsClient 和相关类位于 `MapInfo.Wfs.Client` 命名空间并使用 OGC 规范的 WFS Basic 部分。使用 `WfsClient` 类，可以在任何符合 WFS 1.0.0 标准的服务器上调用以下方法：**GetCapabilities**、**DescribeFeatureType** 和 **GetFeature**。

```
public void MapInfo_Wfs_Client_GetCapabilities1(string Url) {
    WfsCapabilities capabilities =
    WfsClient.GetCapabilities(RequestMethod.GET, Url);
    FeatureTypeList featureTypeList = capabilities.FeatureTypeList;
    IList featureTypes = featureTypeList.FeatureTypes;
    foreach (FeatureType featureType in featureTypes) {
        System.Console.WriteLine("FeatureType: " + featureType.Name);
```

```
    }  
}
```

在 WFS 查询中使用筛选器

可以使用 **IFilter** 接口将筛选器应用到 **GetFeature** 请求。筛选器可以是空间的或非空间的（标量）。通过 **Bbox** 和 **Within** 之类的空间查询，可以获取特定区域中存在的图元。通过标量筛选器，可以查询某个图元类型的特定属性。也可以使用组合筛选器，例如 **AND** 和 **OR**。

MapXtremeWFS 支持以下筛选操作。

- 空间运算符： **BBox**、 **Equals**、 **Disjoint**、 **Intersects**、 **Within**、 **Contains**
- 非空间比较运算符：逻辑运算符：**AND**、 **OR** 和 **NOT**； **PropertyIsEqualTo**、
PropertyIsGreater Than、 **PropertyIsGreater ThanOrEqual To**、 **PropertyIsLess Than**、
PropertyIsLess ThanOr Equal To、 **PropertyIsNotEqual To**、
- 简单算术运算符：**Add**、 **Div**、 **Mul**、 **Sub**

i HTTP 协议规定 **GetFeature** 请求的 URL 长度不得超过 2048 个字符。切记：筛选器创建的 URL 很容易大大超出此长度限制，从而导致异常。

代码示例：使用筛选器请求图元

以下 C# 代码示例说明了如何对 WFS 数据应用筛选器以便只返回您需要的数据。本例中，我们查询 WFS 服务器以了解有哪些可用的图元，然后请求服务器只返回满足以下条件的图元： **WorldCap** 的 **population** 列的值为 1,000,000 或以上。 **GetCapabilities** 和 **GetFeatures** 方法的输出显示在代码示例后面。

```
private void doWFS()  
{  
    string wfsUrl = @"http://localhost/MXTWFS/GetFeatures.ashx";  
  
    /// first we need to get the list of capabilities and list out the tables  
    /// available  
    MapInfo.Wfs.Client.WfsCapabilities capabilities =  
        MapInfo.Wfs.Client.WfsClient.GetCapabilities(  
            MapInfo.Wfs.Client.RequestMethod.GET, wfsUrl);  
    MapInfo.Wfs.Client.FeatureTypeList featureTypeList =  
        capabilities.FeatureTypeList;  
    IList featureTypes = featureTypeList.FeatureTypes;  
    foreach (MapInfo.Wfs.Client.FeatureType featureType in  
featureTypes)  
    {  
        System.Console.WriteLine("FeatureType: " + featureType.Name);  
    }  
  
    /// Now we can look at a specific table to see its contents  
    string featureTypeName = "miwfs:WorldCap"; // Name taken from  
        /// the above output
```

```
MapInfo.Data.MultiFeatureCollection mfc =
    MapInfo.Wfs.Client.WfsClient.GetFeature(wfsUrl,
        new string[] { featureTypeName }, null, "GML3", -1, null);
    DisplayFeatureCollection(mfc[0]);

    /// Now we can apply a filter
    MapInfo.Wfs.Client.IFilter filter = new
        MapInfo.Wfs.Client.PropertyIsGreaterThanOrEqualTo(
            new MapInfo.Wfs.Client.PropertyName(wfsUrl, "CAP_POP"),
            new MapInfo.Wfs.Client.Literal("1000000"));

/// Create the Query container
///
    IList queries = new MapInfo.Wfs.Client.Query[] {
        new MapInfo.Wfs.Client.Query(new
            MapInfo.Wfs.Client.TypeName("http://www.mapinfo.com/wfs",
                "WorldCap"), null, filter)};

/// Run the filter and return the subset.
///
    mfc = MapInfo.Wfs.Client.WfsClient.GetFeature(wfsUrl,
        queries, "GML3", -1);
    DisplayFeatureCollection(mfc[0]);
}
```

上述 GetCapabilities 代码的输出为：

```
FeatureType: miwfs:Ocean
FeatureType: miwfs:WldCty25
FeatureType: miwfs:World
FeatureType: miwfs:WorldCap
```

第一个 GetFeature 调用返回 WorldCap 中的所有图元（行），下面显示了其中的部分图元。第一行显示 WorldCap 数据列，其他行显示各行的数据。

```
Obj, Capital, Country, Cap_Pop, MI_Style,
Point,Abidjan,IVORY COAST,2700000,MapInfo.Styles.SimpleVectorPointStyle:
12 point,
Point,Abu Dhabi,UNITED ARAB
EMIRATES,722000,MapInfo.Styles.SimpleVectorPointStyle: 12 point,
Point,Accra,GHANA,949000,MapInfo.Styles.SimpleVectorPointStyle: 12 point,
Point,Addis Ababa,ETHIOPIA,1423111,MapInfo.Styles.SimpleVectorPointStyle:
12 point,
Point,Agana,GUAM,132726,MapInfo.Styles.SimpleVectorPointStyle: 12 point,
Point,Algiers,ALGERIA,1483000,MapInfo.Styles.SimpleVectorPointStyle: 12
point,
Point,Alma-ata,KAZAKHSTAN,1108000,MapInfo.Styles.SimpleVectorPointStyle:
12 point,
Point,Alofi,NIUE,3300,MapInfo.Styles.SimpleVectorPointStyle: 12 point,
Point,Amman,JORDAN,936000,MapInfo.Styles.SimpleVectorPointStyle: 12
point,
Point,Amsterdam,NETHERLANDS,694656,MapInfo.Styles.SimpleVectorPointStyle:
12 point,
...
...
```

使用筛选的 `GetFeature` 调用，则仅返回 `PropertyIsGreaterThanOrEqualTo`（人口）为 1,000,000 的行。注意：结果中不含 Abu Dhabi，因为其人口为 722,000。

```
Obj, Capital, Country, Cap_Pop, MI_Style,  
Point,Abidjan,IVORY COAST,2700000,MapInfo.Styles.SimpleVectorPointStyle:  
12 point,  
Point,Addis Ababa,ETHIOPIA,1423111,MapInfo.Styles.SimpleVectorPointStyle:  
12 point,  
Point,Algiers,ALGERIA,1483000,MapInfo.Styles.SimpleVectorPointStyle: 12  
point,  
Point,Alma-ata,KAZAKHSTAN,1108000,MapInfo.Styles.SimpleVectorPointStyle:  
12 point,  
Point,Ankara,TURKEY,2553000,MapInfo.Styles.SimpleVectorPointStyle: 12  
point,  
Point,Baghdad,IRAQ,3400000,MapInfo.Styles.SimpleVectorPointStyle: 12  
point,  
Point,Baku,AZERBAIJAN,1115000,MapInfo.Styles.SimpleVectorPointStyle: 12  
point,  
...  
...
```

使用空间运算符进行筛选

以下示例说明如何在 `GetFeature` 请求中使用空间查询。本例中，查询 Europe 表中所有满足指定最小边界矩形的行。这可以使用大多数 WFS 服务器都支持的常用空间运算符 `Bbox` 来实现。

```
/// Now lets find rows in an MBR  
  
// WGS84 - Europe  
MapInfo.Geometry.DRect bbox = new MapInfo.Geometry.DRect(-  
11.69, 35.36, 48.77, 65.05);  
mfc = MapInfo.Wfs.Client.WfsClient.GetFeature(  
wfsUrl,  
new string[] { featureTypeName },  
bbox,  
null,  
"GML3",  
-1);  
DisplayFeatureCollection(mfc[0]);
```

根据 WFS 响应创建地图图层

如果希望将 WFS 图元数据作为地图图层引入 `MapXtreme` 以便进一步分析，则需要运行分析 GML 的代码并创建一个 `MultiFeatureCollection`。

分析 WFS 响应

分析响应未在 WFS 规范中指定，该规范也没有描述 GetFeature 请求结果的准确格式。WFS 规范只规定请求至少必须符合 GML2 标准。由于 GML2 有许多版本，要让 MapXtreme WFS 客户端能够正确地转换来自 WFS 服务器的 GML2 响应，必须创建一个分析器，将 GML2 GetFeature 响应转换为 MapXtreme MapInfo.Data.MultiFeatureCollection。这可以通过实现 IWfsReader 接口并使用 MapInfo.Wfs.Client.WfsReaderFactory 类注册该接口实现来完成。注册必须出现在每个 URL 的基础上，也就是说，如果用户需要和两个具有相同 GetFeature 响应的 WFS 服务器交互，则对于每个服务器 URL 特定的 IWfsReader 实现必须注册两次。无需实现整个接口。可以扩展 MapInfo.Wfs.Client.WfsReader 类以重载 ReadGetFeature(XmlReader) 方法。与不兼容的 WFS 服务器交互时，任何其它方法都可以被重载。

以下代码示例说明了从请求图元到创建地图图层并在地图中显示图元的完整过程。此代码示例包含在 WfsClient 示例应用程序中，后者位于 MapXtreme 安装目录的 \Samples 文件夹中。

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Xml.Schema;

using MapInfo.Data;
using MapInfo.Engine;
using MapInfo.Mapping;
using MapInfo.Wfs.Client;

namespace MapInfo.Wfs.Client.Samples {

    /// <summary>
    /// Simple sample to demonstrate how to register a WfsReader to handle
    /// requests from a specific Wfs server, get the capabilities of the
    /// server,
    /// get the schema for a feature type located on the server and getting
    /// all
    /// of the features from the server.
    /// </summary>
    class SimpleSample {
        private const string URL = "http://www.mapinfo.com/miwfs";

        private const string defaultExportFileName = "WfsClient.gif";

        [STAThread]
        static void Main(string[] args) {

            string exportFileName = defaultExportFileName;
            if (args.Length > 0 && args[0] != null && args[0].Trim().Length
                != 0) exportFileName = args[0];

            // register URL with a specific WFS reader
            WfsReaderFactory.RegisterHandler(URL, typeof(WfsReader));
        }
    }
}
```

```
// Get the WFS capabilities of the WFS server using the HTTP GET
method.
try
{
    // Get the WFS capabilities of the WFS server using the HTTP
    // GET method.
    WfsCapabilities Capabilities =
    WfsClient.GetCapabilities(RequestMethod.GET, URL);
}
catch
{
    MessageBox.Show("Please check if " + URL + " is a valid WFS
URL");
    return;
}

// Do something with the the WfsCapabilities here...

// Get the schema for the USA feature type
string[] TypeNames = new string[] { "miwfs:USA" };

// Do something with the schema here...
XmlSchema usaSchema = WfsClient.DescribeFeatureType(URL,
TypeNames);

// Get all features from the USA feature type
MultiFeatureCollection usa = WfsClient.GetFeature(URL, TypeNames,
null, null, -1, null);
IFeatureCollection fc = usa[0];

// iterate over the Usa MultiFeatureCollection and add each
// IFeatureCollection to a MemTable, etc...
TableInfoMemTable memTableInfo = new
TableInfoMemTable("myMemTable");
foreach (Column c in fc.Columns) {
    memTableInfo.Columns.Add(c);
}
Table memTable =
Session.Current.Catalog.CreateTable(memTableInfo);
memTable.InsertFeatures(fc);

// create a layer from the MemTable
FeatureLayer featureLayer = new FeatureLayer(memTable);

// create the map and add the layer
Map map = Session.Current.MapFactory.CreateEmptyMap(new Size(500,
500));
map.Layers.Add(featureLayer);

// export the map to a file
using (MapExport mx = new MapExport(map))
```

```
{  
    mx.Format = ExportFormat.Gif;  
    mx.Export(exportFileName);  
}  
  
// clean up the map  
Session.Current.MapFactory.Remove(map);  
}  
}  
}
```

Web 地图服务

MapXtreme 提供部署和访问 Web 地图服务 (WMS) 的能力，以便您能够通过各种源获取 WMS 地图或向其他用户提供地图。

在本章中：

◆ MapXtreme Web 地图服务简介	377
◆ 理解 WMS 操作.....	377
◆ 代码示例：请求 WMS 图层.....	379
◆ WMS 和坐标系	380
◆ MapXtreme WMS 和身份验证	381
◆ 设置 MapXtreme WMS 服务器.....	382
◆ 配置 WMS 服务器的图层信息	390

MapXtreme Web 地图服务简介

MapXtreme Web 地图服务 (WMS) 使客户端可以请求，服务器可以传递空间参考地图图像，这些操作可以通过因特网或专有的内联网完成。MapXtreme 提供的工具可以部署 WMS Server 供他人查询，同时支持 WMS 客户端功能与应用程序的结合，实现从 WMS Server 请求图像。WMS 服务器和客户端实现以 1.3.0 OpenGIS® Web Map Service 实现规范为基础，后者位于 www.opengis.org。MapXtreme WMS 服务器符合 1.3.0 和 1.1.1 Web 地图服务的兼容性要求。

 有关 WMS 1.3.0 支持的详细信息，请参阅[MapXtreme 发行说明](#)。

基本的 WMS 将保存到图层中的几何引用信息进行分类并提供显示这些图层的预定义样式。符合 OGC 规范的 WMS 还可以支持某些图像格式的透明像素定义。透明像素使用户可以将 WMS 图像用作光栅覆盖而不是单独地作为地图的背景图层。可用数据的数量和质量由具体的 WMS 服务器决定。

由政府、公司和其他组织提供的可用空间几何信息量正不断增长，所以用户可以检索它们以增强地图的精确性和完整性。需要使用陆地和水文数据的公司可以从美国人口调查局或当地数据提供商获取海拔和人口信息。将交通图案数据和存储位置信息组合在一起，可以提供建立附加存储位置的洞察力或优化市场和产品放置成本。作为开发人员，可以通过扩展 WMS 代码来定制包括特定旅馆、名胜或休假目的地的路径请求。用户需要并且能够想到的功能都可以实现。

理解 WMS 操作

MapXtreme WMS 遵循适用于基本 WMS 服务的 1.3.0 和 1.1.1 OGC 规范。有三种 WMS 操作：

- GetCapabilities
- GetMap
- GetFeatureInfo

GetCapabilities

在请求 WMS 地图之前，必须先查找可用图层的名称、正在使用的样式、使用的空间信息以及 WMS 服务器提供的其他信息。GetCapabilities 是一种 HTTP 请求，通过 Internet 或内联网检索服务级元数据，包括服务器名称、图层名称、数据摘要和可接受的请求参数。

在 MapXtreme 中，GetCapabilities 是从 WMS 服务器收集信息的第一步。随后会使用此功能来请求地图图像。

GetMap

知道 WMS 服务器的功能之后，即可发出 GetMap 请求，向一个或多个 WMS 服务器请求地图图层的地图图像。根据 WMS 服务器的功能，GetMap 请求可能包括以下内容。

- 版本 - 请求版本

- 图层 - 一个或多个地图图层数据
- 样式 - 显示渲染图层所用的样式。若未指定，则使用默认样式。
- 边界框 - 地图图像中要包含的区域
- 适用于 WMS 1.3.0 的 CRS 或 SRS (WMS 1.1.1) 命名空间标识符形式的坐标参考系。
- 输出格式 - 输出地图图像所用的 MIME 类型，例如 GIF 和 PNG
- 输出大小 - 地图图像的高度和宽度（单位为像素）
- 背景色 - 十六进制的红绿蓝颜色值，例如 0xFFFFFFFF（透明度为 true 时需要）
- 透明度 - true/false 设置，指示图层是否可用作其他图层的透明覆盖。

GetFeatureInfo

成功完成 GetMap 请求后，用户可能要了解有关地图中所包括图元的详细信息。GetFeatureInfo 操作返回关于地图中图层的信息和每个图层的可查询属性。此操作由 WMS 服务器控制；不过，服务器可能不提供此功能。

使用 MapXtreme 作为 WMS 客户端

您可以通过 MapInfo.Data.TableInfoWMS 使用 MapXtreme WMS 客户端通过编程的方式请求 WMS 地图图像：此部分提供的代码片段介绍了如何使用 MapXtreme 作为 WMS 客户端。此外，完成安装后，WMSPreview 示例应用程序位于 Desktop samples 文件夹中。

GetCapabilities

在 MapXtreme 中，不能直接调用 GetCapabilities。对地图的请求包含在使用 WMSClient 的调用中，该调用接受功能实例作为输入。

```
// build the capabilities
ICapabilities capabilities = WmsClientUtilities.GetCapabilities
    (url, "1.3.0");

// create the WMS client
WmsClient wmsClient = new WmsClient(capabilities);
wmsClient.AddLayer("WORLD");
wmsClient.Srs = "EPSG:4326";
wmsClient.BGColor = Color.Blue;
wmsClient.MimeType = "image/gif";
```

您必须提供要访问的服务器的 URL。版本为可选项。MapXtreme 客户端会在 HTTP 请求中的 ? 之后追加必需的信息以生成服务器功能的完整请求。例如：

```
http://www.mapsanddata.xyz/gis/services/maps/hydrography/MapServer/WMS
ver?request=GetCapabilities&service=WMS&version=1.3.0
```

如果不指定版本且正访问的 MapXtreme WMS 服务器同时支持二者，则将返回 1.3.0 的响应。

可通过重载方法来获取版本号的字符串数组。版本的顺序取决于所查找的内容。返回第一个成功匹配项。

GetCapabilities 还支持在请求中使用用户定义的参数。使用获取 **NameValueCollection** 的方法。详细信息, 请参阅《开发人员参考》。

GetCapabilities 响应将返回到 XML 文档中, **MapXtreme** 将读取该文档并会从中创建功能对象。

GetMap

当放置将请求表示为 **MapXtreme** 地图中图层的表对象时, **MapXtreme** 负责调用 **GetMap**。下面的代码示例以 **GetCapabilities** 节中的示例为基础。

```
// create the table info
TableInfoWms wmsTableInfo= new TableInfoWms
    ("MyWmsTable", wmsClient);

// create the table
Table wmsTable = Session.Current.Catalog.OpenTable(wmsTableInfo);

// creates a FeatureLayer from the table entry
FeatureLayer featLyr = new FeatureLayer(wmsTable);
```

有关 **TableInfoWms** 的详细信息, 请参阅[第 153 页中的支持的表类型](#)中的 WMS。另请参阅《开发人员参考》中的 **TableInfoWms** 类。

GetFeatureInfo

GetFeatureInfo 请求的输入为地图和像素坐标的边界, 用户使用 **Info** 工具在此边界处单击地图。

```
// Get the feature info

Byte[] info = WmsClientUtilities.GetFeatureInfo(new DRect(45.0, 45.0, 90,
90), 640, 480, new String[] {"WORLD"}, new Point(300,200), "text/xml"));

MemoryStream memoryStream = new MemoryStream(byteArray);
memoryStream.Seek(0, SeekOrigin.Begin);
XmlDocument doc = new XmlDocument();
doc.Load(memoryStream);
//parse the xml doc as desired.
```

代码示例: 请求 WMS 图层

要通过编程的方式使用 **MapXtreme** 请求 WMS 图层, 请参照以下代码示例操作。

i WMS 的 **MapXtreme** 实现限制了返回图像的大小, 宽和高都限制为 4000 像素。

```
// build the capabilities
ICapabilities capabilities = WmsClientUtilities.GetCapabilities
    (url, "1.1.1");
```

```
// create the WMS client
WmsClient wmsClient = new WmsClient(capabilities);
wmsClient.AddLayer("WORLD");
wmsClient.Srs = "EPSG:4326";
wmsClient.BGColor = Color.Blue;
wmsClient.MimeType = "image/gif";

// create the table info
TableInfoWms wmsTableInfo= new TableInfoWms
    ("MyWmsTable", wmsClient);

// create the table
Table wmsTable = Session.Current.Catalog.OpenTable(wmsTableInfo);

// creates a FeatureLayer from the table entry
FeatureLayer featLyr = new FeatureLayer(wmsTable);
```

WMS 和坐标系

MapXtreme 提供了对三个 CRS 颁发机构坐标系的支持，可以在 **GetCapabilities** 请求中返回这些坐标系。

- CRS:84 - 经度/纬度 WGS 84。这等同于 WMS 1.1.1 中的 EPSG:4236
- CRS:83 - 经度/纬度 NAD 83。这等同于 WMS 1.1.1 中的 EPSG:4269
- CRS:27 - 经度/纬度 NAD 27。这等同于 WMS 1.1.1 中的 EPSG:4267

此外，MapXtreme API 提供了公共方法 **RegisterCRSCode()**，用于在 CRS codespace 中注册其它经度/纬度投影。

地图和图像边界

EX_GeographicBoundingBox

EX_GeographicBoundingBox 是 WMS 1.3.0 的一个新参数，用于以小数度来定义图层的最小边界矩形。其作用是便于进行地理搜索，而不要求搜索引擎进行坐标转换。

EX_GeographicBoundingBox 的属性为 **westBoundLongitude**、**eastBoundLongitude**、**southBoundLatitude**、**northBoundLatitude**。提供 **EX_GeographicBoundingBox** 时与 WMS 服务器所支持的 CRS 无关。如果数据采用的不是本地地理坐标，则其可能为近似值。

WMS 1.1.1 中的等价参数为 **LatLongBoundingBox**，其在 EPSG:4326 经度/纬度中的属性为 **minx**、**miny**、**maxx** 和 **maxy**。

BoundingBox

WMS 1.3 服务器提交的每个图层都必须至少具有一个边界框。将按坐标系颁发机构所要求的顺序显示坐标。在 EPSG 和 CRS 代码的示例中，边界框坐标彼此顺序相反。

```
<BoundingBox CRS="EPSG:4326" minx="-59.100605" miny="-86.389389"  
maxx="16.755765" maxy="-32.336389"/>  
<BoundingBox CRS="CRS:84" minx="-86.389389" miny="-59.100605" maxx="-  
32.336389" maxy="16.755765"/>
```

这表示，对于 EPSG:4326，minx 为最南纬度，miny 为最西经度。对于 CRS:84，minx 为最西经度，miny 为最南纬度。

MapXtreme 还支持 resx resy 的可选属性以描述空间分辨率。

图像拉伸

WMS 要求从 GetMap 请求返回的图像对 BBOX 和图像尺寸 (WIDTH, HEIGHT) 参数进行响应。因此，返回的图像不会随意拉伸以适合边界框。

MapXtreme WMS 和身份验证

基本身份验证

当针对安全的 WMS 服务器生成服务器请求时，MapXtreme 的 WMS 客户端实现了对管理基本身份验证协议的支持。可以在发出服务请求前在 **WmsRequest** 对象上定义验证凭据，允许通过 API 对验证进行可编程控制。如果 **WmsRequest** 没有正确的凭据，并且客户端从服务器收到未授权的错误消息，将会显示提示用户输入凭据的 Windows 对话框。使用凭据对话框可以禁用，在此情况下，将直接返回验证异常以调用处理应用程序。

成功通过 WMS 服务器的身份验证时，会在客户端会话的生命周期中缓存服务器的凭据。客户端可生成该服务器的后续 WMS 请求，也可生成其它服务器的请求（带或不带身份验证），不必重新进行身份验证。

为了将身份验证集成到现有的 MapXtreme WMS 客户端实现中，已对 **WmsRequest UserDefinedParameters** 的语义进行了扩展以包含对一组已知身份验证参数的支持，该组参数可直接映射至 **WmsRequest** 对象模型上的身份验证属性集。直接或间接通过对对象模型设置身份验证属性，**UserDefinedParameters** 具有同样的效果。

有关详细信息，请参阅《开发人员参考》中的 **WmsRequest** 类。

设置 MapXtreme WMS 服务器

MapXtreme 提供的 WMS 服务器兼容 OGC WMS 1.3.0 和 1.1.1 规范。

要设置自己的 WMS 服务器，则必须配置与 IIS 连接的服务器并创建必要的 XML 文件以提供管理 Web 地图服务所需的数据连接。我们假定用户了解 WMS 和 MapXtreme 工作空间构架的基本知识。

WMS 服务器在 Microsoft Internet 信息服务 (IIS) 内运行。以下是设置 WMS 服务器的配置步骤。

步骤 1: 创建或修改 Web.config 文件以包含 MapXtreme 特定的 WMS 信息以及 IIS 经典或集成管线模式的正确句柄。

步骤 2: 创建有效的 WMS 服务器配置文件，该文件包含与待托管数据有关的信息。为避免运行 WMS 服务器时出错，必须根据 WMS 架构文件 (MXP_WMS_Configuration_1_2.xsd) 来校验该文件。

步骤 3: 配置并测试 WMS 服务器的设置。提供了 IIS7 和 IIS6 配置说明。

在 MapXtreme 产品介质上，我们提供 Web.config 和 WMS 配置文件示例，您可以参考它们并根据自己的需要进行修改。该 Web.config 文件定义 ASP.NET 过程的处理方式以及 WMS 服务器与 MapXtreme 之间的关系。WMSSample.xml 定义了数据源和图层定义，您可将其用作 WMS 服务器实现的模型。

步骤 1：创建 Web.config 文件

Web.config 是一个标准的 Web 应用程序配置文件。此处对该文件进行了修改，以便提供 MapXtreme 特定的 WMS 信息并定义 ASP.NET 过程的处理方式。

MapXtreme 支持经典和集成管线模式下的 IIS 7.x。

1. 创建一个用于存放 Web.config 和配置文件 WMSSample.xml 的文件夹。本例中，文件夹的位置为 **c:\wms**。从 MapXtreme 产品介质中将 Web.config 和 WMSSample.xml 复制到该文件夹。
2. 在文本编辑器中打开 Web.config 并修改 `<appSettings>` 行使之指向该 WMS 配置文件。

```
<configuration>
<appSettings>
    <add key="configFile" value="C:\wms\WMSSample.xml" />
```

3. 对于 IIS 7 经典模式，请更新系统（下面的粗体类型）上安装的 **MapInfo.Wms.Server** 和 **MapInfo.CoreEngine** 程序集的版本号和 PublicKeyToken（如果必要）。

程序集位于以下位置处: C:\Windows\Microsoft.NET\assembly\GAC_32 或 GAC_64。

```
<system.web>
    <httpHandlers>
        <add verb="GET, POST" path="*.ashx"
            type="MapInfo.Wms.WmsHttpHandler, MapInfo.Wms.Server, Version=7.3.0.140, Culture=neutral, PublicKeyToken=4ac3224575145b20" />
    </httpHandlers>
    <httpModules>
        <add type="MapInfo.Engine.WebSessionActivator,
MapInfo.CoreEngine, Version=7.3.0.140, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator" />
    </httpModules>
```

- 对于 IIS 7 集成管线模式, 请将以下部分复制到 web.config 中。您不必注释掉 <system.web> 部分以在集成管线模式下运行。不过, 如果您需要在 IIS 7 经典模式下运行, 必须注释掉此 <system.webServer> 部分。

遵循说明步骤 3 以更新 MapInfo.CoreEngine 和 MapInfo.Wms.Server 的程序集版本。

```
<system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <directoryBrowse enabled="true" />
    <modules>
        <add type="MapInfo.Engine.WebSessionActivator,
MapInfo.CoreEngine, Version=7.3.0.140, Culture=neutral,
PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator"/>
    </modules>
    <handlers>
        <add name="WMSHandler" verb="GET, POST" path="*.ashx"
type="MapInfo.Wms.WmsHttpHandler, MapInfo.Wms.Server, Version=7.3.0.140,
Culture=neutral, PublicKeyToken=4ac3224575145b20" />
    </handlers>
</system.webServer>
```

- 保存 web.config 文件, 并将其复制到在步骤 1 中创建的位置。

步骤 2: 为待托管的数据创建有效的 WMS 配置文件

WMSSample.xml 是 MapXtreme 附带提供的 WMS 配置文件。该文件定义与 WMS 服务器有关的信息, 包括服务器名称、标题、摘要、指向该 WMS 服务器的 URL 以及希望托管的数据。

- 在文本编辑器中打开 WMSSample.xml 并修改 <mfp-wms:OnlineResource> 行使之指向 WMS 服务器。

您可以根据需要修改其它元素, 包括服务器名称、标题、抽象、关键字列表和词汇、联系信息、费用以及访问约束。

```
<mfp-wms:Service>
    <mfp-wms:Name>Sample WMS Server</mfp-wms:Name>
    <mfp-wms:Title>Sample WMS Server</mfp-wms:Title>
    <mfp-wms:Abstract>This is a sample WMS server</mfp-wms:Abstract>
    <mfp-wms:KeywordList>
        <mfp-wms:Keyword vocabulary="ISO 19115:2003">biota</mfp-
wms:Keyword>
        <mfp-wms:Keyword vocabulary="ISO 19115:2010">biota</mfp-
wms:Keyword>
        <mfp-wms:Keyword>rivers</mfp-wms:Keyword>
    </mfp-wms:KeywordList>
    <!-- The following is the URL of your WMS server, here assume it is
localhost -->
    <mfp-wms:OnlineResource>http://localhost:port/WMS/GetMap.ashx</mfp-
wms:OnlineResource>
    <mfp-wms:Fees>$10</mfp-wms:Fees>
    <mfp-wms:AccessConstraints>none</mfp-wms:AccessConstraints>
</mfp-wms:Service>
```

2. 通过修改 `<DataSourceDefinitionSet>` 中的路径来注册数据图层。希望在 WMS 中使用的每个图层都需要一个注册条目。以下是查找示例 WMS 服务器中 TAB 文件的 WMSSample.xml 的一部分：

```
<!-- The following data sources reference local TAB files, you need  
to replace MYPATH with the real path to those tab files.-->  
<TABFileDataSourceDefinition id="id1" readOnly="false"  
xmlns="http://www.mapinfo.com/mxp">  
    <DataSourceName>STATES</DataSourceName>  
    <FileName>MYPATH\USA.TAB</FileName>  
    </TABFileDataSourceDefinition>  
<TABFileDataSourceDefinition id="id2" readOnly="false"  
xmlns="http://www.mapinfo.com/mxp">  
    <DataSourceName>US_HIWAY</DataSourceName>  
    <FileName>MYPATH\US_HIWAY.TAB</FileName>  
    </TABFileDataSourceDefinition>  
<TABFileDataSourceDefinition id="id5" readOnly="false"  
xmlns="http://www.mapinfo.com/mxp">  
    <DataSourceName>OCEAN</DataSourceName>  
    <FileName>MYPATH\OCEAN.TAB</FileName>  
    </TABFileDataSourceDefinition>
```

3. 在 `<mxp-wms:WMSLayer>` 部分中描述希望在 WMS 服务器上托管的图层。对于计划在 WMS 服务器上提供的每个图层，都需要为它建立一个条目。还可嵌套图层，这样就可以通过请求父图层将所有子图层包括在响应中。

以下示例是名为 **States** 的图层的对应条目。下面的粗体文本是为了支持 WMS 1.3.0 而添加的元素。

有关如何生成这些条目的更多详情，请参阅[配置 WMS 服务器的图层信息](#)。

```
<mxp-wms:WmsLayer>  
    <mxp-wms:Name>States</mxp-wms:Name>  
    <mxp-wms:Title>States</mxp-wms:Title>  
    <mxp:SRSName>EPSG:4326</mxp:SRSName>  
    <b><mxp-wms:Attribution type="FGDC:1998"></b>  
        <mxp-wms:Title>Attribution</mxp-wms:Title>  
        <mxp-wms:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:type="simple" xlink:href="http://localhost/metadata/roads.txt">  
        </mxp-wms:OnlineResource>  
        <mxp-wms:LogoURL width="500" height="600">  
            <mxp-wms:Format>text/plain</mxp-wms:Format>  
            <mxp-wms:OnlineResource  
xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"  
xlink:href="http://localhost/metadata/roads.txt">  
            </mxp-wms:OnlineResource>  
        </mxp-wms:LogoURL>  
    </mxp-wms:Attribution>  
    <b><mxp-wms:MetadataURL type="FGDC:1998"></b>  
        <mxp-wms:Format>text/plain</mxp-wms:Format>  
        <mxp-wms:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:type="simple" xlink:href="http://localhost/metadata/roads.txt" />  
        </mxp-wms:MetadataURL>  
        <mxp-wms:MetadataURL type="ISO19115:2003">  
            <mxp-wms:Format>text/xml</mxp-wms:Format>
```

```
<mfp-wms:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:type="simple" xlink:href="http://localhost/metadata/roads.xml" />  
</mfp-wms:MetadataURL>  
<mfp-wms:FeatureListURL>  
    <mfp-wms:Format>text/xml</mfp-wms:Format>  
    <mfp-wms:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:type="simple"  
xlink:href="http://localhost/featurelist/feature1.xml" />  
</mfp-wms:FeatureListURL>  
<mfp-wms:FeatureListURL>  
    <mfp-wms:Format>text/plain</mfp-wms:Format>  
    <mfp-wms:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:type="simple"  
xlink:href="http://localhost/featurelist/feature2.xml" />  
</mfp-wms:FeatureListURL>  
<mfp-wms:WmsStyleSet>  
    <mfp-wms:WmsStyle>  
        <mfp-wms:Name>BlueFill</mfp-wms:Name>  
        <mfp-wms:Title>Blue Fill</mfp-wms:Title>  
        <mfp-wms:Abstract>This is a blue area fill with a red  
border.</mfp-wms:Abstract>  
        <mfp-wms:LegendURL width="100" height="100">  
            <mfp-wms:Format>image/gif</mfp-wms:Format>  
            <mfp-wms:OnlineResource  
xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"  
xlink:href="http://localhost/WMS/legends/1.gif" />  
        </mfp-wms:LegendURL>  
        <mfp-wms:LegendURL width="200" height="200">  
            <mfp-wms:Format>image/gif</mfp-wms:Format>  
            <mfp-wms:OnlineResource  
xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"  
xlink:href="http://localhost/WMS/legends/2.gif" />  
        </mfp-wms:LegendURL>  
        <AreaStyle xmlns="http://www.mapinfo.com/mfp">  
            <!-- The following defines the red border -->  
            <LineStyle stroke="red" width="1" width-unit="mapinfo:imagesize  
pixel">  
                <Pen>mapinfo:pen 2</Pen>  
            </LineStyle>  
            <!-- The following defines the blue fill -->  
            <Interior fill-opacity="0" fill="#id7" />  
        </AreaStyle>  
    </mfp-wms:WmsStyle>  
<Defs>  
    <Pattern id="id7">  
        <Bitmap uri="mapinfo:brush 2">  
            <ColorAdjustmentSet>  
                <ColorAdjustment color-1="nonWhite" color-2="blue"  
opacity="1"/>  
                <ColorAdjustment color-1="white" opacity="1"/>  
            </ColorAdjustmentSet>  
        </Bitmap>  
    </Pattern>  
</Defs>
```

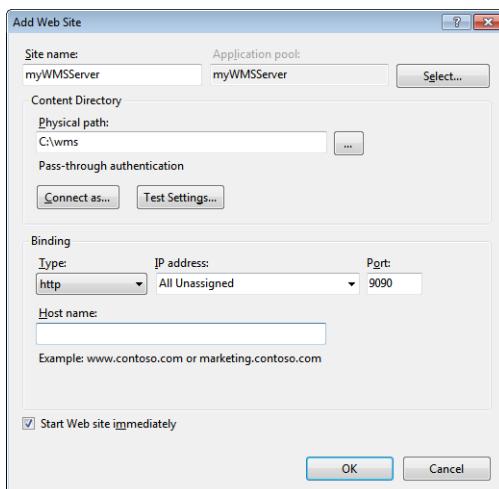
```
</Interior>
</AreaStyle>
</mxp-wms:WmsStyle>
</mxp-wms:WmsStyleSet>
<mxp-wms:MinScaleDenominator>10</mxp-wms:MinScaleDenominator>
<mxp-wms:MaxScaleDenominator>20</mxp-wms:MaxScaleDenominator>
<FeatureLayer id="id8" name="STATES" alias="STATES" volatile="unknown"
xmlns="http://www.mapinfo.com/mxp">
    <DataSourceRef ref="id1"/>
</FeatureLayer>
</mxp-wms:WmsLayer>
```

4. 在添加图元图层信息之后，保存 WMSSample.xml。

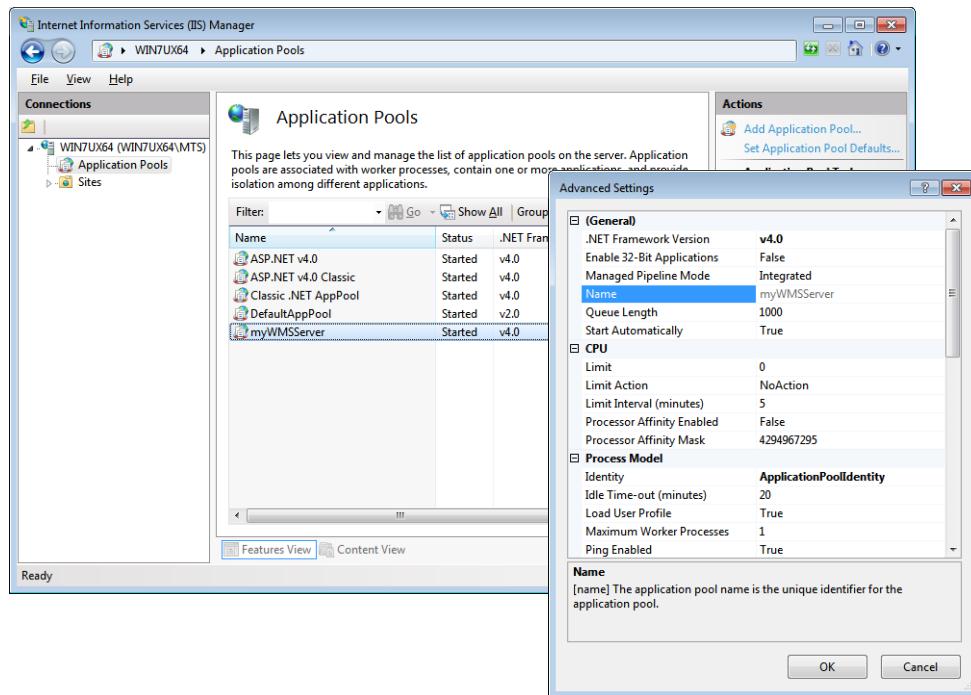
使用 Internet 信息服务 (IIS) 注册您的 WMS 服务器，并测试您的设置。遵循 IIS7 的说明

步骤 3a: 使用 IIS7 配置和测试 WMS 服务器

1. 右键单击 WMS 文件夹（例如 c:\wms）并选择“属性”。选择“共享”选项卡，然后单击“共享”按钮。将 IUSR 帐户添加到用户列表中。单击“共享”，然后单击“完成”返回到“属性”对话框。单击“安全性”选项卡。将 IUSR 帐户添加到安全性列表中。
有关设置 IUSR 帐户权限的信息，请参阅[了解 IIS 7 中的内置用户和组帐户](#)。
2. 打开 IIS7 管理器。右键单击“站点”，然后从菜单中选择“添加网站”。在显示的对话框中，为您的站点命名（例如，“WMS”），然后导航到 WMS 文件夹的物理路径。单击“连接为”按钮，并确保选中了“应用程序使用”。为网站指定一个未被占有的端口号，或者使用默认端口 80。单击确定。

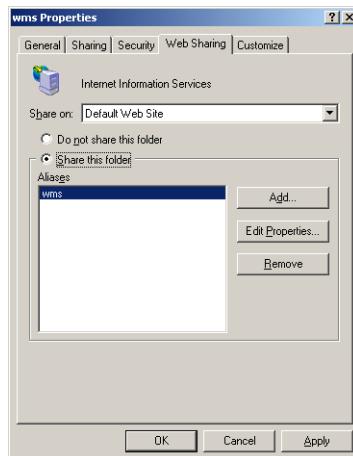


3. 单击左侧窗格中的“应用程序池”。右键单击 IIS7 已为您的 WMS 网站 ("WMS") 创建的新应用程序池，然后选择“高级设置”。将 .Net Framework 版本更改为 'v4.0'，然后单击“确定”。



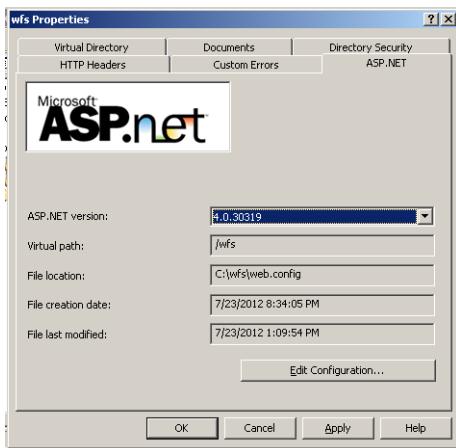
步骤 3b：使用 IIS6 配置和测试 WMS 服务器

- 右键单击 WMS 文件夹（例如 c:\wms）并选择“共享和安全”。在“Web 共享”选项卡上，选择“共享这个文件夹”单选按钮。如果希望为 Web 服务器设置别名，请单击“添加”按钮并在该对话框中提供名称。



- 打开 IIS（“开始”菜单 > 控制面板 > 管理工具 > Internet 信息服务）。展开默认的 Web 站点并按文件夹名称或按别名（若已使用别名）定位您的 WMS 服务器。

3. 右键单击该 Web 站点并选择“属性”。在 ASP.NET 选项卡下，从 ASP.NET 版本中选择 4.0.30319（MapXtreme 程序集在 4.0 Framework 下编译）。



4. 仍在刚才的“属性”对话框中，在“目录安全性”选项卡下，单击右上方的“编辑”按钮。在“身份验证方法”对话框中，选中“匿名访问”复选框。这样，WMS 服务的用户可以跳过用户名/密码身份验证过程。单击两次“确定”，关闭 IIS 窗口。



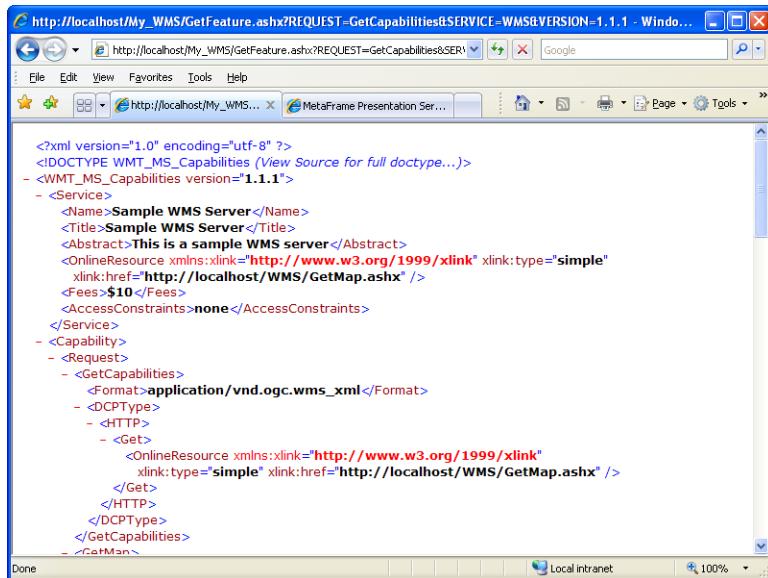
5. 通过从 Web 浏览器发送 **GetCapabilities** 请求，测试您的设置。在地址框中键入：

`http://localhost/wms/GetMap.ashx?REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.3.0`

使用您的 Web 服务器代替 localhost。如果已为 Web 服务器设置别名，请务必在 URL 中包含该别名。例如：

`http://localhost/My_WMS/GetMap.ashx?REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.3.0`

测试成功将返回类似下图的网页。若未返回功能列表，请检查您的配置文件以确保您的输入正确无误。既然是手动创建配置文件，自然很容易出现打字错误或丢失标记。



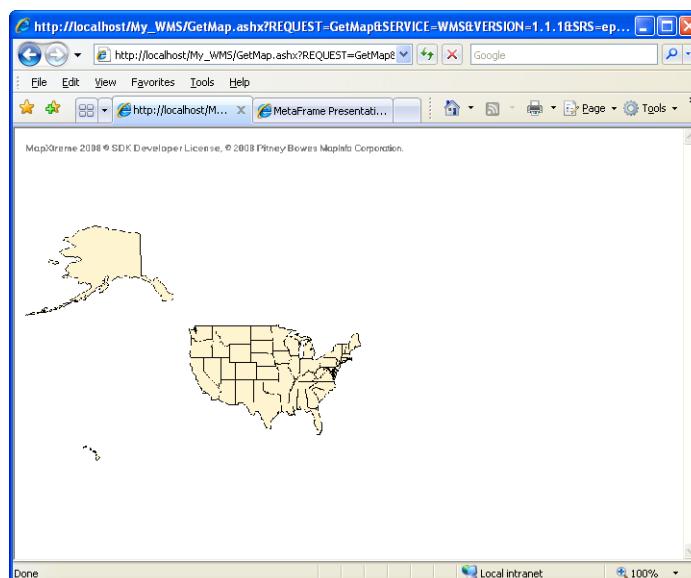
6. 如果在访问 WMS 服务器之后对 WMSSample.xml 作出任何更改，则需重置 IIS 才能使更改生效。
要重置 IIS，请在命令提示符下键入：

```
c:\>IISreset
```

7. 要请求测试地图图像，请参照以下示例发送 **GetMap** 请求：

```
http://My_Wms/GetMap.ashx?service=WMS&VERSION=1.3.0&SRS=epsg:4267&REQUEST=GetMap&LAYERS=States&STYLES=&BBOX=-180,0,0,90&WIDTH=800&HEIGHT=600&FORMAT=image/gif
```

WMS 服务器返回 **States** 图层的地图图像。



配置 WMS 服务器的图层信息

在 WMS 服务器配置文件中根据 MXP_WMSConfiguration_1_2.xsd 架构中列出的各个元素来描述图层。WMS 1.3 的新增内容以粗体显示。这些元素包括：

- 名称 - 图层名称，请求中使用该名称引用图层。
- 标题 - 用户易于识别的图层名称。
- 摘要 - 关于图层的较长叙述性说明。
- **KeywordList** 和 **Vocabulary** 属性 - 关键字或关键字短语列表，有助于进行目录搜索。**Vocabulary** 指示关键字的颁发机构。
- **SRSNameSet** - 该图层适用的空间参考系。如果该图层上可以使用多个 **SRS**，则逐一列出每个 **SRS**。此列表是无序的，同时应用于父图层和子图层。
- **Attribution** - 标识地理信息的来源。
- **MetadataURL** - 对应于特定层的数据的标准化详细元数据。
- **FeatureListURL** - 指向图层中图元的列表。
- **MinScaleDenominator** 和 **MaxScaleDenominator** - 定义 WMS 地图的适当刻度范围
- **WmsStyleSet** - 该图层及其任何子图层支持的无序预定义样式集。**WmsStyleSet** 是一组 **WmsStyle**，每个 **WmsStyle** 都有一个名称、标题、摘要和对样式定义的引用。每个样式都有一个 **LegendURL**，指向该样式图例的图像。

WMS 图层可由单个数据图层组成，也可由若干个分层图层组成。WMS 的基础实现允许客户端指定所需的图层、坐标系以及将这些图层渲染成地图图像所用的样式。

WMS 服务器开发人员的任务是决定如何将数据组装到 WMS 图层中。例如，您可以将地理数据分解为 10 个独立的 .TAB 文件，例如边界文件、点文件和折线文件。在 WMS 配置文件中，您可以将这 10 个图层表示成一个 WMS 图层，也可以将它们保留为各个图层。如果是以单个图层的形式向 WMS 客户端提供数据，则客户端将获得以单个 WMS 图像表示的所有 10 个图层。这种情况下，客户端不能剔除它们并不需要的信息。

但如果以多个独立的图层提供这 10 个图层，WMS 客户端可以根据需要选择以单个图像或多个图像的形式返回其中一个、几个或全部图层。在配置 WMS 服务器时必须决定需要为用户提供多大的灵活性。

您也可以嵌套图层，这样就可以通过请求父图层将所有子图层包含在地图图像中。

使用 MapXtreme Workspace 生成 WMS 配置文件

要生成 WMS 配置文件中的图层信息，请考虑从先前保存的 MapXtreme Workspace (.MWS) 中提取与每个图层有关的信息。

以下是使用 MapXtreme Workspace 管理器创建的 .MWS 的摘要，该文件定义了一个名为“World Countries”的图层。.MWS 布局是一个遵循 MXP_Workspace_1_5.xsd 中所含架构的 XML 文件。

```
<FeatureLayer id="id10" name="World Countries" alias="world"
volatile="unknown">
    <Visibility visible="true">
        <VisibleRange enabled="false">
```

```
<ZoomRange uom="mapinfo:length mi" minInclusive="true"  
maxInclusive="false">0 0</ZoomRange>  
    </VisibleRange>  
    </Visibility>  
    <DataSourceRef ref="id4" />  
</FeatureLayer>
```

现在, WMS 配置文件中也包含同样的图层信息。注意: 此信息几乎完全相同, 只是每个标记都包含 "mxp:", 表示这些元素属于 MapXtreme 工作空间架构 (MXP_Workspace_1_5.xsd)。

```
<mxp:FeatureLayer id="id10" name="World Countries" alias="world"  
volatile="unknown">  
    <mxp:Visibility visible="true">  
        <mxp:VisibleRange enabled="false">  
            <mxp:ZoomRange uom="mapinfo:length mi" minInclusive="true"  
maxInclusive="false">0 0</mxp:ZoomRange>  
        </mxp:VisibleRange>  
    </mxp:Visibility>  
    <mxp:DataSourceRef ref="id4" />  
</mxp:FeatureLayer>
```

您可以通过在 **Workspace** 管理器中设计图层并将信息复制到 WMS 配置文件中来构建图层。请参阅 [第 24 章: Workspace 管理器](#)。

图元图层及更多信息

在 WMS 图层中并非只能包含图元图层信息。例如, 您也可以在 **Workspace** 管理器中捕获标注、主题和样式覆盖并将其粘贴到 WMS 配置文件中的对应部分。

以下是为某个人口图层定义分级符号主题的 .MWS 的一部分。注意: 此处捕获了定义该主题的每个元素。在将它粘贴到 WMS 配置文件中时, 请务必在每个元素中添加 mxp:

```
<ObjectThemeLayer id="id8" name="Graduated Symbol Theme on World Capitals  
by Cap_Pop" alias="GraduatedSymbolThemeLayer1" volatile="unknown">  
    <Visibility visible="true">  
        <VisibleRange enabled="false">  
            <ZoomRange uom="mapinfo:length mi" minInclusive="true"  
maxInclusive="false">0 0</ZoomRange>  
        </VisibleRange>  
    </Visibility>  
    <DataSourceRef ref="id4" />  
    <FeatureGraduatedSymbolTheme id="id9">  
        <Alignment>  
            <HorizontalAlignment>center</HorizontalAlignment>  
            <VerticalAlignment>center</VerticalAlignment>  
        </Alignment>  
        <SymbolBaseSize useScale="false">  
            <MapScale>1.0</MapScale>  
            <DataValueAtSize>20000000</DataValueAtSize>  
            <PositiveSymbol>  
                <PointStyle>
```

```
<FontSymbol size="18" size-unit="mapinfo:length pt"
stroke="red" stroke-opacity="1" text=""" family="MapInfo 3.0
Compatible" />
    </PointStyle>
</PositiveSymbol>
<NegativeSymbol visible="false">
    <PointStyle>
        <FontSymbol size="18" size-unit="mapinfo:length pt"
stroke="blue" stroke-opacity="1" text=""" family="MapInfo 3.0
Compatible" />
            </PointStyle>
        </NegativeSymbol>
        <SymbolGraduation>sqrt</SymbolGraduation>
    </SymbolBaseSize>
    <SymbolLayout />
    <NumericValueExpression>
        <AttributeName>Cap_Pop</AttributeName>
    </NumericValueExpression>
    <SymbolLegendRowOverrideSet>
        <SymbolLegendRowOverride visible="false" row="4" />
        <SymbolLegendRowOverride visible="false" row="5" />
        <SymbolLegendRowOverride visible="false" row="6" />
    </SymbolLegendRowOverrideSet>
    </FeatureGraduatedSymbolTheme>
</ObjectThemeLayer>
```

Workspace 管理器

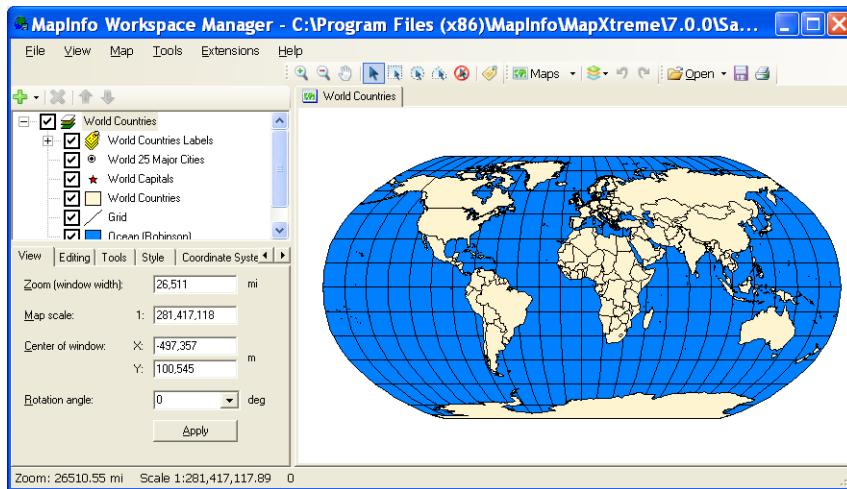
通过随 MapXtreme 附带的 Workspace 管理器应用程序，可以创建和管理基于 XML 的工作空间文件 (.MWS 格式)。**Workspace Manager** 简便的用户界面使得用户更易于组装组成地图的表，应用大量附加的设置并作为工作空间保存地图。完成的工作空间还可用于自己的应用程序，或用于打印或文件输出。

在本章中：

◆ Workspace 管理器的特性	394
◆ 工作区格式和内容.....	395
◆ Workspace 管理器菜单命令	395
◆ 图层控制	404
◆ 使用 Workspace 管理器的功能	415

Workspace 管理器的特性

Workspace 管理器允许用户控制大部分可以存储在工作空间文件中的设置。例如，工作空间可以包含有关制图图例和修饰的信息；不过，Workspace 管理器未提供任何创建制图图例或修饰的选项。要创建制图图例或修饰，请使用 API。



通过 Workspace 管理器，用户可以：

- 加载 XML 工作空间、表、geosets 和 MapInfo 工作空间 (.mws，不是 .wor 文件)。
- 将工作空间保存为 .MWS。
- 控制作为工作空间的一部分打开的表。
- 使用命名连接管理器创建和加载命名的连接。
- 添加、删除和查看一个或多个地图。
- 使用 Map 窗口顶部的选项卡切换地图。
- 使用内建的图层控件设置地图和图层可视性、图层和标记样式以及主题的属性。
- 添加、删除或改变定制标记。
- 创建允许用户将图层组织到逻辑组的组图层，以便用户通过单击就可以显示和隐藏整个组。
- 查看具有多个视图的下一个和前一个地图视图。
- 使用导航和手动标记放置的地图工具，并使用选择集工具来验证图层可选择设置是否正确。
- 预览和打印地图。
- 从最近文件列表快速打开最近使用过的工作空间
- 为地图、标注和主题图层创建半透明效果
- 为折线图元创建曲线标注
- 为地图添加网格图层。
- 将栅格图层重投影到当前地图窗口投影。
- 将由栅格和矢量图层组成的地图窗口重投影为不同的投影。
- 添加扩展可向“Workspace 管理器”和“图层控制”添加自定义功能。

工作区格式和内容

工作空间文件是一种 XML 文档 (.MWS)，其中包含了所有地图、表、图层的位置、描述和元数据以及工作空间的设置。因为工作空间文件是 XML 文档，所以工作空间是可移植的，这意味着用户可以与不同计算机、不同网络上的其它用户跨平台共享工作空间。

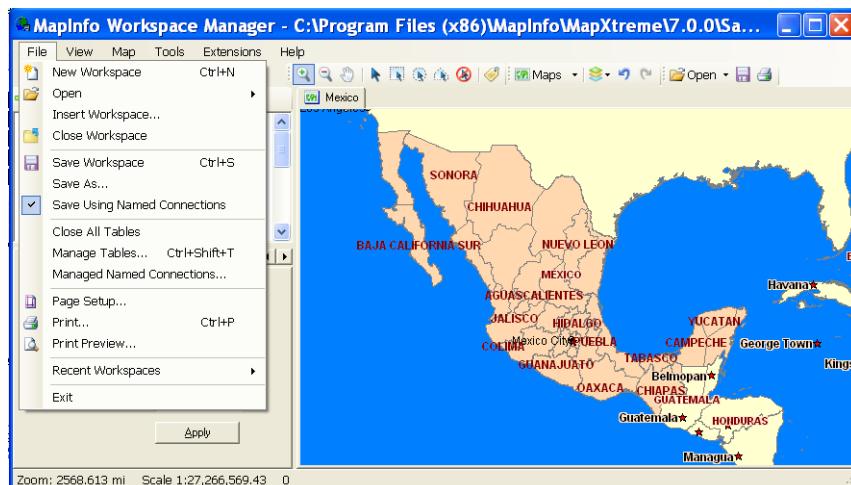
在 MapXtreme 中，XML 工作空间的可移植性通过使用命名连接实现。命名连接使用户可以根据自己的环境定义其它驱动器、路径和数据库连接信息，以便可以使用其它用户创建的工作空间。工作空间在打开时解析所有的路径和连接字符串。可以直接在 Workspace 管理器（文件 > 管理命名连接）中设置命名连接。有关 XML 工作空间结构的详细信息请参阅附录 C：理解 MapInfo 工作空间。

Workspace 管理器菜单命令

本节解释了 Workspace 管理器中可用的每个菜单命令。

“文件”菜单命令

“文件”菜单中的命令提供了所有标准的“文件”菜单功能（例如打开、保存和打印文件），还提供了一些 Workspace 管理器特有的功能。每个命令的信息如下所述。



新建工作空间

创建新的空地图，用户可以使用图层控件的“添加”工具或“文件”菜单的“打开表”命令将表添加到该地图。如果对当前工作空间做出更改，则创建新的工作空间之前将提问是否要保存更改。

打开

打开现有的工作空间或表。如果对当前工作空间做出更改，则打开现有的工作空间之前将提问是否要保存更改。“打开表”命令允许用户打开一个或多个表并将这些表添加到地图。

插入工作空间

将一个或多个工作空间中的地图和表添加到当前工作空间。

关闭工作空间

提示您保存任何更改并关闭工作空间。

保存工作空间

将地图保存为工作空间。

另存为

用新的文件名保存工作空间的副本。

使用命名连接进行保存

将命名连接信息保存到工作空间。另请参阅[管理命名连接](#)。

关闭所有表

“关闭所有表”命令关闭所有打开的表。

管理表

“管理表”命令显示列出组成地图的表的对话框，还可以使用户打开地图中可能包括的其它表。单击**打开**以显示“打开”对话框并打开表。打开的表添加到“管理表”对话框中打开表的列表。然后可以使用“添加”工具将表添加到地图，“添加”工具位于 **Workspace** 管理器图层控件窗口上。



要关闭表，首先单击列表中的表以突出显示要关闭的表，此时将激活**关闭**按钮。最后单击**关闭**，关闭该表。引用关闭的表的图层从地图删除。

工作空间可打开不在地图中的表。

管理命名连接

命名连接使用别名说明了到数据源的连接。您可以创建以下类型的命名连接：FilePath、DatabaseSource、ODBC 或 Oracle OCI。指定连接字符串或文件路径后，可以将命名连接保存为 XML 用于以后检索。可以设置默认的连接，以便运行 **Workspace** 管理器时到数据源的连接可用。命名连接也保存到工作空间。



页面设置

“页面设置”允许用户指定打印地图的纸张大小、方向和页边距。也可以使用该选项访问特定于打印机的设置。

打印

“打印”命令允许用户将地图打印到纸张或作为文件输出。在“打印”对话框中，可以指定要使用的打印机，如果作业是多个页面则可以指定打印的页面范围，此外还可以指定打印的份数。打印机属性允许用户设置布局和特定于正使用的打印机的其它选项。**打印到文件**复选框允许用户打印输出到文件。

打印预览

使用“打印预览”命令可以在打印之前查看输出的外观如何。

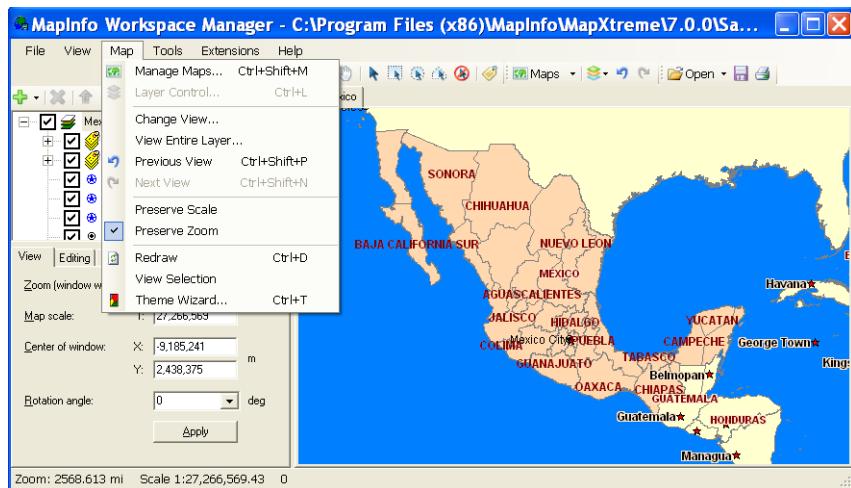
最近使用的工作空间

“最近使用的工作空间”显示了最近打开的工作空间的列表。

查看菜单命令

此菜单可显示或隐藏“工具栏”、“状态栏”和“图层控制”。

“地图”菜单命令



使用“地图”菜单中的命令可以添加和删除地图，操控地图的视图，还可以创建主题地图。也可以在弹出菜单中使用视图命令。在 **Workspace** 管理器地图窗口中右键单击显示菜单。

管理地图

此菜单可显示“管理地图”对话框，您可使用该对话框添加或删除地图、设置默认地图、重命名地图和/或别名、显示/隐藏图例选项卡（如地图有图例）。“新建”命令允许用户使用目前打开的表创建新的地图窗口。管理工作空间地图还提供地图中的图层数量和图例数量。

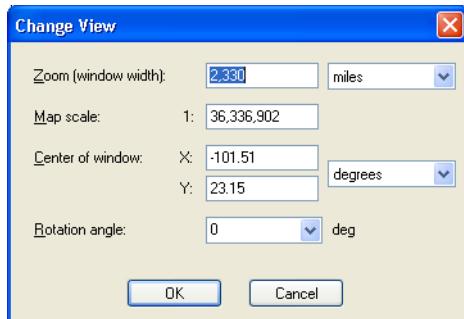
图层控制

此菜单可打开“图层控制”对话框。当“查看”>“图层控制”未选中、“图层控制”不可见时，它处于活跃状态。有关“图层控制”功能的描述，请参阅[图层控件工具](#)。

改变视图

“改变视图”命令允许用户更改地图的当前视图（也就是说，窗口中目前显示的地图区域）。可以将地图的缩放和比例更改为自己的设置。也可以设置地图窗口的中心或更改旋转角度。

“改变视图”对话框允许用户选择缩放宽度的单位（英里、千米等）和中心 X/Y 坐标的单位（米、度等）。“改变视图”对话框中选择的任何单位也都用于图层控件。例如，如果要将图层控件中的所有距离以千米显示而不以英里显示，则显示“改变视图”对话框并从显示在“缩放”字段旁边的单位列表选择千米。



查看整个图层

使用“查看整个图层”命令查看整个图层或地图中的所有图层。“查看整个图层”对话框显示构成地图的图层的列表。从列表选择想要的图层或选择所有图层，以便所有的图层完全显示在视图中，然后单击确定。地图重绘以显示整个地图。

前一视图

使用“前一视图”命令返回地图的前一个视图。

下一视图

使用“前一视图”后可以使用“下一视图”命令。使用“前一视图”命令之前使用“下一视图”命令重新显示屏幕上地图的视图。

“前一视图”和“下一视图”命令可以一起使用来在地图的两个视图之间向前和向后切换。这些命令可以像工具栏的工具一样使用。

保持比例 / 缩放

使用“保持比例”和“保持缩放”命令使得缩放和（或）比例与用户更改地图的大小和形状相同。

重绘

使用“重绘”命令以重绘地图。

视图选择集

使用“视图选择集”命令放大或缩小一个或多个选中的对象。

主题图向导

通过地图 > 主题图向导菜单命令可以创建图元主题图和标记主题图。图元主题包括范围、单值、点密度、分级符号、饼图和条形图。标记主题包括有范围和单值。

如果地图包括至少一组标注（显示在标记图层中），则用户可以创建标记主题图。标记主题图根据表中的数据将不同的标记样式（不同颜色、字体大小等）分配到每个标注。例如，使用标记主题显示其它标记位置上的突出部分。有范围标记主题组根据类似的数据值（例如人口）标记。处于某些人口范围内的城市使用一种样式标记，而处于其它范围的城市用另一种样式标记，通常不太突出的样式指示了不必使用人口值标记的城市大小。

“主题图向导”可帮助您完成 3 个对话框以创建主题图。

要创建主题图：

1. 选择地图 > 主题图向导。

将显示“创建主题图：步骤 1（共 3 步）”对话框。

2. 选择图元主题或标记主题，并选择要创建的主题的类型。

3. 单击下一步。

将显示“创建主题图：步骤 2（共 3 步）”对话框。

4. 选择要加底纹的表。

5. 选择要使用的数据。从包含数据的表选择列或选择表达式以使用表达式从表派生使用的数据。

6. 单击下一步。

将显示“创建主题图：步骤 3（共 3 步）”对话框。在这一步中，可以定制主题类型设置、样式和图例。（有关对主题地图应用半透明效果的说明，请参阅[如何对主题图应用半透明效果](#)。）

7. 单击应用以应用定制的设置。

8. 完成后单击确定。

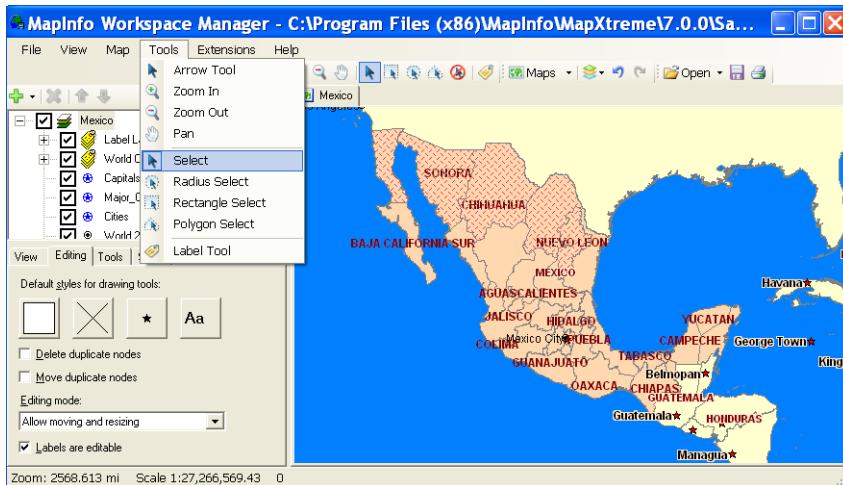
在图元或标注上创建饼图、条形图和 IndividualValue 主题图时，3 个对话框中第 2 步中的“忽略零值”选项可用。注意，设置该选项时，Workspace 管理器中主题图构建操作的性能可能会受到影响。

要修改主题图，请突出显示“图层”树中的主题图，然后单击显示的“主题图”选项卡。单击“修改主题图”按钮，以针对主题图进行任何更改。

要删除“图层”列表中的主题图，请右键单击主题图的名称，然后选择“删除”。也可重命名主题图并显示/隐藏“图层”树中的该图层。

“工具”菜单命令

Workspace 管理器的“工具”菜单提供了通过菜单命令对地图工具的访问。这些工具允许用户以各种方式放大和缩小地图，更改地图的位置并选择地图对象。在 Workspace 管理器工具栏上也可以使用这些相同的工具。下文解释了每种工具。



箭头

这是基本的点选工具。它不选择地图图元。

放大

使用“放大”工具放大地图的区域视图。要放大地图：

1. 选择工具 > 放大以激活该工具。

光标更改为带有加号的放大镜。

2. 单击地图。

地图在放大的区域视图上重绘，并以单击的点为视图的中心。

缩小

使用“缩小”工具缩小地图的区域视图。要缩小地图：

1. 选择工具 > 缩小以激活该工具。

光标更改为带有减号的放大镜。

2. 单击地图。

地图在缩小的区域视图上重绘，并以单击的点为视图的中心。

一种工具实现放大缩小

要使用相同的缩放工具进行放大和缩小操作，请按下 **Ctrl** 键。按下 **Ctrl** 键使用放大工具时，每单击一次可逐步缩小。按下 **Ctrl** 键使用缩小工具时，每单击一次可逐步放大。

平移

使用“平移”工具重定位地图，而不更改缩放级别。例如，可以重定向地图的视图使得某些国家或城市位于中心。要平移地图：

1. 选择工具 > 平移以激活该工具。
光标更改为手形图标。
2. 单击地图并按下鼠标按钮将地图拖到想要的位置。
地图重绘以反映新的位置。

选择

使用“选择”工具一次选择一个对象或选择通常位于相同区域的所有对象。

要使用“选择”工具选择对象：

1. 选择工具 > 选择以激活工具。
光标更改为箭头。
2. 在地图上单击要选择的对象。
选中的对象突出显示。

半径选择

使用“半径选择”工具选择所有位于给定半径内的对象。例如，如果有捐血人的表和捐血点的表，则使用“半径选择”工具就可以创建捐血人的临时列表，这些捐血人居住在距离每个捐血点方圆 1.5 英里内。

注意，“半径选择”工具选择所有质心位于圆内的对象。对象不必完全以圆圈为边界。要选择半径内的对象：

1. 选择工具 > 半径选择来激活该工具。
将鼠标指针在地图上方移动时，指针将变为下方带有小圆圈的选择箭头。
2. 单击地图上的某个位置，用作半径搜索的中心点。例如，如果要搜索消防站方圆 2 英里内的所有消防栓，则单击消防站并将其用作中心点。
3. 按下鼠标按钮并将鼠标拖离中心点。
Workspace 管理器围绕该点绘制圆并在 StatusBar（屏幕的左下角）报告圆的半径。
4. 达到想要的半径后释放鼠标按钮。
Workspace 管理器突出显示所有位于圆内的地图对象。

矩形选择

使用“矩形选择”工具选择矩形内的对象。通过单击和拖放使用“矩形选择”工具，创建虚线矩形，或围绕要选择的对象创建选取框。

注意，“矩形选择”工具选择所有质心位于矩形内的对象。对象不必完全以矩形为边界。要选择矩形内的对象：

1. 选择工具 > 矩形选择以激活该工具。将鼠标指针在地图上方移动时，指针将变为下方带有小矩形的选择箭头。
2. 单击地图上处于选取框区域外部的某个位置。
3. 按下鼠标按钮并拖动鼠标围绕要选择的点形成虚线矩形。
4. 达到想要的矩形大小后释放鼠标按钮。

Workspace 管理器突出显示所有位于矩形内的地图对象。

多边形选择

“多边形选择”工具选择位于地图上绘制的多边形内的地图对象。

要使用“多边形选择”工具选择对象：

1. 选择工具 > 多边形选择以激活该工具。

将鼠标指针在地图上方移动时，指针将变为下方带有小多边形的选择箭头。

2. 单击要放置多边形的第一个终点的地图位置。沿任意方向在地图上移动光标。

Workspace 管理器从用户单击光标的点处绘制直线。

3. 单击创建另一个终点。继续移动光标，然后在达到想要的多边形边数时单击。

4. 要封闭多边形，请在尽可能接近第一次单击的地方做最后一次单击，或者双击最后一个点。

Workspace 管理器封闭多边形并选择多边形内的对象。

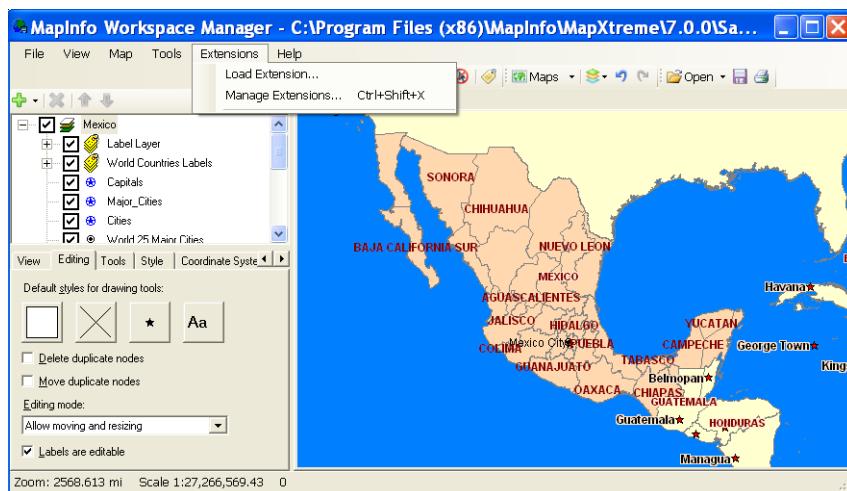
Label

“标记”工具可用来手动将标记添加到用户单击位置的地图。水平标注和曲线标注均可以使用标注工具来绘制。“标注”工具最简单的使用方法为使用默认设置绘制标注。但 MapXtreme 具有广泛的标注功能，它提供数量极多的样式、位置和可视性选项，可使标注更生动。另请参阅[标注图层设置](#)和[曲线标注](#)。

扩展菜单命令

从 MapXtreme 7.0.0 版开始，可将自定义 .NET 程序集加载到 Workspace 管理器中，以通过“扩展”菜单扩展其功能。例如，您可能需要多个选项卡窗口，每个窗口的设置都可以应用于不同的地图。“扩展”还可以将菜单项添加到任何一个 Workspace 管理器菜单（包括“扩展”菜单）。

此菜单提供两个命令：“加载扩展”和“管理扩展”。



加载扩展

此命令显示“打开”对话框，以便导航到自定义扩展的位置。创建和保存扩展时，扩展必须使用文件命名规则命名为：`<your_extension>.WorkspaceManager.Extensions.dll`。

管理扩展

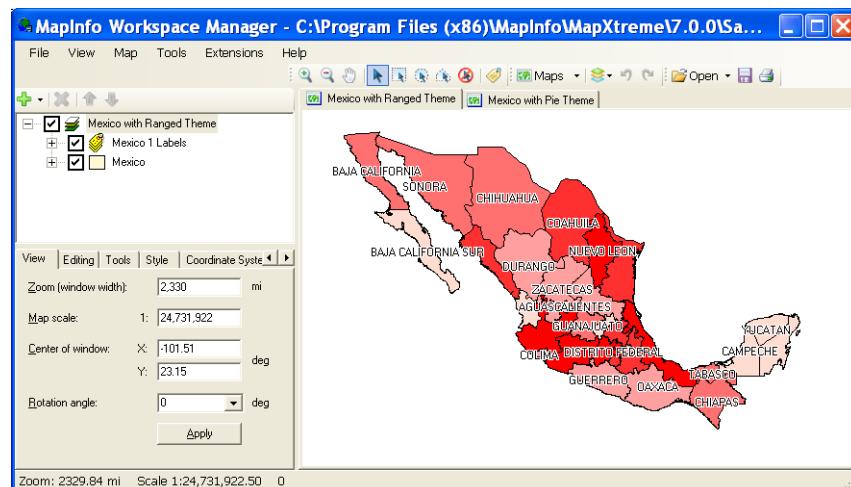
此命令显示“扩展管理器”对话框，以便加载或卸载扩展并修改扩展的属性。已加载的扩展显示在列表对话框中并显示其名称、版本号和程序集位置。

要了解创建自定义扩展的方法，请参阅附录 B：自定义 MapXtreme 中的 **Workspace** 管理器扩展。

图层控制

Workspace 管理器应用程序窗口分为两个主要部分。图层控件窗口和命令位于左侧，而地图窗口位于右侧。图层控制窗口由图层树组成，图层树显示已打开的地图及其附带的所有图层以及包含该地图和图层设置的对话框。

Workspace 管理器的图层控件特性使得用户可以组装地图的图层，并将设置应用到管理图层或地图如何显示的独立图层或整个地图。



图层控件工具

通过图层控件窗口顶部的工具，用户可以方便地从图层树窗口中添加、移动和删除图层：

- “添加”工具允许用户打开表，并将组图层和标记图层插入地图中。
- “删除所选项目”工具从地图删除选中的图层。
- “上移”和“下移”工具栏按钮允许用户在图层列表中上移和下移图层，更改图层在图层列表中显示的顺序。

图层树

图层控件显示图层树，该树显示地图以及地图中的所有图层，包括主题、标注和网格图层。图层树允许用户执行以下操作：

地图别名

将鼠标指针悬停在地图节点（图层树顶端的节点）上方时，工具提示会显示该地图的别名。工具提示对需要知道地图别名的开发人员很有帮助（例如，在设置 Web 控件的属性时需要指定地图别名）。

要更改地图别名，可右键单击地图名称并选择“设置地图别名”。



显示图层

通过图层树中各图层旁边的复选框，单击一次即可切换图层的可视性。

更改图层顺序

要更改图层顺序，可以选择图层并单击“上移”或“下移”工具栏按钮。此外，可以向上或向下拖动图层以更改图层在列表中的位置。

有多种与拖放操作有关的特殊情况：

- 要将图层移动到组图层中，请将图层拖动到组上。
- 要将新的标记源添加到标记图层，请将图层拖动到标记图层上。

如果不想将图层移动到组中（相反，如果想重定位图层使得该图层位于组图层上），则完成拖放操作之前按下 Shift 键。类似地，如果不将新的标记源添加到标记图层，则按下 Shift 键。

显示上下文菜单

图层树中的每个地图或图层都有上下文菜单。要显示上下文菜单，请右键单击地图或图层，或按下 Shift+F10 显示目前所选图层的菜单。

上下文菜单的项目取决于指定图层的类型。特别要注意以下部分：

- 要将样式覆盖添加到图层，请显示上下文菜单并选择“添加样式覆盖”。注意，每个图层具有多个样式覆盖，每个图层还具有不同的缩放范围；这允许用户设置地图以便在缩放时点显示得更大，道路显示得更宽。

您也可以为无缝栅格图层添加样式覆盖。在突出显示样式覆盖时，您可以从“可视性”选项卡中访问“栅格图像样式”对话框。

- 要重命名图层树中的任意项目，请右键单击项目并选择“重命名”。也可以按下 **F2** 重命名选中的项目。注意，以该方式重命名图层不重命名初始表；重命名操作只更改显示在图层树中的文本。名称信息将在保存工作空间时存储在工作空间文件中。
- 要删除图层树中的图层，请右键单击该项并选择“删除”。此外，也可按下 **Del** 键，删除该图层。
- 默认情况下，图层添加到地图中时处于可选状态。这是由“图层控制”中“选项”选项卡内的“可选”复选框控制的（请参阅[选项](#)）。但也可将其指定为唯一的可选图层，所有其它图层都将被设置为不可选状态。如果地图有许多图层但用户只想让一个图层具有可选特性，此功能非常方便。仅将一个图层设置为可选：
 - 右键单击“图层控制”中的图层名称，或访问上下文菜单。
 - 选中[将其设置为唯一的可选图层](#)。现在只有这一个地图图层具有可选特性。即可用任一选择工具选择该地图上的对象。所有其它地图图层都将处于不可选状态。

图层控件选项卡

图层树下方的图层控件选项卡提供了可以应用到地图及地图中每个图层的其它设置和控件。不同的选项卡控制地图和图层设置。显示何种选项卡取决于图层树中突出显示的是地图还是图层以及突出显示了哪些类型的图层。例如，当图层树中突出显示地图时，将显示地图选项卡；突出显示图层时，将显示图层选项卡。如果突出显示某个主题、标注或网格图层，则将显示这些图层特定的选项卡。下一节将介绍各个选项卡的选项。

地图设置

在图层树中选择某个地图后，将有以下选项卡可用：视图、编辑、工具、样式、坐标系和边界。

View

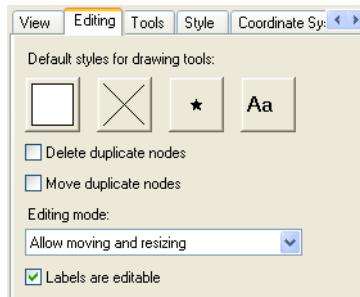
“视图”选项卡允许用户控制总的地图外观。可以设置缩放级别、比例、中心点（以度为单位）和旋转角度。单击**应用**按钮以应用设置。



对于光栅重投影，可设置此选项卡中的旋转角度，并打开 / 关闭“光栅重投影”选项卡中的重投影。请参阅[栅格重投影](#)。

编辑

应用程序使用绘制工具、调整对象大小、移动和删除对象节点时，“编辑”选项卡中的选项允许用户控制某些地图编辑任务（例如用于绘制对象的样式）。



样式框允许用户指定应用程序使用的任何绘制工具的默认样式。单击样式框打开对应的样式对话框。选择的设置保存在工作空间中。用户在使用绘制工具的应用程序中打开工作空间且在地图上绘制对象时，应用程序使用这些样式设置。

也可以指定是否要删除或移动重复节点。勾选相应的复选框。

可以为地图指定编辑模式：

- 无 - 不可以在地图上执行任何编辑。
- 允许移动和重定大小 - 地图对象可以移动和调整大小
- 允许编辑节点 - 节点可以移动或删除。
- 允许添加节点 - 可以将节点添加到对象。

选择的编辑模式应用到所有地图中可编辑的图层。要控制图层的可编辑性，可勾选图层“选项”选项卡上的“可编辑”复选框。

进入“编辑”模式时，不妨打开地图图元的节点。突出显示图层节点（不是地图节点），并勾选“可视性”选项卡上的复选框。

要控制标注是否可编辑，可勾选或取消勾选复选框标注可以编辑。

有关运用“编辑”模式的示例，请参阅[使用 Select 工具编辑 FeatureGeometry](#)。

工具

“工具”选项卡允许用户控制 InfoTips 的显示，激活“对齐节点”并设置对齐公差，还激活“动态选择集”工具。

显示提示信息复选框用于控制用选择工具悬停在图元上时是否在弹出窗口中显示有关该图元的信息。请参阅[选项](#)。

如果抓取到节点复选框被选中，则诸如“选择”工具等地图工具将自动搜索附近的节点。如果节点在附近，则显示交叉线指示最近节点的位置。例如，使用“半径搜索”工具时可能要选择“对齐节点”复选框，且要确保以地图上点图元的精确位置为中心进行搜索。在提供绘制工具的应用程序中，“对齐节点”图元尤其重要，因为用户通常需要在现有图元的精确位置绘制图元。

抓取范围设置指定了工具将搜索“可抓取”节点的范围。可以选择哪些图层使用“对齐节点”图元。例如，可能要打开“对齐节点”，但仅在特定图层中光标位于图元附近时才显示对齐交叉线。要打开或关闭特定图层的“对齐节点”图元，请选择图层树中的图层，然后在“选项”选项卡中选中或清除对齐节点复选框。

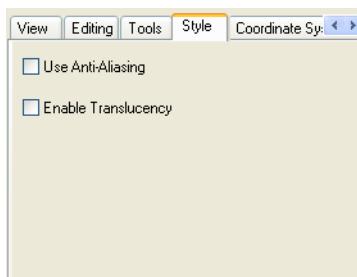
动态选择集工具复选框可控制图元是立即选中（使用选择集工具时）还是在使用选择集工具释放鼠标按钮时选中：

如果“动态选择集工具”复选框未选中，则使用工具完成之前选择集工具实际上不选择地图上的任何图元。例如，指定半径并释放鼠标按钮之前“半径选择”工具不选择任何图元。

如果“动态选择集工具”复选框被选中，则拖动鼠标时图元动态选择或删除。例如，如果使用“半径选择”工具，则拖动鼠标扩大半径时可以选中更多图元。

样式

通过“样式”选项卡中的选项能控制半透明度和抗锯齿属性。



- **使用抗锯齿** - 如果以较低分辨率呈现高清晰度转换，则可用此选项对线、曲线和区域边界的锯齿状边缘进行平滑处理。如果选中“使用抗锯齿”，则会自动选中“启用半透明”。每当取消选中“启用半透明”时，都会自动取消选中“使用抗锯齿”。
- **启用半透明** - 在屏幕、打印机或文件导出中绘制地图时，通过使用此选项可向样式颜色和图层中添加半透明值。启用半透明效果后，可使用样式对话框中的半透明滑块。
此属性对栅格半透明度没有影响；但必须选中“启用半透明”才能打印半透明的栅格图像。



如果通过启用半透明效果和抗锯齿渲染质量较高的地图，尤其是在有三个或更多透明图层的地图中，则往往会降低渲染速度。

有关抗锯齿和半透明度的详情，请参阅[使用 GDI+ 半透明度和抗锯齿增强渲染能力](#)。

坐标系统

“坐标系”选项卡指示地图的坐标系并允许用户更改坐标系。

请按以下步骤执行：

1. 单击坐标系按钮以显示“选择坐标系”对话框。
2. 从列表中选择坐标系，然后单击确定。

边界

在地图中，“边界”选项卡显示了当前地图视图的边界。单击查看整个地图按钮查看地图的全貌。

栅格重投影

通过“栅格重投影”选项卡，您可以指定地图中栅格和矢量图层的重投影设置。

也可以控制栅格和矢量图层的重投影。在向地图添加栅格或矢量图层时，新的图层将重投影到当前地图窗口投影中。

在更改同时包含矢量和栅格图层的地图窗口的投影时，所有图层（包括栅格和矢量）都将重投影到新的地图窗口投影中。

这些设置也会在 **MapXtreme API** 中公开。有关详细信息，请参阅《开发人员参考》中 **MapInfo.Mapping.FeatureViewer** 类的 **RasterReprojection** 和 **RasterReprojectionResampling** 属性。



使用“栅格重投影”复选框打开或关闭“栅格重投影”。选中该复选框以打开“栅格重投影”；清除该复选框以关闭“栅格重投影”。栅格重投影默认为关闭，这意味着在向地图中添加栅格图层时不执行重投影。

使用“重投影方法”设置指定栅格重投影的方法。请选择以下选项之一：

- **优化** - 图像的栅格重投影由目标矩形在转换到源图像空间之后的外观确定。如果目标矩形看上去像是“严格”的矩形，也就是说，矩形的两条边与 X 轴平行，另外两条边与 Y 轴平行，则标准的 Windows 函数会沿两个方向拉伸源图像。如果图像未能通过矩形测试，则使用指定的重复取样方法来执行重投影。
- **始终** - 始终执行栅格重投影。**MapXtreme** 根据精确的公式计算图像的坐标，然后使用立方回旋法或最近邻居法对像素进行重复取样。

使用“重投影重复取样”设置指定重投影栅格图像的重复取样方法。

重复取样是根据源图像（第一个连续的亮度区域的离散表示）恢复目标栅格图像的像素值（通常与图像传感器瞬时感测的区域所覆盖的地面对象上某个点的亮度成正比）的方法。

目标图像的像素值可以使用两种重复取样方法来计算：立方回旋法和最近邻居法。请选择以下选项之一：

- **立方回旋法**具有离散性，因而能够提供最佳的像素值还原功能。在立方回旋法中，目标图像中的像素根据 **4x4** 像素窗口（位于源图像基准像素的中心）中的像素值来计算。遵照特别优化的流程为目标图像的每个像素计算基准像素的坐标。随后，根据基准像素坐标来计算像素。一般来说，对于航空图像和卫星栅格图像，为获得较好的图像质量，建议您使用立方回旋重复取样方法。¹
- **最近邻法**使用源图像中的基准像素值代替重投影图像中的像素值。这种重复取样方法的渲染速度比立方回旋法快，但精度比后者低。通常来说，对于栅格地图、网格和扫描的地图，为获得最佳的速度，建议使用最近邻居法。

要更改光栅重投影的旋转角度, 请转到“视图”选项卡, 提供角度并单击“应用”。请参阅 [View](#)。

图层设置

单击“图层”树中的地图图层节点时, “图层控制”中的以下选项卡均可用:

可视性

选中可视复选框可将地图图层设置为可视状态。选中图层树中图层旁边的复选框的效果与之相同。

选中在范围内显示复选框以指定图层显示的缩放范围或比例范围。如果选择缩放范围, 则以英里为单位指定最小和最大范围。图层显示在该范围内。如果选择比例范围, 则指定最近和最远的比例。图层显示在该比例范围内。

也可选中显示节点、显示质心和显示线条方向, 以便在地图图层上显示相应的项目。节点是用于定义线段、折线或多边形的点。质心是地图对象的中心。线条方向是线条的绘制方向(这有助于在街道图层上表示正确的地址设置顺序)。如果要编辑地图图元, 则可显示这些元素。“可编辑”复选框在“选项”选项卡中。

选项

“选项”选项卡复选框简化了对图元图层的编辑和定制:

- 可选 - “可选”复选框选中时, 可以使用“工具”菜单命令或工具栏中的“选择”工具来选中图层中的对象。清除不想从中选择的任意图层的“可选”复选框。
- 可编辑 - 选中该复选框以生成可编辑的图层。
- 画图工具可添加图元 - 如果准备将该工作空间用于提供绘制工具的应用程序中且要绘制工具在该图层中创建新的图元, 则选择该复选框。
- 显示 InfoTips - 选中“显示 InfoTips”复选框可于悬停在所选图层中的地图对象上时显示 InfoTips。InfoTip 文本由“InfoTip 表达式”字段中表达式的结果组成。例如, 如果表达式是表中的列, 则 InfoTips 组成该列的值。如果表达式使用表中的列信息计算, 则 InfoTips 组成该计算的结果。
- 对齐节点 - 选择打开或关闭特定图层的“对齐节点”图元, 选择图层树中的图层, 然后选择或清除“对齐节点”复选框。
- 信息提示表达式 - 控制要显示在图层信息提示中的信息提示。在表中显示的范围可以不限于第一列。请参阅[第 10 章: 创建表达式](#)。

“选项”选项卡中的“设置”不受图层可视性影响。无论该图层当前是否可见, 都能够看到“选项”选项卡设置。

边界

对于图层, “边界”选项卡显示了选中图层的边界。单击查看整个图层按钮查看所有的图层, 或单击查看默认区域来查看图层的默认视图。

1. MapXtreme 中使用的立方回旋算法基于 S.K.Park 和 R.A.Schowengerdt, *Computervision, Graphics and Image Processing* (1983、第 23 卷, 第 258 到 272 页) 的作品。

信息

“信息”选项卡提供了与选中图层有关的信息。还给定了表（例如 MapInfo 表）的名称、路径和类型及其坐标系。

主题图层设置

单击某个专题图层时，“图层控制”中有以下选项卡可用。

可视性

主题图层选中时，“可视性”选项卡选项控制选中主题的显示。选择可视复选框显示主题图层；清除复选框关闭主题的显示。

选中在范围内显示复选框以指定主题图显示的缩放范围或比例范围。如果选择缩放范围，则指定最小和最大缩放距离。主题显示在该范围内。如果选择比例范围，则指定最近和最远的比例。主题显示在该比例范围内。

Theme

“主题”选项卡指示主题地图的类型和用于获得值的表达式。“主题”选项卡还允许用户修改主题地图。单击修改主题以更改样式或图例。

标注图层设置

在图层树中单击某个标注图层后，图层控件底部的“可视性”选项卡将处于可用状态。展开标注图层查看标注源时，将显示控制标注源中标注外观和内容的其他选项卡：AutoLabel、Style、Text、Position 和 Rules。

可视性

标记图层选中时，“可视性”选项卡中的选项控制标记的显示。选中可视复选框，显示标注；清除复选框，关闭标注显示。

选中在范围内显示复选框以指定标注显示的缩放范围或比例范围。如果选择缩放范围，则指定最小和最大缩放距离。标记显示在该范围内。使用缩放范围时，最大值具有排斥性 - 只有当地图的缩放值小于最大值时，该图层才可见。因此，如果将最大缩放值设置为 5000 英里并将地图缩放至恰好 5000 英里，该图层将会消失。

如果选择比例范围，则指定最近和最远的比例。标记显示在该比例范围内。

单击清除标注修改可将标注恢复为默认状态。此按钮用于删除通过“标注”工具手动添加的单个标注，以及恢复标注的原始位置。

自动标注

“自动标注”选项卡允许用户创建和管理自动标记的显示。选中**自动创建标注**复选框可为地图生成自动标注。选中**在范围内显示**复选框以指定自动标注显示的缩放范围或比例范围。如果选择缩放范围，则指定最小和最大缩放距离。如果选择比例范围，则指定最近和最远的比例。自动标记显示在该比例范围内。



样式

“样式”选项卡控制标记文本和标记线的样式。对于标记文本，使用“文本”样式框访问“文本样式”对话框。可以指定标记的字体、颜色、背景和其它文本效果。对于标记线，在设置标记线的样式的位置使用“直线”样式框访问“直线样式”对话框。在“标记线”组中，选择是否不要标记线、简单线或带有箭头的直线。

i 曲线标注不支持标注线。MapXtreme 可以加载在 MapInfo Professional 中创建的标注线或使用自定义标注工具创建的标注线。请参阅[曲线标注](#)。

文本

“文本”选项卡允许用户指定表达式，该表达式从列或表中派生的信息生成标记文本。

位置

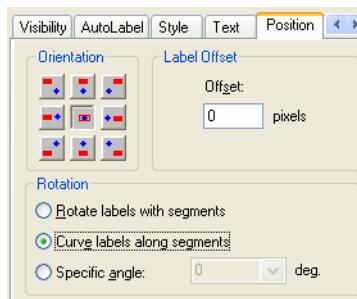
使用“位置”选项卡中的设置来设置标记的方向、偏移和旋转。

标记的方向是相对于其标记点的标记位置。单击其中一个按钮以选择方向。

标记偏移是标记离开其标记点的距离（以像素为单位）。

标记旋转是绘制标记的角度。有三种“旋转”设置：

- **沿线段旋转标注** - 若要标注诸如高速公路的直线图元，并且要以某个角度绘制每个标注（使得标注与高速公路的最近区段平行），则选中该选项。
- **沿线曲线标注** - 如果要标注诸如高速公路的直线图元，并且要让标注文本沿用高速公路的形状（即要曲线标注），请选择此选项。如果通过“图层控制”中的“样式”选项卡启用抗锯齿，则会改善最终曲线标注的外观。有关详细信息，请参阅[曲线标注](#)。
- **指定角度** - 以度为单位指定角度（例如零度）使得所有标注处于水平状态。



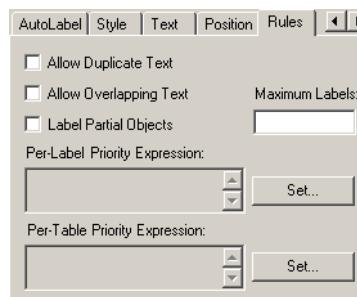
标尺

“规则”选项卡允许用户在地图上设置某些条件用于显示标记。

允许重复文本 选中“允许重复文本”复选框，允许对不同对象显示重复的标注，例如 Portland, OR 和 Portland, ME。在街道地图中，此选项亦可用来单独标注街道区段。

允许重叠文本 选择“允许重叠文本”复选框允许标注以互相重叠的方式绘制。某些标记不显示，因为这些标记与在地图上给定更高优先权的标记重叠。

标记部分对象 选择“标注部分对象”复选框标注折线和质心未显示在“地图”窗口中的对象。



标注最大数目 指定要显示的标注的最大数目。例如，将数据排序，使最突出的项（例如人口最多、收入最高、成长率最高等）位于表的顶部。如仅显示这类记录的前 100 项，请在“标注最大数目”对话框中输入 100。前 100 项之外的记录将不会显示标注，因此显示的 100 个标注对地图的影响更大。

每标记优先级表达式 该表达式字段可选。如果将该表达式字段保留为空，则单一标记源内的图元以不可预知的方式标记。例如，用户会发现某些小城市已标记，但一些大城市未标记（因没有足够的空间）。如果指定了表达式（必须是数值），则表达式将在地图的可视部分上为每个图元计算，且具有较大值的图元将被给定较高的标记优先权。要指定表达式，请单击设定按钮。

例如，假定用户正配置 WorldCapitals 图层的标记，该图层包含了表示城市的点图元。可能需要具有最多人口的城市具有最高的标记优先权。这种情况下，将指定标记表达式，例如：

`cap_pop`

`cap_pop` 列表示了每个首府的人口。指定 `cap_pop` 的“每标记优先级表达式”时，用户指定具有最多人口的城市应具有最高的标记优先权。因此，人口最多的城市首先标记，而只有留下足够的空间时才标记其它城市。

每表优先级表达式 该表达式字段可选。标记图层可以包含多个标记源；例如，可以具有一个表示一组标记世界国家的标记源，还可以具有另一个表示一组标记世界首都（城市）的标记源。默认情况下，列表顶部的标记源具有最高的优先权。如果要为标记源分配更高的优先权，则可以上移标记源（在标记源的图层控件列表中），或可以为每个标记源指定“每表优先级表达式”。

例如，如果为“世界国家”标记源给定的每表优先级表达式是 10，而为“世界首都”标记源给定的每表优先级表达式是 5，则国家标记比首都城市标记的优先权要高。

“组图层”设置

突出显示“图层控制”中的“组图层”节点时，“可视性”选项卡出现。可在此处控制可应用于组中所有图层的可视性设置。

组中的每个图层如要应用不同的可视性设置，可突出显示“组图层”下方显示的图层，然后更改显示的“可视性”选项卡。

“样式覆盖”设置

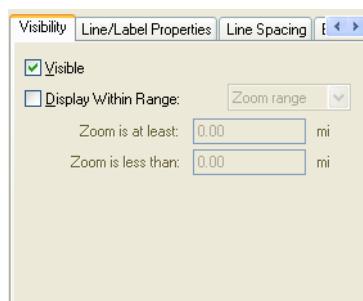
突出显示“图层控制”中的样式覆盖图层时，“可视性”选项卡出现。可在此处控制样式覆盖是否可视以及其缩放范围。也可在此处使用“样式”按钮，以设置该图层的区域、直线、符号和文本样式。

有两种绘制模式可供选择：正常和重叠。“正常”绘制模式使用样式覆盖绘制图层。“重叠”在图层顶部绘制样式覆盖。

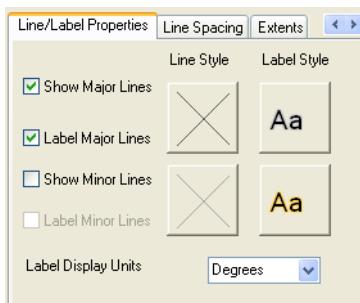
网格图层设置

单击某个网格图层时，图层控件的底部将有以下选项卡可用。

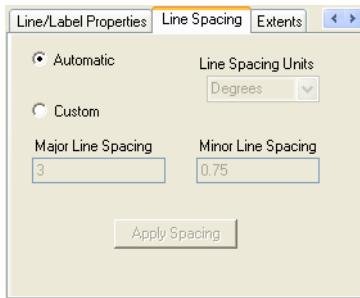
可视性



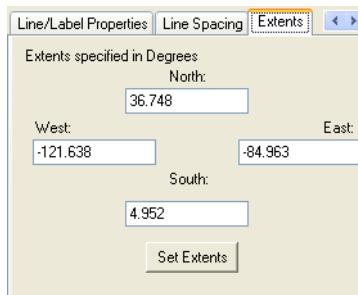
直线 / 标注属性



行距



边界



在该选项卡中作出任何选择之后，单击设置边界以应用更改。

有关网格的详细信息，请参阅[网格图层](#)。

使用 **Workspace** 管理器的功能

本节显示如何应用 **Workspace** 管理器的某些功能来增强地图的制图质量。它包括以下主题：

- 使用 **GDI+** 半透明度和抗锯齿增强渲染能力
- 创建半透明效果

- 曲线标注
- 网格图层

使用 GDI+ 半透明度和抗锯齿增强渲染能力

Microsoft Windows GDI+ 是 Windows XP 操作系统或 Windows Server 2003 操作系统的一部分，提供二维矢量图形和图像的打印输出。GDI+ 是 Windows 图形设备接口（GDI，早期版本 Windows 附带的图形设备接口）的改进版，在后者的基础上增加了新的功能并优化了现有的功能。通过 MapXtreme 中的 GDI+ 渲染可创建半透明的标注、主题和图层，还可对直线、曲线和区域边界的粗糙边缘进行平滑处理，从而以较低的分辨率实现高清晰度的渲染。在需要以较低分辨率应用抗锯齿以便以较低分辨率呈现高清晰度的渲染。

您可以在 API 中通过 `MapInfo.Mapping.DrawingAttributes` 和 `MapInfo.Mapping.LegendDrawing Attributes` 类中的两个新属性（`EnableTranslucency` 和 `SmoothingMode`）启用 GDI+ 渲染。您也可以在 Workspace 管理器中的“图层控件”中使用“样式”选项卡来启用这些属性。



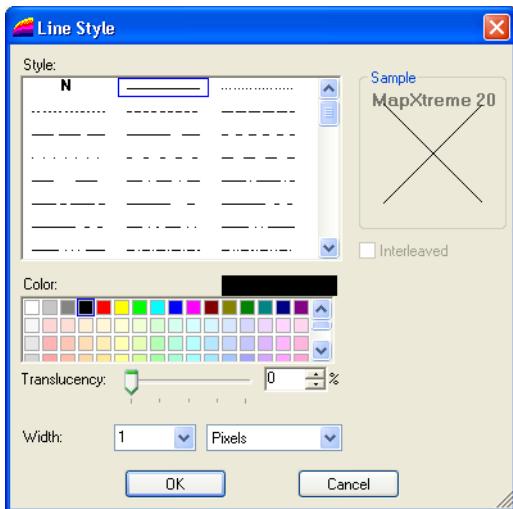
i 只有启用半透明度时，才可使用抗锯齿。这可由用户通过 Workspace 管理器实现，也可通过编程方式实现。如果选中“使用抗锯齿”，则会自动选中“启用半透明”。每当取消选中“启用半透明”时，都会自动取消选中“使用抗锯齿”。

以下地图反映了标注半透明度和抗锯齿对公路和告诉公路的影响。注意观察启用抗锯齿后区域边界和折线的平滑度。

在 GDI+ 增强渲染之前：在 GDI+ 增强渲染之后：



通过任何样式对话框中的半透明度滑块（**TrackBarValuePicker** 控件），您可使用滑块或数字选择框来选择介于 0 和 100 之间的百分比值。但是，此滑块仅在启用半透明度时才可用。



要通过编程方式更改颜色的半透明度，请使用 **System.Drawing.Color.FromArgb()** 方法。这样，您可以指定所需颜色的 alpha 值（在 GDI+ 中，alpha 通道是像素颜色数据中为透明度信息保留的部分）。有关详细信息，请参阅 [.NET 文档](#)。

该工具支持所有颜色中的半透明度；在启用或禁用半透明度时，该工具可正常工作。虽然必须启用“启用半透明度”选项才可打印半透明栅格图像，但它对后者的显示或导出没有任何影响。



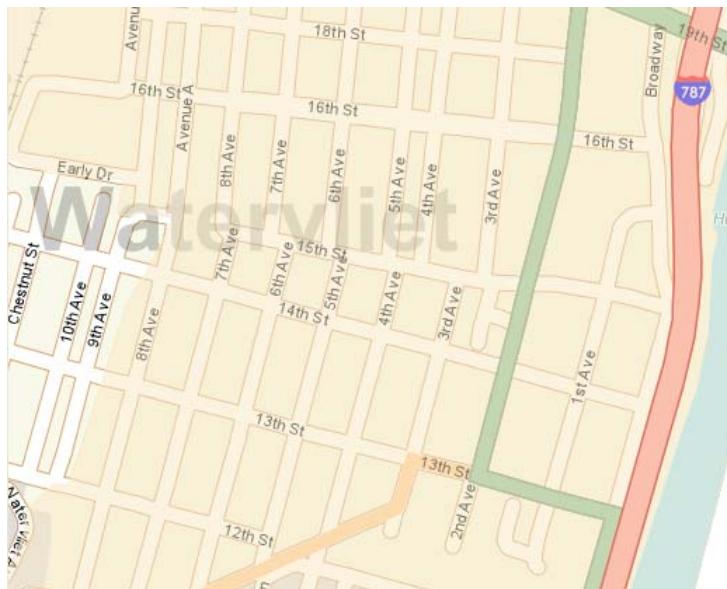
如果通过启用半透明效果和抗锯齿渲染质量较高的地图，尤其是在有三个或更多透明图层的地图中，则往往会降低渲染速度。

创建半透明效果

对地图图元和标注的颜色或对主题图层扩展样式的颜色应用半透明效果，可让您透过该颜色看到地图的其他图元。

如何对地图应用半透明效果

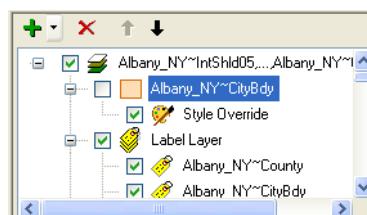
以下地图包含半透明的城市边界区域。此区域覆盖在“Watervliet”标注上方的部分显示出标注未被覆盖的部分（“Watervliet”的前两个字母）和标注被覆盖部分之间的区别。标注本身相对地图的其他部分也是半透明的。



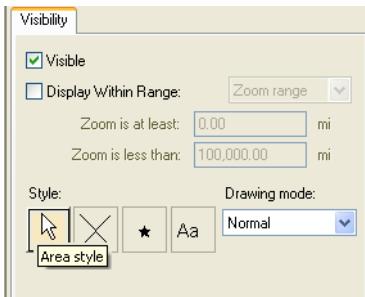
要对图层添加半透明效果，请使用样式覆盖来更改图层的原始样式。

① 通过编程方式添加覆盖样式修饰符 (OverrideType = AddNew) 时，绘制的第一个样式便是该图元的样式。由于重叠的样式是半透明的，因此可以透过它查看图元。

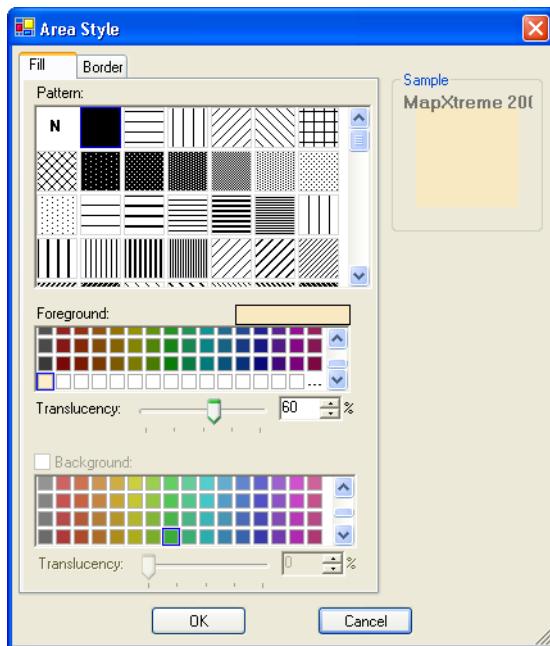
1. 在 Workspace 管理器的“图层控制”中，选择图层控制的“根”节点（对应于地图）。在“样式”选项卡中，选择启用半透明度。
2. 选择您希望添加半透明效果的图层，将该图层移动到列表的顶端，然后添加样式覆盖。



3. 在样式覆盖的“可视性”选项卡中，单击该图层中各个对象特定的样式按钮。对于本例，请单击区域样式按钮。



此时将显示“区域样式”对话框。



4. 使用半透明度滑块选择您希望应用的半透明度。每种颜色都有自己的半透明度。

如何对标注应用半透明效果

上例也显示了与水印类似的半透明标注。如果关闭半透明城市边界区域的可视性，则标注将如下所示：

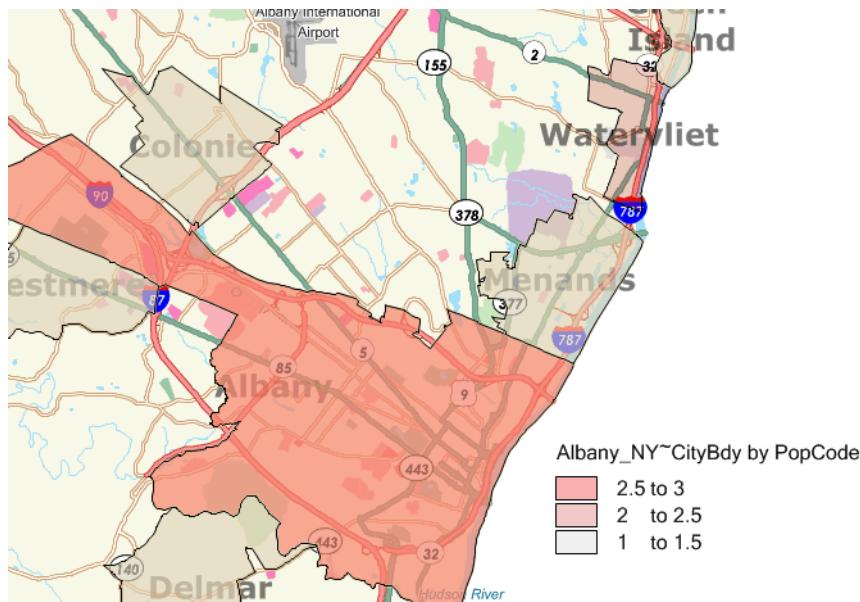


如果希望在地图上添加如本例所示的半透明标注，请执行以下操作：

1. 在 Workspace 管理器的“图层控制”中，选择图层控制的“根”节点（对应于地图）。在“样式”选项卡中，选择启用半透明度。
2. 在图层控件中选择标注源，然后进入“样式”选项卡。
3. 单击文本样式按钮，显示“文本样式”对话框。
4. 使用半透明度滑块选择希望为标注颜色设置的半透明度。上例显示了前景色半透明度为 80% 的文本。

如何对主题图应用半透明效果

可对主题地图应用半透明效果。下例所示的是对城市边界图层应用的范围主题。该图层位于其余图层的上面。



如果已启用半透明效果，则可为起始和结束主题范围（容器）选择半透明值，然后自动展开半透明色。这还会自动在起始和结束主题容器之间展开透明色。例如，上图中主题起始容器（灰色）的透明度值为 75%，结束容器（红色）的透明度值为 50%。由于此主题有 3 个容器，因此会自动将中间容器的透明度值设置为 63%（从 50 到 75 均匀分布）。

曲线标注

曲线标注是沿着某条线的曲线部分的标注。曲线标注改善了由多条线组成的地图图元（例如街道和河流）的外观。

通过在 **Workspace** 管理器中选择标注图层，然后在“位置”选项卡中选中与多段弯曲，即可生成曲线标注。有关通过 API 渲染曲线标注的信息，请参阅《MapXtreme 开发人员参考》中的“ILayout 接口”部分。

MapXtreme 试图为地图中的每个弧形和折线记录创建一个曲线标注，就像创建非曲线标注那样。例如，在街道地图中，街道可由多条折线或一条长折线组成。折线的长度和数量、控制是否可以创建曲线标注的规则以及您选择的标注选项均会对创建哪些曲线标注以及在何处显示这些标注造成影响。

图层数据中可能有些折线和弧线段不含标注名称条目。此时，**MapXtreme** 无法显示该段的标注。

有几个规则决定 **MapXtreme** 是否可以显示曲线标注：

- **MapXtreme** 只能使用 TrueType 字体绘制曲线标注。如果选择非 TrueType 字体，则会代之以相当的 TrueType 字体，您选择的标注的显示可能与预期的显示有所不同。此外，如果将使用非 TrueType 的水平标注更改为曲线标注，则由于字体替换可能导致新标注的显示有所不同。
- 标注的字符串部分必须与标注的弧线或折线相称。如果不相称，**MapXtreme** 会认为该标注太长并抛弃该标注。
- **MapXtreme** 不能为锯齿非常严重的折线绘制曲线标注，不过这得取决于该线的曲率。
- 有的街道段虽然看似笔直，但其标注却可能是曲线标注。这是因为该街道段的折线数据包含当前缩放级别下无法察觉的曲线。标注将沿着街道的弯道，尽管此弯道不可见。如果放大地图至足够接近的距离，便可看到街道中的弯道。
- 在自身上弯曲的标注将被丢弃，且不会显示。
- 对于重叠检测、重复文本和局部段标注，曲线标注遵循的规则与非曲线标注相同。这些规则中的每一条都会影响标注的显示方式和显示时间。
- 在弧线或折线的任一点，均可使用标注工具创建曲线标注。
- 您不能像移动其他标注那样移动曲线标注；但您可以使用标注工具来调整曲线标注的位置。
- 使用曲线标注时，“标注线”控件将被禁用。
- 曲线标注和非曲线标注保留给工作空间中的图层使用。
- 不能为曲线标注添加下划线。
- 无论是否启用半透明度和抗锯齿，曲线标注始终平滑绘制。

曲线标注中的双向字符串支持

曲线标注支持文字从右到左的语言类型。增加了对 Windows 随附的 Uniscribe 库的支持，该库可以处理环境字形修整和重新排序双向脚本，例如希伯来文和阿拉伯文。其方法可以分析使用复杂脚本的字符串，并将其分为可以重新排序以供显示的多个单独批次。

在 MapInfo Professional 中创建的曲线标注

可以将 MapInfo Professional 中可用的曲线标注显示设置保存到 .MWS 工作空间并加载到 MapXtreme 中。在 MapInfo Professional 中对曲线标注位置所作的更改也可保存为 .MWS 工作空间并加载到 MapXtreme 中。下面介绍了 MapXtreme 可以加载的显示设置和位置修改设置：

- 虽然 MapXtreme 不支持在 Workspace 管理器中为曲线标注创建标注线，但它可以加载在 MapInfo Professional 中创建的标注线或使用自定义标注工具创建的标注线。

- 对于包含自定义（用户定义）线终点位置的标注，MapXtreme 可以为其加载标注行。例如，如果在 MapInfo Professional 中移动标注的终点并将此位置更改保存到工作空间 (.MWS)，则当在 MapXtreme 中打开该工作空间时将会如愿调整端点位置。MapXtreme 可以加载和保存自定义的线终点，但不支持线终点的创建。
- 在 MapInfo Professional 中创建并保存到 MWS 的曲线标注在 MapXtreme 中会显示在相同的位置。
- 在 MapInfo Professional 中拖放到新位置的标注在 MapXtreme 中可显示为曲线标注。如果将标注更改保存到 .MWS 工作空间并在 MapXtreme 中打开该工作空间，则当打开曲线标注选项时，这些标注将显示为曲线标注。然而，要注意的是：如果在 MapInfo Professional 中拖放该标注并在保存到工作空间之前旋转该标注，那么 MapXtreme 将无法弯曲该标注。这是因为 MapInfo Professional 已经修改标注的角度，而 MapXtreme 会据此角度确定曲线标注设置。

创建曲线标注

沿曲线放置标注：

- 在 Workspace 管理器中，打开要更改其标注的地图。
- 突出显示含有所要更改的标注的图层源。
- 验证那些标注是否可视。
- 单击“位置”选项卡。选中“旋转”部分中的沿线曲线标注，以显示曲线上的标注。



- 如有必要，可用“位置”选项卡中的“方向”按钮相对于标注的锚定点来设置其位置。如果选择：
左，则会从弧/折线的开头对曲线标注进行左对齐
中心，则会以弧/折线的中点作为曲线标注的中心
右，则会从弧/折线的末尾对曲线标注进行右对齐



标注的放置方式受折线长度的影响。折线越长，标注显示效果的可预见性就越强。

调整曲线标注的位置

可用主工具栏中的“标注”工具调整曲线标注的位置。对于要更改的标注源，请确保已选中“位置”选项卡上的弯曲区段上的标注选项。

1. 在列表中突出显示要移动其标注的图层。
 2. 选中“选项”选项卡中的可选复选框。
 3. 突出显示要从中移动曲线标注的标注源。

4. 单击主工具栏中的  (“标注”工具)。

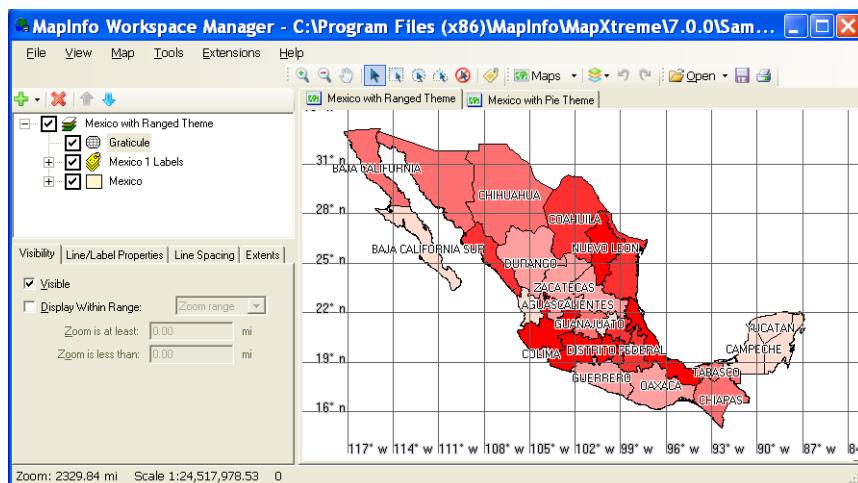
5. 单击要在其上重新放置标注的线条。

6. 单击该标注的新位置，直到对标注的位置满意为止。

 如果数据中没有与所选线段相关联的标注名称，则不会显示标注。

网格图层

网格是指以固定间距（例如每 5 度、每 15 度）覆盖在地图上的网格（纬线和经线）。网格用于建立参考框架。

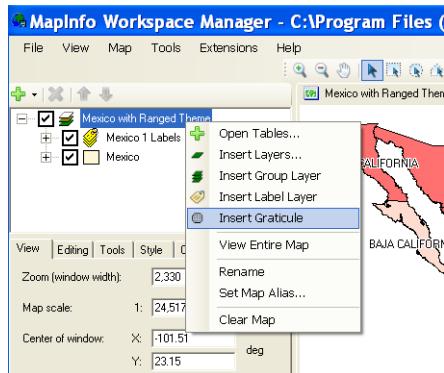


在许多方面，网格的行为与其他图层相似。但网格图层与其他类型图层有以下不同之处：

- 与其他类型的图层不同，**Graticule** 图层不可编辑。换而言之，不能为它添加新的图元。
 - 虽然可以配置网格标注样式和位置，但不能在网格图层上创建标注图层。
 - 不能在网格图层上创建主题图层。

添加网格图层

可以直接从图层控件来添加网格（通过右键单击地图节点可访问此上下文菜单）：



默认情况下，网格的间距和边界根据地图窗口的缩放比例和大小确定。因此，在向地图添加网格图层时，无论缩放级别如何，都会显示网格线。

管理网格图层

在添加网格图层后，可以在图层控件中通过许多方式管理和自定义该图层。

- 控制网格图层的可视性和缩放/比例设置
- 显示和配置主要和次要网格线的间距
- 控制网格标注和网格线的可视性和样式
- 指定网格（东西南北）的边界（单位为度、英尺或米）。
- 按图层控件顺序上下移动图层
- 删除网格图层（上下文菜单是删除网格图层的唯一方法）
- 创建多个网格图层和组图层

有关详细信息，请参阅[网格图层](#)。

网格图层由网格图标表示，如下所示：



使用 GeoDictionary 管理器

GeoDictionary 管理器是 MapXtreme 中的实用程序，用于地图绘制应用程序。本章说明如何使用该工具。

在本章中：

- ◆ 使用 GeoDictionary 管理器..... 426
- ◆ GeoDictionary 管理器中的变化..... 426
- ◆ GeoDictionary 管理器的用户界面..... 426
- ◆ GeoDictionary 文件..... 429

使用 GeoDictionary 管理器

GeoDictionary 管理器设计用于支持 GeoDictionary 文件的操控。GeoDictionary 是包含 MapInfo 表注册信息的 XML 文件，可以在数据自动绑定过程中和用户的应用程序匹配。只有可以或将要匹配的 MapInfo 表应该在 GeoDictionary 中注册。

不需要注册应用程序使用的 GeoDictionary 中的每个 .tab 文件，实际上，注册不必要的文件会增加系统开销。只需要注册那些想要匹配的表。

GeoDictionary 管理器中的变化

MapXtreme 的 GeoDictionary 管理器在外观上类似于包含在 MapX 和 MapXtreme 先前版本中的实用程序。主要区别是现在基础表文件 (*.dct) 是 XML 文件且完全可编辑（使用文本或 XML 编辑器）。

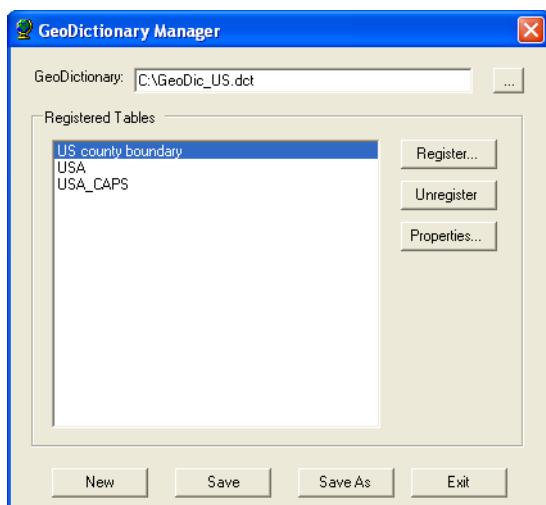
GeoDictionary 管理器的用户界面

本节描述 GeoDictionary 管理器的用户界面。

运行 GeoDictionary 管理器

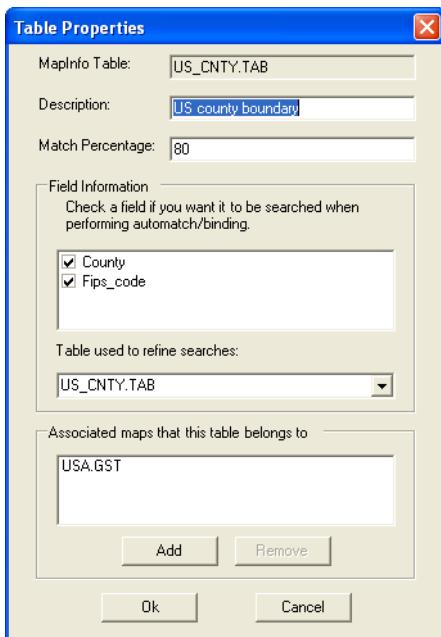
手动注册图层时运行 GeoDictionary 管理器：

- 开始 -> 所有程序 -> MapInfo -> MapXtreme -> GeoDictionaryManager。
显示 GeoDictionary 管理器。启动时重新加载最后打开的文件。



部分	描述
GeoDictionary	GeoDictionary 编辑框包含当前管理的 GeoDictionary 的完整路径。GeoDictionary 编辑框右侧的按钮允许用户浏览并管理另一个 GeoDictionary。二进制和 XML GeoDictionary 文件都可以打开，但只能保存 XML 格式。
已注册的表	“已注册的表”列表框包含了所有注册到 GeoDictionary 的表的好记的名称列表。突出显示特定的表以取消注册或修改它的属性。双击表弹出“属性”对话框以编辑该表的属性。
注册	单击“注册”按钮弹出普通“打开文件”对话框，对话框上的“文件类型”组合框设置为“MapInfo Tables (*.tab)”。从文件选取器中选定表后，显示该表的“表属性”对话框。如果在“打开文件”对话框中选定多个文件，则对于每个添加的表都打开一个“属性”对话框。
取消注册	单击该按钮，从 GeoDictionary 中删除选中的表。也可以选中多个表（通过按住 shift 键单击）取消注册。使用“取消注册”按钮不会将文件从硬盘删除。
属性	单击“属性”按钮，弹出选中表的“表属性”对话框。参阅以下的“表属性”对话框。
新建	“新建”按钮用于清除屏幕上的字段，这样可以创建新的 GeoDictionary 文件。选择关闭该 GeoDictionary 文件时，将提示是否保存更改。
保存	“保存”按钮用于将当前文件另存为 XML 格式，无论该文件被打开时的格式是什么。
另存为	“另存为”按钮用于将当前文件另存为新名称或新位置。
退出	“退出”按钮用于关闭应用程序。如果 GeoDictionary 文件已经修改，将提示是否保存更改。

使用“属性”对话框为给定表设置匹配表属性。要访问“属性”对话框，单击“属性”按钮，或双击“注册的表”列表中的任意表名。



部分	描述
MapInfo 表	如果 MapInfo 表文件位于 GeoDictionary 的相同目录下，则只读编辑框包含 MapInfo 表的文件名，否则包含该文件的相对路径名。如果该文件位于不同的驱动器或卷，则显示完整路径。
描述	该字段用于更改表的好记的名称。该控件默认为 .TAB 文件中的 Description 标记，如果无法找到 Description 标记则为文件名，但用户可以更改。注意，在 GeoDictionary 管理器中对描述的更改只存储在 GeoDictionary 中并且不反映在表自身。这使得 GeoDictionary 管理器可以很容易地使用只读数据，例如 CD-ROM 上的数据。
匹配百分比	该字段最初填充 GeoDictionary 的默认值，并可以通过更改该值而改变以适应特定的表。值必须介于 1 和 100 之间。
字段信息	该列表框包含表中所有索引列的列表。如果选中某给定列的框，则将在匹配进程中搜索该字段。
用于优化搜索的表	该字段使用户可以设置优化表，决定特定索引中不惟一数据的精确匹配。 某些表，例如 US Counties，包含不惟一的索引列。在这种情况下，需要优化表以决定数据的精确匹配。如果表包含不惟一的索引列，使用该字段指定用于匹配的表，以找到惟一的条目。

部分	描述
该表属于以下相关联的地图。	该列表框显示了包含该特定表的 Workspaces、GeoSets 或其它文件。
添加	显示了普通文件选取器，使用户可以选择 GeoSets、Workspaces 或与特定 Tab 文件关联的其它文件。选中一个或多个文件，将它们添加到关联地图列表。单击时，可以通过按住 shift 或 control 键来选择多个文件。
删除	删除选定的 geoset、workspace 或关联地图列表中的其它文件名。

GeoDictionary 文件

如果用户选择跳过使用 GeoDictionary 管理器，则可以手动操控 GeoDictionary 文件 (*.dct)。如果用户理解并可以熟练编写 XML，则该方法是使用 GeoDictionary 管理器之外的可行选择。.dct 文件是直接的 XML。

示例 .dct 文件

以下是 GeoDictionary 文件示例 GeoDic_US.dct。此文件是非常简单的 GeoDictionary，用于示范文件结构。

```
<?xml version="1.0" encoding="Windows-1252" standalone="yes"?>
<!--GeoDictionary file-->
<GeoDictionary>
    <DefaultMatchThreshold>80</DefaultMatchThreshold>
    <MatchTables>
        <MatchTable>
            <TablePath>US_CNTY.TAB</TablePath>
            <TableDescription>US county boundary</TableDescription>
            <RefineTableName>USA.TAB</RefineTableName>
            <MatchThreshold>90</MatchThreshold>
            <AssociatedMaps>
                <AssociatedMap>USA.GST</AssociatedMap>
            </AssociatedMaps>
            <MatchFields>
                <MatchField>
                    <FieldName>County</FieldName>
                </MatchField>
                <MatchField>
                    <FieldName>Fips_code</FieldName>
                </MatchField>
            </MatchFields>
        </MatchTable>
    </MatchTables>
```

</GeoDictionary>

结构中的元素对应 GeoDictionary 管理器中的独立字段和控件，如以下表格中定义：

XML 元素	表属性对话框的字段	描述
GeoDictionary	无	文件的根元素
DefaultMatchThreshold	无	特定的 MatchThreshold 不是为特定的表定义时，DefaultMatchThreshold 是匹配的阈值。 该值不可以使用 GeoDictionary 管理器设置。
MatchTables	无	注册的 MatchTables 的容器元素。对于“已注册表”列表中的每一个项目，应该都有单一 MatchTable 元素与之对应。
MatchTable	MapInfo 表。	每个要匹配的表的文件名。
TablePath	MapInfo 表。	每个 MatchTable 的相对路径。
TableDescription	说明字段（也是出现在“已注册表”列表中的名称，该列表位于主对话框内）。	MatchTable 的好记的名称。
RefineTableName	用于优化搜索的表	用于优化匹配的相关表。
MatchThreshold	匹配百分比	该 MatchTable 的匹配阈值。如果没有指定值，则使用 DefaultMatchThreshold。
AssociatedMaps, AssociatedMap	该表属于以下相关联的地图。	该列表框显示了包含该特定表的 Workspaces 和 GeoSets。
MatchFields	无	该元素是 MatchField 和 FieldName 元素的容器。
MatchField, FieldName	字段信息列表框。	搜索列表框中选中的每个项目用于实现自动匹配和绑定。

A

如何创建和部署 **MapXtreme** 应用程序

本附录是设计用于演示如何简化 MapXtreme 应用程序的创建、打包和部署的教程。在本教程结尾处，您将查看到成功开发、打包和部署已实现地图绘制应用程序的所有步骤。

本教程假定您已成功安装 Visual Studio 并熟悉该产品的使用。如果不熟悉 Visual Studio，则可以从 Microsoft MSDN web 页面获得相关资源。详情请查看 msdn.microsoft.com/developerscenter/。此外，还应已安装 MapXtreme。本文档包含的代码片断通过 Visual Basic 实现。对于多数示例应用程序，我们还提供了 C# .NET 示例。

在本附录中：

- ◆ 定制 **MapXtreme** 示例 432
- ◆ 构建桌面应用程序 432
- ◆ 生成 Web 应用程序 441

定制 MapXtreme 示例

本附录包含两个教程：

- 构建桌面应用程序
- 生成 Web 应用程序

这两个教程指导您完成 MapXtreme 桌面应用程序和 Web 应用程序的定制过程并做好部署的准备。

构建桌面应用程序

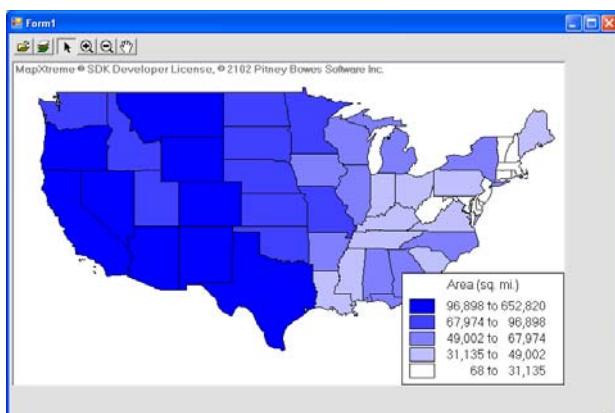
我们将以 MapXtreme 提供的一个主题示例应用程序为基础构建我们的桌面应用程序。此桌面示例应用程序已经是可正常工作的地图绘制应用程序。只需稍加修改，您便可以定制示例应用程序使之满足您自己的需要。您可以使用本教程中介绍的修改作为定制任何 MapXtreme 桌面项目的模型。

MapXtreme 提供的所有桌面示例应用程序都通过该产品提供的地图应用程序模板创建。有关详细信息，请参阅在 [Visual Studio 中创建应用程序](#)。

运行示例应用程序

首先，我们运行 ThemeLegend 应用程序来看看该程序是什么样子。

1. 在 Visual Studio 中，选择文件 > 打开 > 项目/解决方案，然后为名为 ThemeLegend 的示例应用程序打开项目。此项目的默认路径是：
`C:\Program Files\MapInfo\MapXtreme\7.x.x\Samples\VisualStudio\Desktop\Features\ThemeLegend\vb\ThemeLegendVB.vbproj`
(此处 x 是您使用的 MapXtreme 版本的版本号。)
2. 选择调试 > 开始调试，或按 F5 运行该应用程序。



应用程序显示美国的标准地图，以各州的陆地面积为主题。较大州加上蓝色阴影，较小州则加上白色阴影。该图例显示为陆地面积值的每个范围分配的颜色。

应用程序提供的工具让您能够执行某些基本的功能。您可以打开表，放大或缩小特定区域，在地图上选择区域，移动地图，访问“图层控件”以便为每个图层应用特定的设置。您可能希望实际使用这些工具来了解应用程序的工作方式以及用户的体验。

- 完成后，关闭活动的 Form1 窗口，结束该应用程序并返回到“设计”模式。

修改应用程序

下一步，我们将对 **ThemeLegend** 代码进行修改以说明如何定制示例应用程序。我们将更改该应用程序的以下元素：

- 数据源
- 范围（容器）数
- 范围类型
- 范围颜色
- 图例位置
- 工具栏

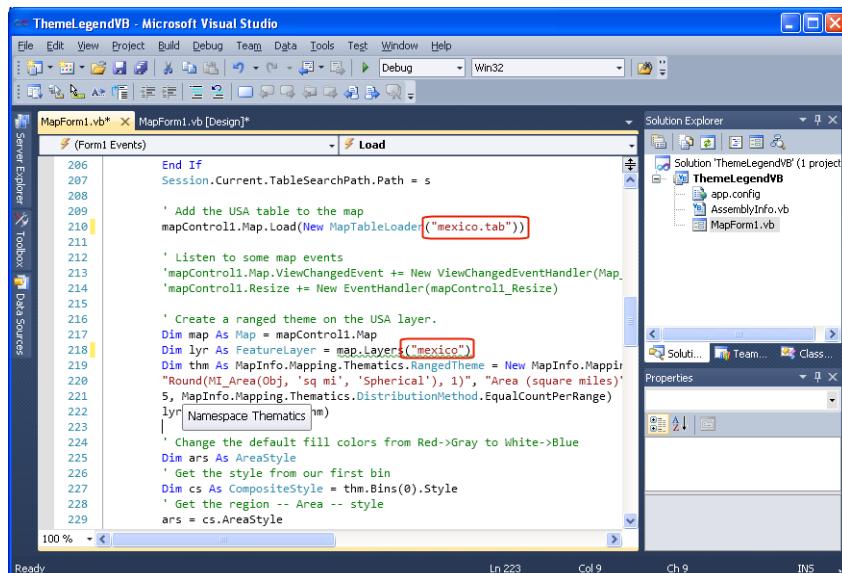
1. 在 Visual Studio 中，将鼠标指针移动到“解决方案资源管理器”并右键单击 **MapForm1.vb**。选择“查看代码”，显示代码页。

如果已在设计模式中显示 **MapForm1.vb**，则可在该窗体的任意位置右键单击并选择“查看代码”。

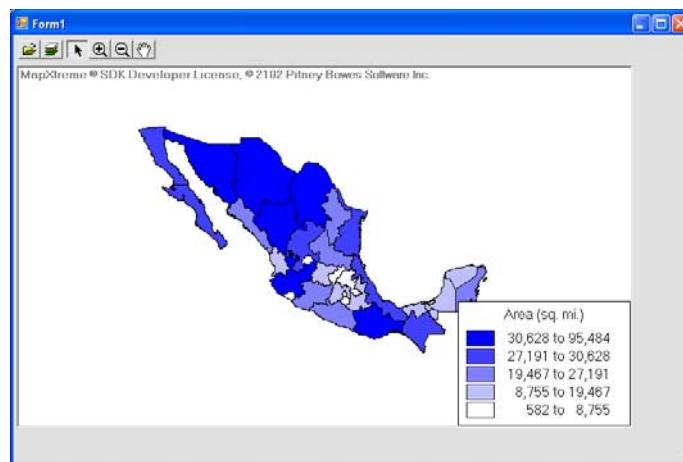


本教程中讨论的所有示例应用程序都取自 **MapXtreme** 发行光盘。引用的行号可能与安装中的代码不完全对应。

2. 此时将出现新选项卡，显示应用程序的代码。向下滚动代码页，熟悉代码。注意：应用程序中的大部分操作都在 **Form1()** 类中执行。这也正是我们着手修改的地方。
3. 我们来更改作为主题图基准的数据源列。为此，我们需要更改应用程序使用的 **mexico** 表格的名称并指定不同的 **FeatureLayer**。需要执行以下更改：
 - 在第 210 行，将表从 **usa.tab** 更改为 **mexico.tab**。
这将加载墨西哥地图，而不是美国地图。
 - 在第 218 行，将 **FeatureLayer** 从 **usa** 更改为 **mexico**。
4. 下图以红色方框突出显示代码的变动。



5. 选择调试 > 开始调试（或按 F5）运行该应用程序并查看所作更改。完成后关闭 Form1 窗口。

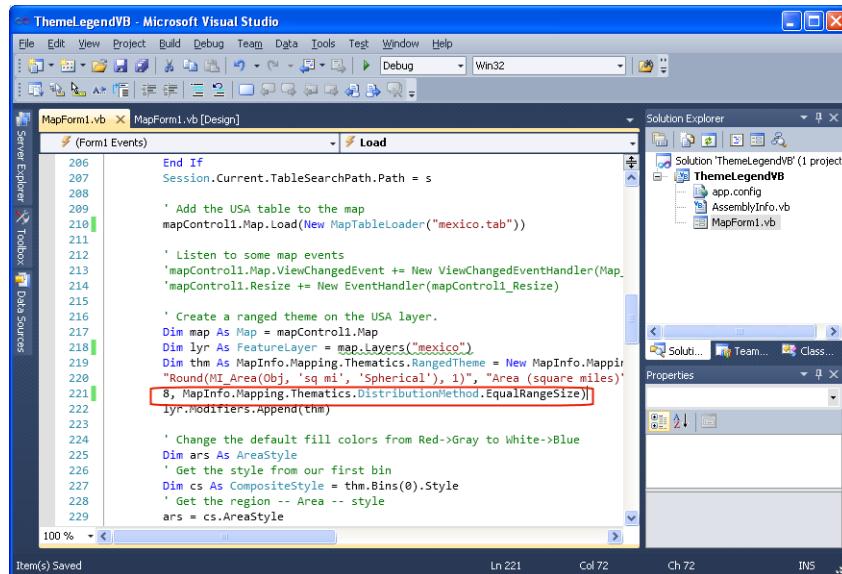


应用程序使用 Mexico 数据创建相同类型的主题。通过更改表名和 FeatureLayer，您可以传入适合该主题的任何数据源。

下一步，我们来更改主题本身。通过控制传递到 RangedTheme 构造函数的参数，我们可以更改主题图的计算和显示方式。

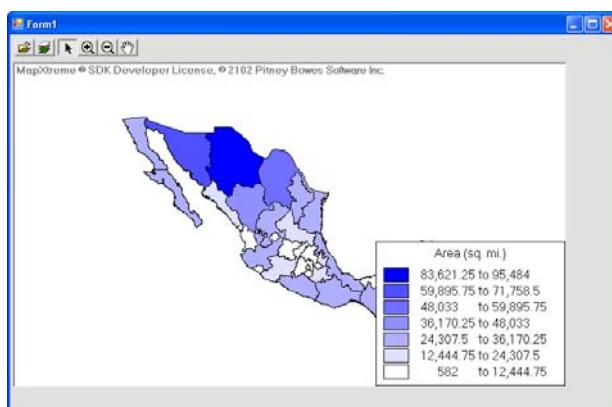
6. 突出显示代码中的单词 RangedTheme 并按 F1（第 212 行）。屏幕上将显示 RangedTheme 的帮助主题。您可以阅读此主题以了解每个参数在传递给构造函数时的行为。

7. 我们将增加主题中的范围（容器）数。在第 221 行，将容器数从 5 更改为 8。（容器是由最大值和最小值组成的范围，为方便添加阴影，可以使用容器来组织主题，就像组织数值一样）。
8. 下一步，我们更改范围的分布方法。分布方法规定范围的计算方法。在第 221 行，将 EqualCountPerRange 更改为 EqualRangeSize。在 EqualRangeSize 分布方法中，每个范围的值数目都相等。
9. 下图以红色方框突出显示代码的变动。



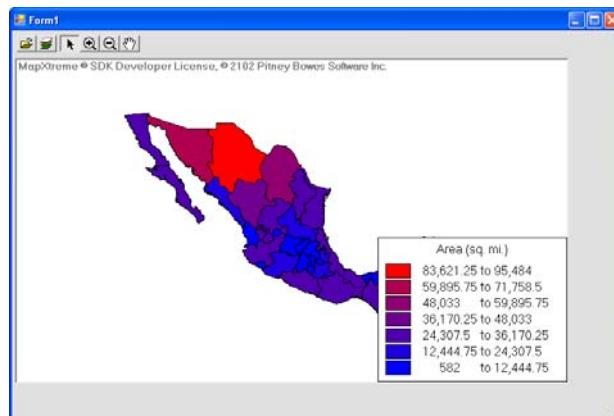
```
206 End If
207 Session.Current.TableSearchPath.Path = s
208
209 ' Add the USA table to the map
210 mapControl1.Map.Load(New MapTableLoader("mexico.tab"))
211
212 ' Listen to some map events
213 'mapControl1.Map.ViewChangedEvent += New ViewChangedEventHandler(Map_
214 'mapControl1.Resize += New EventHandler(mapControl1_Resize)
215
216 ' Create a ranged theme on the USA layer.
217 Dim map As Map = mapControl1.Map
218 Dim lyr As FeatureLayer = map.Layers("mexico")
219 Dim thm As MapInfo.Mapping.Thematics.RangedTheme = New MapInfo.Mapping.
220 "Round(ML_Area(Obj, 'sq mi', 'Spherical'), 1)", "Area (square miles")
221 lyr.DistributionMethod = MapInfo.Mapping.Thematics.DistributionMethod.EqualRangeSize
222 lyr.Modifiers.Append(thm)
223
224 ' Change the default fill colors from Red->Gray to White->Blue
225 Dim ars As AreaStyle
226 ' Get the style from our first bin
227 Dim cs As CompositeStyle = thm.Bins(0).Style
228 ' Get the region -- Area -- style
229 ars = cs.AreaStyle
```

10. 选择调试 > 开始调试（或按 F5）以生成并运行修改后的应用程序。注意：主题容器的数目已增加，分布方法已更改。完成后关闭 Form1 窗口。



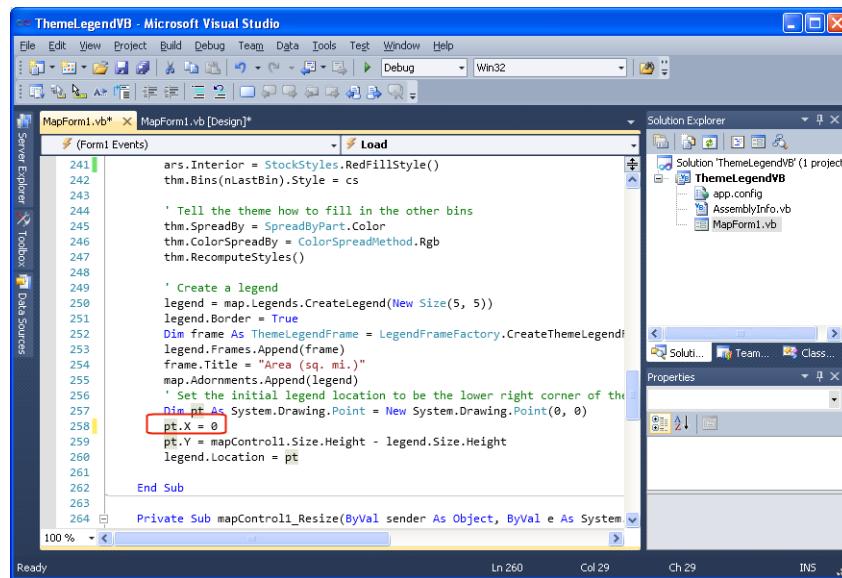
下一步，我们更改范围的颜色。修改颜色有两种方法。可以为每个容器分别设置特定的颜色，也可以为第一个容器和最后一个容器设置颜色。中间容器的阴影将从第一个容器的颜色渐变为最后一个容器的颜色。目前，此代码指定颜色范围为从白色到蓝色。我们将颜色范围从蓝色更改为红色。

11. 要更改范围的颜色，需要编辑填充样式颜色。请确保显示代码视图并作出以下更改：
 - 要更改第一个容器的颜色，请在第 231 行，将 `WhiteFillStyle` 更改为 `BlueFillStyle`
 - 要更改第二个容器的颜色，请在第 241 行，将 `BlueFillStyle` 更改为 `RedFillStyle`
12. 选择调试 > 开始调试（或按 F5）以生成并运行修改后的应用程序。地图的颜色已更改，从而反映了我们的新设置。完成后关闭 Form1 窗口。

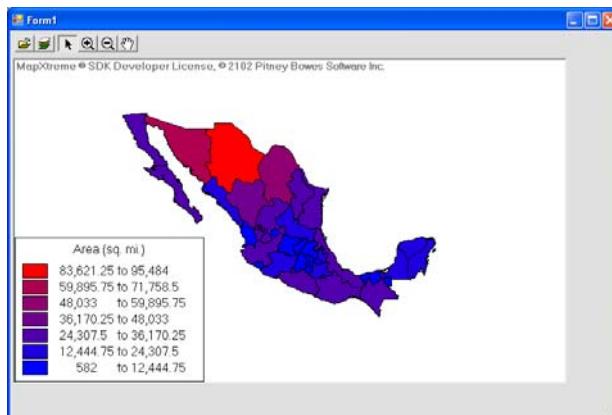


由于进行了大量修改，现在，该图例部分遮挡了地图。我们将移动该图例，以便能够看到完整的地图。我们可以使用平移工具来移动地图，但平移并不能永久性更改应用程序。只有当该应用程序仍在运行时，新的地图位置才会生效。取而代之的是，我们将通过编程的方式改变图例的位置。

13. 请确保已显示代码页并对图例位置作出以下更改：
 - 在第 258 行，将 “`mapControl1.Size.Width - legend.Size.Width`” 替换为 “0”
这将图例位置的 X 坐标更改到框架的左侧。而 Y 坐标保持不变。
14. 下图以红色方框突出显示代码的变动。

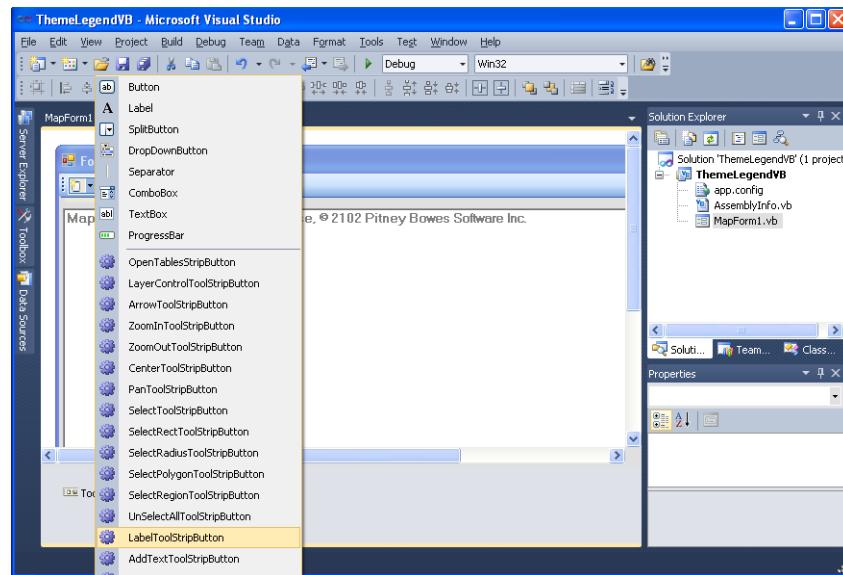


15. 选择调试 > 开始调试（或按 F5）生成并运行该应用程序。此时，图例位于新的位置。完成后关闭 Form1 窗口。

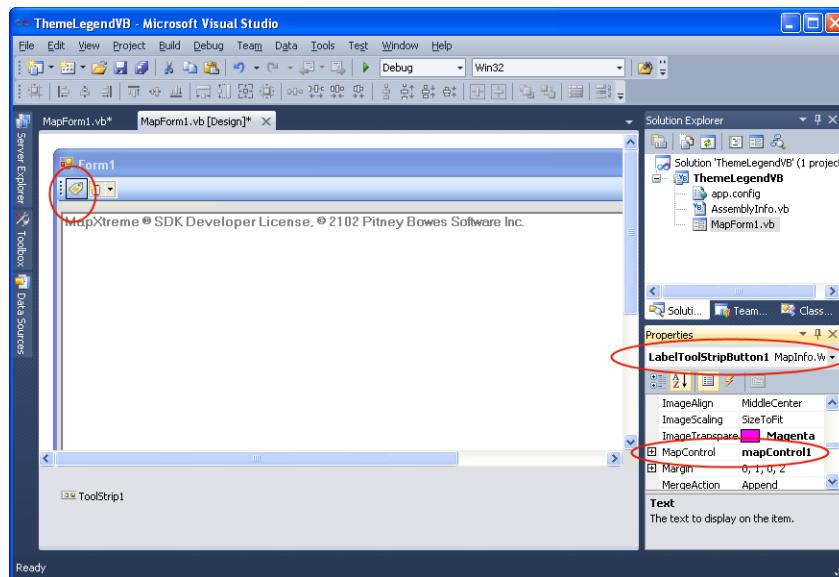


我们所作的最后一处修改是向工具栏中添加标注工具。

16. 在 Visual Studio 中，以“设计”模式打开 MapForm1.vb。
17. 从工具箱中的“菜单和工具栏”组中选择“ToolStrip”，然后在想要添加“标注”工具的 MapControl 上单击它。随后出现划分按钮。
18. 单击划分按钮上的向下箭头并在按钮列表中选择 LabelToolStripButton。



注意，“标注”工具会显示在左上角位置的窗体上并自动分配到“属性”窗口中的 mapControl1。



如您所见，可在这些示例的基础上进行定制设置。您可以方便地代之以自己的数据，修改主题并添加工具。可对该主题图地图执行的其它修改包括：

- 使用该数据源（表）中的另一列，即可让该主题基于另一个值，例如 **population**。不要忘记更新要匹配的图例。
- 增加其它主题图和图例。
- 添加附加工具。

在发行模式下生成

完成修改后，即已做好建立发行版的准备。发行版是指即将面向客户或内部用户发行的应用程序版本。要制作发行版，请执行以下操作：

1. 选择生成 > 配置管理器。
2. 在活动解决方案配置下拉列表中，将生成类型更改为发行。
3. 在“配置管理器”对话框中，单击关闭。
4. 在“解决方案资源管理器”中，右键单击 ThemeLegendVB 项目并选择“生成”。

完成后，请查阅打包桌面应用程序。

打包桌面应用程序

此部分教程介绍如何打包桌面应用程序。程序包是指软件产品所需文件和目录的集合。产品必须生成一个或多个程序包，以便将其转移到分发介质上（例如 CD-ROM 或 DVD-ROM）。该桌面应用程序的程序包由 Setup.exe 文件组成，该文件包含安装和运行桌面应用程序所需的全部文件。

MapXtreme 设计用于尽可能简化此过程。使用 Visual Studio 中的功能及 MapXtreme 中的自动操作，可将正确的合并模块加入程序包中。合并模块 (MSM) 是一个独立的程序包，包含了安装组件必需的所有文件、资源、注册表项和安装逻辑。在部署应用程序中可以查找到 MSM 列表以及 MapXtreme 中运行时安装程序和其它可用选项的相关讨论。

Visual Studio 2012 没有对安装项目的支持，因此您需要使用第三方工具创建一个。可以通过安装“Visual Studio Installer Projects Extension”，从而在 Visual Studio 2013 中启用安装项目的创建。有关更多信息，请参阅：

<https://visualstudiogallery.msdn.microsoft.com/9abe329c-9bba-44a1-be59-0fbf6151054d>

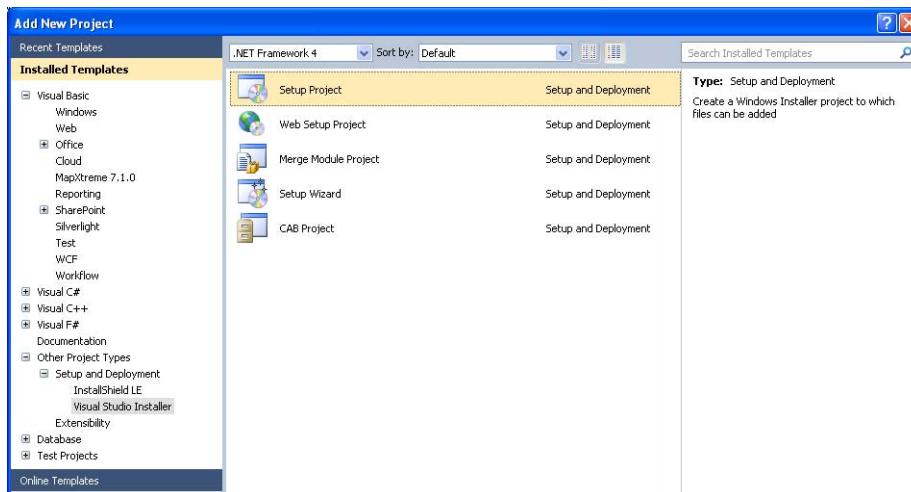
或

<http://blogs.msdn.com/b/visualstudio/archive/2014/04/17/visual-studio-installer-projects-extension.aspx>

创建安装项目

首先需要为解决方案创建安装项目。通过安装项目，您可以创建 Setup.exe 文件。

1. 在“解决方案资源管理器”中，单击以突出显示解决方案的名称并选择文件 > 添加新项目。屏幕上将显示“添加新项目”对话框。
2. 在项目类型列表中，展开其它项目类型，然后单击安装和部署并选择“Visual Studio 安装程序”。如果您的项目不是解决方案的一部分，则不会显示“添加项目”快捷方式。您可以从“文件”>“添加项目”中添加“安装”项目，然后选择“安装”。将创建解决方案和“安装”项目。
3. 突出显示安装项目。为安装项目输入名称和位置并单击确定。此时，Visual Studio 将显示新安装项目的文件系统。



4. 下一步，需要添加数据文件。在“解决方案资源管理器”中，右键单击安装项目，然后选择菜单上的添加 > 文件。
5. 在添加>文件对话框中，选择文件类型框中下拉列表上的“所有文件”，然后导航至数据文件所在位置。
6. 选择需要添加到安装项目的所有文件。
7. 单击打开。这些文件将添加到“解决方案资源管理器”中 ThemeLegendVB 项目下。
8. 若添加到安装项目的文件处在不同路径中，请重复 5-8 步骤。



如果要将该应用程序部署到另一台计算机，则开发计算机和部署计算机上数据的绝对路径必须一致。

要查看或更改安装文件写入的路径：

1. 右键单击安装项目并选择菜单上的属性。
2. 在“安装属性页面”上，可更改“输出文件名称”。也可在此页面上选择配置（调试或发行）生成安装文件。

若希望将“开始”菜单中的生成文件放置在“文件系统选项卡”上，右键单击“用户程序菜单”，单击“用户程序菜单”的“创建”快捷键，然后设置需要的名称和属性。或者，可写入脚本，将程序放置在正是您想放置的开始菜单上（例如，在子菜单上）。

同样，要把快捷键放置到用户桌面上，右键单击用户桌面，单击“用户程序菜单”的“创建”快捷键，然后设置需要的名称和属性。或者，可写入脚本，创建快捷键并将其放置在用户桌面上。

添加许可证文件

下一步，需要为应用程序提供桌面许可证文件 (MapXtremeDesktop.lic)，以便用户能够使用该应用程序。请参阅[用于部署的运行时许可证](#)。

为说明如何添加许可证文件，我们将在该程序包中加入一份试用许可证。¹在打包自己的应用程序时，需要提供运行时许可证。

1. 在 Visual Studio 界面中，单击文件系统（安装）选项卡。
2. 右键单击应用程序文件夹图标，然后选择添加 > 文件。
3. 在“添加文件”对话框中，导航至 C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x) 并选择 MapXtremeTrial.lic。单击打开。

此时，文件增加到安装项目并将与其它应用程序文件共同安装到应用程序文件夹。

可选设置：添加 **Workspace** 管理器和 **GeoDictionary** 管理器

如果需要，可在程序包中包括 MapXtreme 实用程序 Workspace 管理器和 GeoDictionary 管理器，以便客户能够使用它们。按照[添加许可证文件](#)下的步骤操作。用 <MapXtreme installation folder>\WorkspaceManager.exe 和 GeoDictionaryManager.exe 代替 MapXtreme.lic。

生成安装项目

在生成安装项目后，将会创建一个 **Setup.exe** 文件，该文件包含该项目所需的所有数据、生成代码和必备的 MSM。此安装项目是 MapXtreme 解决方案的一部分。

生成安装项目后继续生成解决方案。在生成整个解决方案之后，Visual Studio 将编译并生成地图应用程序，然后编译并生成用于执行安装的 **Setup.exe** 文件。现在，您的程序包已完成并已做好部署的准备。

部署桌面应用程序

在生成桌面应用程序并将所有组件打包以便进行部署之后，最后一步便是将程序包传送到生产工作站中。

要部署应用程序，必须向最终用户提供之前创建的 **Setup.exe** 文件。您可以将 **Setup.exe** 文件复制到 CD-ROM 或 DVD-ROM，以此方式提供该文件。用户在计算机上启动 **Setup.exe** 可执行文件时，安装开始并逐步引导他们完成安装步骤。以上就是本部分全部内容！

生成 Web 应用程序

本教程中，我们的应用程序将以 MapXtreme 提供的 **Web Thematics** 示例为基础。这些 MapXtreme Web 示例都是 Visual Studio Web 应用程序项目。它们包含一个能够完全正常运行的 Web 应用程序，包括为支持该应用程序和处理状态管理而预加载的数据。

1. 如需为部署创建程序包，则 MapXtremeTrial.lic 必须来自 SDK 或 MapXtreme 的运行时安装。使用 MapXtreme 的评估版本无法创建安装项目。

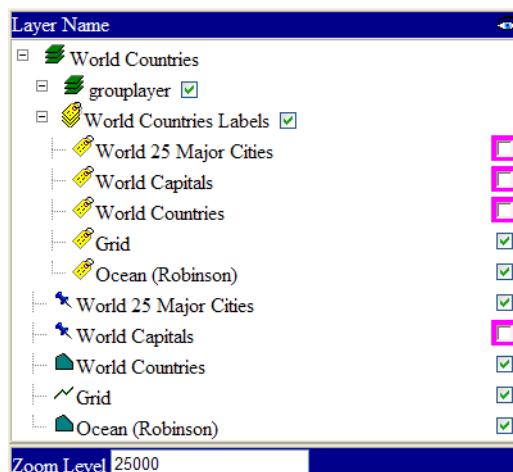
运行示例 Web 应用程序

首先，我们运行 Thematics Web 示例来看看该程序是什么样子。

1. 在 Visual Studio 中，选择文件 > 打开项目，然后打开 ThematicsVB 示例所在的文件夹。此文件夹的默认路径是：C:\Program Files\MapInfo\MapXtreme\7.x.x\Samples\Web\Features\Thematics\ThematicsVB（此处 x 是所用 MapXtreme 版本的版本号。）
2. 选择调试 > 开始调试（或按 F5，在“调试”模式下运行该示例）。



Thematics Web 应用程序示例项目包含一张世界背景地图、一个下拉菜单（列出待应用的各种主题图）、MS Access 表的属性列以及应用这些主题图的“应用”按钮。该 Web 页面包含 MapXtreme Web 模板中包含的标准工具按钮：ZoomIn、ZoomOut、Center 和 Pan。该应用程序还包含图层控件，方便您设置哪些图层可见并查看当前地图缩放级别。



有关详细信息，请参阅 [MapXtreme Web 应用程序组件](#)。

完成后，关闭该 Web 页面，但仍保持打开该项目。

修改应用程序

要修改此示例应用程序，我们来查看代码隐藏页面 `WebForm1.aspx.vb`。代码隐藏页面是展示 Web 窗体的 HTML 的代码。

① 此 Thematics Web 应用程序示例的设计与其桌面应用程序示例版本截然不同。最重要的区别是 Web 示例与数据表和数据列名称密切相关。这些项目的修改不像桌面示例中那样简单。

在代码隐藏页面中，我们会对主题属性值和范围颜色稍作修改。这些修改会改变用户运行该方法时的主题显示效果，这里我们要修改的是位于第 244 行的 `CreateThemeOrModifier` 方法。第一处修改是针对点密度主题及其显示方式。

1. 在 Visual Studio 中，将鼠标指针移动到“解决方案资源管理器”并右键单击 `WebForm1.aspx`。选择“查看代码”，显示代码隐藏页 `WebForm1.aspx.vb`。

如果已在设计模式中显示 `WebForm1.aspx`，则可在该窗体的任意位置右键单击并选择“查看代码”。

① 本教程中讨论的所有示例应用程序都取自 MapXtreme 发行光盘。引用的行号可能与安装中的代码不完全对应。

向下滚动代码页，熟悉代码。

2. 需要执行以下更改：

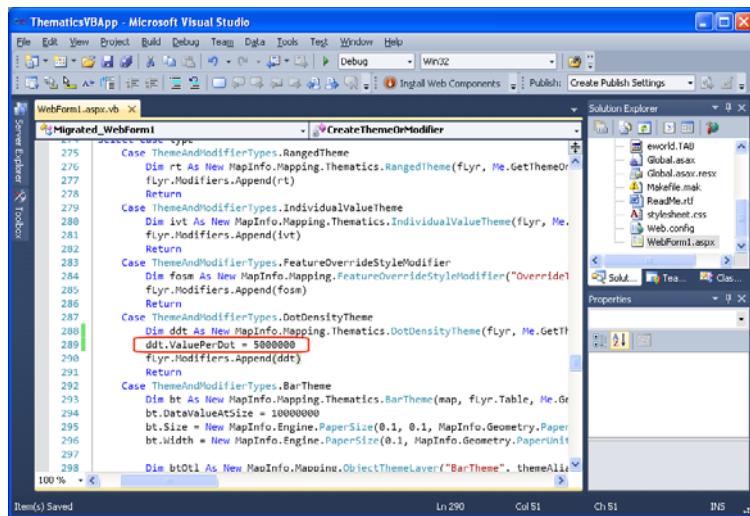
- 在 `ThemeAndModifierTypes.DotDensityTheme` 语句下，跳至第 288 行并将 `Color.Purple` 更改为 `Color.DarkGreen`

运行该应用程序时，此修改会改变选择点密度主题时点的显示颜色。

- 在第 289 行，将 `ValuePerDot` 属性从 2000000 更改为 5000000。

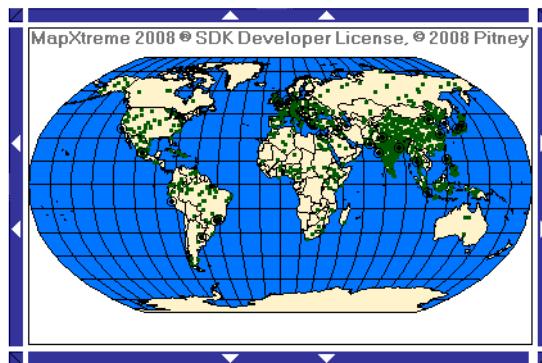
这将更改根据人口显示的点数。结果是点数减少。

在下图中，代码更改以红色方框突出显示。



- 选择调试 > 开始调试（或按 F5）生成并运行该 Web 应用程序。在浏览器中显示该页面后，选择 DotDensityTheme 主题，然后检查 Pop_1994。

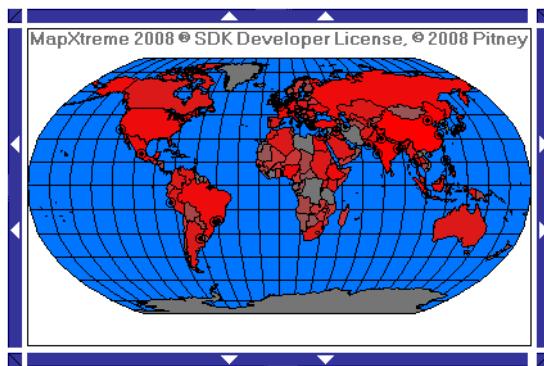
地图上点的颜色现在是深绿色。



现在我们对该 Web 示例进行更多简单的修改。现在更改的对象是范围主题。

- 需要执行以下更改：
 - 在 `ThemeAndModifierTypes.RangedTheme` case 语句下，跳至第 276 行并将容器数参数从 5 改为 8。
这将更改该主题中数据容器的数目。
 - 仍是第 276 行，将容器分布方法从 `EqualCountPerRange` 更改为 `NaturalBreak`。
这将更改将数据划分为数据容器的方法。
- 选择调试 > 开始调试（或按 F5）生成并运行包含这些新更改的 Web 应用程序。显示页面后，选择 RangedTheme，然后检查 Pop_1994。

注意如何更改主题使之包含更多数据容器（颜色代码）以及如何根据新的数据分布方法更改国家的主题颜色。



如您所见，相对来说，简单地修改该示例 Web 应用程序会比较方便。还可对该示例进行其他修改，包括：

- 更改网页布局，例如四处移动控件。
- 更改网页样式，例如颜色和字体。

关于状态管理的考虑

定制 Web 应用程序示例不仅设计到更改应用程序的元素，还应牢记使用最佳状态管理操作生成该应用程序。由于在 Web 环境下处理应用程序和用户状态的复杂性，所以必须了解 MapXtreme 是如何处理它们以及如何在自己的开发中应用这些概念。有关对 Web 应用程序“状态管理”的讨论，请参阅[第 6 章：了解状态管理](#)。

最佳实践中的这个示例，使用预载的地图工作空间和设置，手动管理应用程序和用户状态并使用池会话对象，使应用程序能够有效地服务于多项请求。

其设计目的是检测用户是首次发出请求（这种情况下应用程序显示其初始状态）还是重访该应用程序（此时应保留用户的个人设置）。由于本示例与多用户共享会话对象（称作对象池），因此它会检测会话时位于其初始状态（“干净的”）还是包含其他用户的更改（“不干净的”）。

为分别维护所有用户设置，此示例将 `Session.State` 设置为 `manual`。在 MapXtreme 会话状态自动保存的 Web 应用程序中，`Session.State` 应设置为 `HttpSessionState`。这意味着整个会话都将保存到 HTTP 会话中。此选项虽然会自动保存状态，但由于它不能确定要保存哪些状态，因而会降低性能。整个会话都会得到保存。

配置发行模式

完成修改后，即已做好建立发行版的准备。

ASP.NET 2.0 中需要理解的一个重要概念是 Visual Studio 并不参与编译 Web 应用程序项目。以前，ASP.NET 1.1 和 Visual Studio .NET 会生成代码隐藏源代码和 Web 窗体并将编译的代码存放到输出目录中。Visual Studio 在 ASP.NET 平台上执行编译。Visual Studio 生成命令仅校验 Web 应用程序配置。

要在不打开调试功能的前提下配置待打包的 Web 应用程序项目，可在 Web 应用程序项目的 Web.config 文件中更改配置设置。在 Web.config 中，查找 XML 元素“compilation”（第 27 行）并将属性“debug”的值更改为“false”。其结果应如下所示：

```
<compilation defaultLanguage="vb" debug="false">
```

此 Web.config 设置通知 ASP.NET 编译器以发行模式编译所有源代码。

要生成应用程序，请在“解决方案资源管理器”中右键单击 ThematicsVB 项目并选择“生成 Web 应用程序”。

打包 Web 应用程序

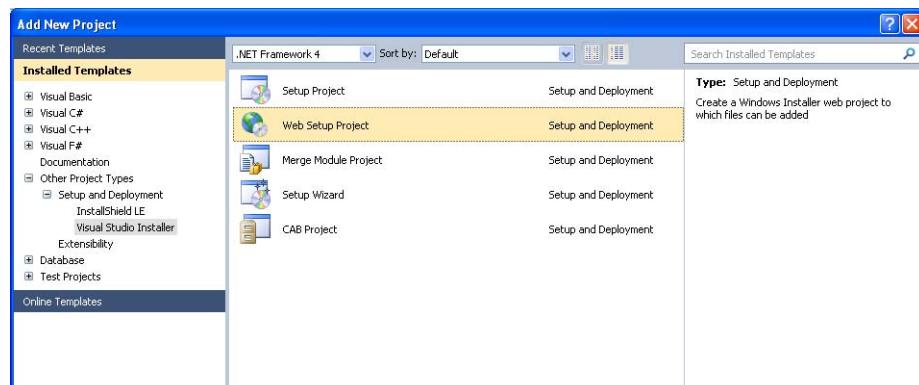
为 Web 应用程序创建程序包与为桌面应用程序创建程序包类似。

创建 Web 安装项目

首先，将 Web 安装项目添加到解决方案。要创建 Web 安装项目，请执行以下操作：

1. 在“解决方案资源管理器”中，右键单击“解决方案”，然后选择添加 > 新项目。屏幕上将显示“添加新项目”对话框。
2. 在项目类型列表中，展开其他项目类型，然后单击安装和部署。
3. 在“安装和部署项目”中，选择 Web 安装项目。单击确定。此时将创建一个 Web 安装应用程序项目。

如果您的项目不是解决方案的一部分，则不会显示“添加项目”快捷方式。您可以从“文件”>“添加项目”中添加“安装”项目，然后选择“安装”。将创建解决方案和“安装”项目。

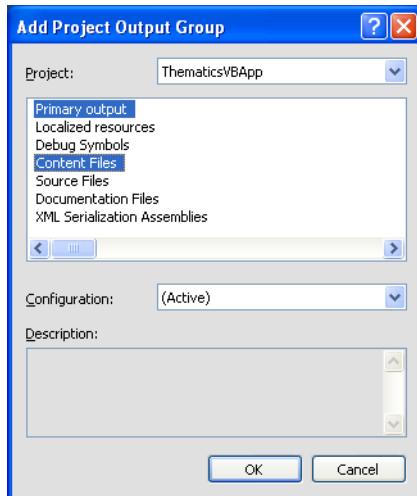


下一步，指示要包括解决方案的哪些部分。

4. 在“解决方案资源管理器”中，右键单击新创建的 Web 安装项目并选择添加 > 项目输出。屏幕上将显示“添加项目输出组”对话框。

5. 单击内容文件，然后单击确定。

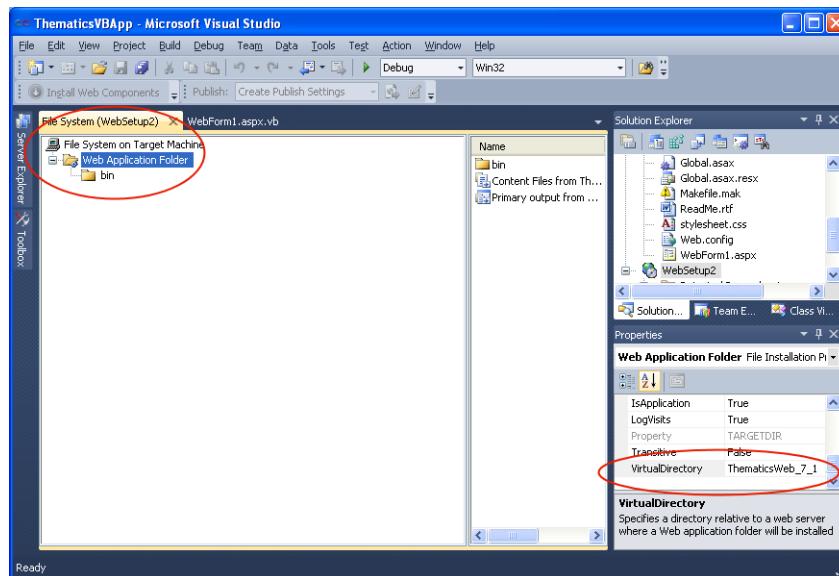
必须包括内容文件，因为该应用程序中 Web 页面的生成形式是 HTML 文件，而桌面应用程序中是可执行文件。



6. 如果要为 Web 应用程序创建自己的部署，则需要手动添加 MSM。有关详细信息，请参阅[部署 web 应用程序](#)。

7. 下一步，必须将数据发布到生产服务器上。您可以选择将数据手动复制到生产服务器上，也可在独立的安装程序中安装数据。

8. 在 Web 安装项目 Visual Studio 中的“文件系统”窗口中，右键单击 Web 应用程序文件夹，然后打开“属性”窗口。将“虚拟目录”名称更改为 ThematicsWeb_7_0，它定义了目标计算机上的虚拟目录。

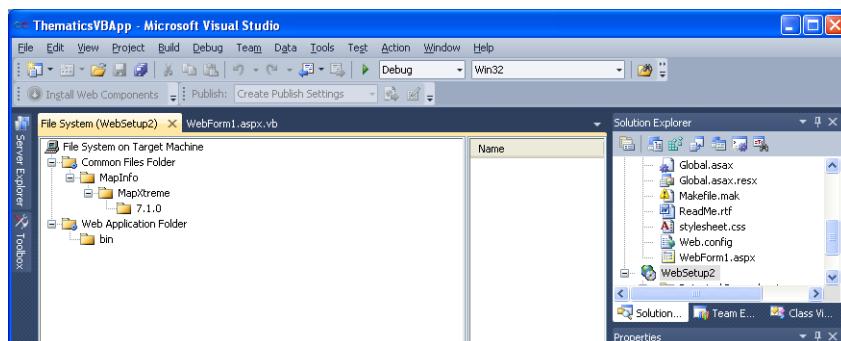


添加许可证文件

在打包 MapXtreme Web 应用程序时，您需要有效的 **MapXtremeWeb.lic** 副本才能让用户访问任何地图。有关部署许可的详细信息请参阅[用于部署的运行时许可证](#)。在该示例中，使用了与¹ 版本对应的试用许可**MapXtreme**。

- 首先创建许可证文件的位置。此 **MapXtreme** 安装中，我们希望重复 **MapXtreme** 试用许可证的路径。此路径是：
C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x
在“文件系统”窗口中，右键单击目标计算机上的文件系统并选择“添加特殊文件夹”>“Common Files”文件夹”。
- 右键单击 **Common Files** 文件夹并选择“添加 > 文件夹”。将文件夹重命名为“**MapInfo**”。
- 添加连续文件夹的方法是：右键单击路径中最后一个文件夹，选择“添加”>“文件夹”，根据文件夹结构重命名新创建的文件夹。您的文件树看上去应与下图中的文件树相似。

1. 如需为部署创建程序包，则 **MapXtremeTrial.lic** 必须来自 SDK 或 **MapXtreme** 的运行时安装。使用 **MapXtreme** 的评估版本无法创建安装项目。



4. 右键单击 7.x.x 文件夹并选择“添加”>“文件”。导航至 MapXtreme 试用许可证文件 (MapXtremeTrial.lic) 并将其添加到 7.x.x 文件夹。MapXtreme 试用许可证文件位于：C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x)。

生成 Web 安装项目

现在，您已做好生成 Web 安装项目的准备。

1. 选择生成 > 批处理文件生成。屏幕上将显示“批处理文件生成”对话框。
2. 查找 Web 应用程序项目的发行配置并选中相应的生成复选框。
3. 单击生成。

完成生成后，使用 Windows Explorer 转至 Web 安装项目的 Release 目录。您将看到一个 Setup.exe 文件或 setup.msi 文件。使用此文件进行部署。

部署 Web 应用程序

在 Web 部署中，Setup.exe 文件必须由托管 Web 应用程序的组织在生产服务器上运行。Setup 文件必须从开发服务器转移到生产服务器，然后安装在生产服务器上。在小型组织中，此部署可能由开发人员完成。在较大的组织中，可能由管理员执行此部署。之后，用户可以通过在 Internet 浏览器中输入站点的 URL 来访问此站点。有关不同安装程序、访问数据、代理服务器和权限的详细信息，请参阅[部署应用程序](#)。

本教程假定由您（作为开发人员）执行此部署。生产服务器必须已安装 .NET Framework v.2.0 或更新版本以及 MapXtreme。

在生产服务器上安装应用程序并将许可证文件置于适当的位置之后，修改 Web.config 文件，以便能够在本地计算机上引用预加载的工作空间。

部署 Web 应用程序时，安装应用程序在 IIS 中创建所需的虚拟目录并将所有其他必要组件放到正确位置。本应用程序中，许可证文件存放在 Program Files\Common Files\MapInfo\MapXtreme\7.x.x 目录下。

要查看 Web 应用程序的运行状况，请打开浏览器并输入 URL:

`http://<ProductionServerMachine>/ThematicsWeb_7_0/`。此 URL 应为 Web 应用程序在生产服务器上的位置，它使用您打包应用程序时指定的虚拟目录。

安装后应通过发布提示符中的 *iisreset* 重新启动计算机的 IIS 服务器。另一个选项是回收调用应用程序的对象池。

B

自定义 MapXtreme

本附录包含 MapXtreme 中可定制的类、接口和其它元素的示例。

在本附录中：

- ◆ 可定制的类 452
- ◆ **Workspace** 管理器扩展 459
- ◆ 应用程序数据文件的位置 463
- ◆ 查找缩写文件 465

可定制的类

以下类设计用于简化子类，以创建更适合开发需要的定制子类。

- **MapInfo.Data.Provider** 命名空间
- **ADO.NET**
- **Engine.CustomProperties**
- **Search**
- **FeatureStyleModifier** 或 **FeatureOverrideStyleModifier**
- **UserDrawLayer**
- **Windows.Controls**
- 工具
- 样式
- **GmlFeatureCollection**
- **WorkSpacePersistence** 和 **WorkSpaceLoader**

MapInfo.Data.Provider 命名空间

MapXtreme 提供“可扩展数据提供方”模型，以便使用 MapXtreme 在本地无法理解的形式访问数据。此模型需要大量的开发工具，而且只有在 MapXtreme 提供的其他数据访问方法不足时才可以使用。此模型的相关说明请参阅附录 D：可扩展数据提供方。

ADO.NET

在使用上述的“可扩展数据提供方”模型尝试执行您自己的数据提供方时，请考虑 **MapInfo.Data.TableInfoAdoNet** 类。使用此类可访问我们未提供专用数据源的不可制图的数据。请参阅《开发人员参考》以了解更多信息。

Engine.CustomProperties

使用 **CustomProperties** 类将定制信息添加到对象。可以由 **CustomProperties** 添加信息的不同种类的对象是：**FeatureCollection**、**FeatureStyleModifer**、**GmlFeatureCollection**、**GroupLayer**、**IFeatureCollection**、**IMapLayer**、**ISession**、**LabelModifier**、**LabelSource**、**Legend**、**LegendFrame**、**Map**、**MapLayer**、**MapTool**、**MultiFeatureCollection**、**Session.PooledSession** 和 **Table**。使用 **Add** 方法添加信息。使用 **Item** 方法检索信息。**CustomProperties** 可以是其中的任意类型。

i 不要直接将 MapXtreme 对象增加到 **CustomProperties** 集合中。如果这样做，会在串行化期间引起错误。相反，应该添加 **Alias**。例如，不将 **Map** 添加到 **CustomProperties** 集合。而是添加 **Map.Alias**。

以下是一个示例，说明了 **CustomProperties** 如何用于添加和检索属性：

```
Public Shared Sub MapInfo_Engine_CustomProperties()
    Dim bag As CustomProperties = New CustomProperties
    bag.Add("One", 1)
    bag.Add("DateNow", DateTime.Now)
    Dim i As Integer = CType(bag("One"), Integer)
    Dim ts As DateTime = CType(bag("DateNow"), DateTime)
End Sub
```

Search

有一些类可以用于定制搜索函数，以实现几乎任何种类的搜索需要。QueryFilter 类用于创建定制“where”子句用于 SQL 查询。QueryDefinition 类用于执行定制的自定义 SQL 查询语句；SearchResultProcessor 类则设置查询结果的后处理。

关于如何使用上述类的示例，请在 MapXtreme 安装的 Samples 目录中参阅 Search 示例应用程序。

（默认安装位置为：

C:\Program Files\MapInfo\MapXtreme\7.x.x\Samples\Desktop\Features\Search）。

QueryFilter

IQueryFilter 接口定义所有查询筛选器必须支持的接口。QueryFilter 用于定义 QueryDefinition 的全部或部分“where”子句。

QueryDefinition

QueryDefinition 类定义搜索执行的查询。QueryDefinition 由筛选器 (IQueryFilter)、Columns 和 OrderBy 组成。如果不指定 Columns，则使用“*”。

I SearchResultProcessor 或 SearchResultProcessor

SearchResultProcessor 实现 I SearchResultProcessor 接口，该接口用于执行搜索结果的后处理以缩小选定行的范围。

FeatureStyleModifier 或 FeatureOverrideStyleModifier

创建定制 FeatureStyleModifiers 以通过 FeatureStyleModifier 或 FeatureOverrideStyleModifier 的派生更改图层外观。上述类用于指定特定图层中对象的特定 FeatureStyle 修改。

以下示例代码创建修饰符，如果人口从 1990 到 2000 年之间在减少，则修饰符将 USA.TAB 中的区域设置为红色。

```
using MapInfo.Mapping;
using MapInfo.Styles;
using System.Drawing;

internal class UsaPopulationDecreasedModifier :
    MapInfo.Mapping.FeatureStyleModifier
```

```
{  
    public UsaPopulationDecreasedModifier() : base(null, null)  
    {  
        // use 2 columns for expressions  
        Expressions = new string[] { I18N.Wrap(  
            "Pop_1990", I18N.WrapType.Column),  
            I18N.Wrap("Pop_2000", I18N.WrapType.Column) };  
    }  
  
    // called during draw events  
    protected override System.Boolean  
        Modify(MapInfo.Styles.FeatureStyleStack styles,  
        object[] values)  
    {  
        // compares the values from each column  
        if (double.Parse(values[0].ToString()) >  
            double.Parse(values[1].ToString()))  
        {  
            // if population decreased, color the region red  
            CompositeStyle style = styles.Current;  
            if (style.AreaStyle.Interior is SimpleInterior)  
            {  
                ((SimpleInterior)style.AreaStyle.Interior).ForeColor =  
                    Color.Red;  
                // modifies the region  
                return true;  
            }  
        }  
        // does not modify the region  
        return false;  
    }  
}
```

UserDrawLayer

UserDrawLayer 类用于绘制地图中的定制图层。可以在该图层中填充任何需要的对象，包括定制徽标、点的透明覆盖等。要使用该类，从该类派生新类并重载绘制方法。以下示例代码说明了上述方法：

C# 示例：

```
using System.Runtime.Serialization;  
using System;  
using System.Drawing;  
using MapInfo.Mapping;  
  
[Serializable]  
class MyUserDrawLayer : UserDrawLayer {  
    // Call the base class constructor with name and alias.  
    public MyUserDrawLayer(string Name, string Alias) : base(Name, Alias)  
{}}
```

```
// Override the abstract Draw method to draw a rectangle.  
public override void Draw(System.Drawing.Rectangle ClientRect,  
System.Drawing.Rectangle  
    UpdateRect, Graphics graphics) {  
    // Create a pen.  
    Pen blackPen = new Pen(Color.Black, 3);  
    // Create location and size of rectangle.  
    float x = 10.0F;  
    float y = 10.0F;  
    float width = 200.0F;  
    float height = 200.0F;  
    // Draw rectangle to screen.  
    graphics.DrawRectangle(blackPen, x, y, width, height);  
}  
}  
}
```

VB UserDrawLayer 示例应用程序与 ..\MapInfo\MapXtreme\7.x.x\Samples\Desktop\Features 文件夹中的 MapXtreme 一起提供。

Windows.Controls

Windows.Controls 命名空间中的很多类可以创建子类以实现定制行为。以下是使用该途径定制 LayerControl 的两个示例。

MapInfo.Windows.Controls.PropertiesUserControl

如果需要将定制的选项卡添加到 LayerControl，则创建 PropertiesUserControl 的子类。然后通过调用 GetLayerTypeControls() 和 SetLayerTypeControls() 方法，将定制类添加到其它选项卡类的集合中。该方法工作的示例，请参阅包含在 MapXtreme 安装路径 Samples\Features 目录下的示例应用程序。关于 PropertiesUserControl 的子类 MapBackgroundControl 类，请参阅 LayerControl 示例应用程序中的 MapBackgroundControl.cs 文件。

MapInfo.Windows.Controls.LayerNodeHelper 和它的所有子类

LayerNodeHelper 类控制 LayerControl 的图层树中节点的外观和行为。例如，这些帮助程序类指示用户可以删除的图层。如果 ILayerNodeHelper 接口的属性不能对图层节点的外观或行为提供足够的控制（例如，如果需要应用按表逻辑，该逻辑允许用户删除部分地图图层，但不能删除其它），可以将任意帮助类子类化，重载适当的方法，执行定制重载方法中的按表逻辑。然后通过调用 LayerControl.SetLayerTypeHelper 方法应用新的帮助程序类。

默认情况下，用户删除 GroupLayer 时，将出现确认对话框，除非在 GroupLayer 完全为空的情况下—在这种情况下，空的 GroupLayer 将被简单删除而不出现确认提示（LabelLayers 的工作方式类似）。要强制 Layer Control 始终显示确认提示，甚至在 GroupLayer 为空时也显示，可以创建从 GroupLayerNodeHelper 派生并重载 IsRemovalConfirmed 方法的类。以下代码示例说明如何实现该方法：

```
using System;
```

```
using MapInfo.Windows.Controls;

namespace MapInfo.Samples.LayerControl
{
    public class CustomLayerNodeHelper : GroupLayerNodeHelper
    {
        public CustomLayerNodeHelper()
        {
        }

        public override bool IsRemovalConfirmed(object obj)
        {
            return true;
        }
    }
}
```

然后，要在应用程序中使用该新帮助程序类，使用如下的语法调用 LayerControl 的 SetLayerTypeHelper 方法：

```
layerControlDlg.LayerControl.SetLayerTypeHelper(
    typeof(GroupLayer), new CustomLayerNodeHelper() );
```

LayerNodeHelper 的子类是：

- MapNodeHelper
- MapLayerNodeHelper
- GroupLayerNodeHelper
- LabelLayerNodeHelper
- LabelSourceNodeHelper
- LabelModifierNodeHelper
- FeatureStyleModifierNodeHelper
- RangedThemeNodeHelper
- DotDensityThemeNodeHelper
- IndividualValueThemeNodeHelper
- PieThemeNodeHelper
- BarThemeNodeHelper
- GradSymbolThemeNodeHelper

工具

可通过两种方式定制工具：通过现有工具的子类化或通过将代码添加到定制基本工具。有关桌面工具的详细信息，请参阅[第 7 章：桌面应用程序、控件、对话框和工具](#)。有关 web 工具的详细信息，请参阅[第 5 章：Web 应用程序、控件和工具](#)。

样式

以下类可以用于将定制信息添加到应用于对象的样式集。

BitmapPointStyleRepository

BitmapPointStyleRepository 类用于重复当前所有位图点样式，并允许用户从指定的目录添加新位图图像。使用 **Refresh()** 方法将任意新位图图像放置到保存所有图像的库中。以下代码示例演示了“C:\MyData\MyBitmapImages”目录下用于 **BitmapPointStyles** 的图像集合的重新加载。

```
using MapInfo.Styles;

StyleRepository styleRepository = Session.Current.StyleRepository;
BitmapPointStyleRepository bmpPointStyleRepository =
    styleRepository.BitmapPointStyleRepository;
bmpPointStyleRepository.Reload("C:\MyData\MyBitmapImages");
```

InteriorStyleRepository

InteriorStyleRepository 类用于重复当前所有内部样式，并允许用户从指定的目录添加新内部样式的位图图像。要添加到用于填充区域的内部图案的当前集上，请使用 **AppendBitmapPattern()** 方法。以下代码示例演示了将新位图图像添加到 **InteriorStyleRepository**。

```
using MapInfo.Styles;

StyleRepository styleRepository = Session.Current.StyleRepository;
InteriorStyleRepository interiorStyleRepository =
    styleRepository.InteriorStyleRepository;
// AppendBitmapPattern returns the zero-based index of the pattern in
// the repository. The index returned may be used to
// retrieve a SimpleInterior style using the repository indexer.
int index = interiorStyleRepository.AppendBitmapPattern(
    "C:\MyImages\trees.bmp"
);
if (index == 0)
{
    SimpleInterior style =
        (SimpleInterior)interiorStyleRepository[index];
}
```

LineStyleRepository

LineStyleRepository 类用于重复当前所有直线样式并允许用户添加新的直线样式。（可以使用 **Reload()** 方法将新文件放置到库中。“C:\MyData”目录下 **PEN** 文件中的示例代码演示了用于 **SimpleLineStyles** 的直线样式图案集的重新加载。

```
using MapInfo.Styles;

StyleRepository styleRepository = Session.Current.StyleRepository;
LineStyleRepository lineStyleRepository =
    styleRepository.LineStyleRepository;
lineStyleRepository.Reload("C:\MyData\MyLineStyles.PEN");
```

VectorPointStyleRepository

`VectorPointStyleRepository` 类用于重复当前所有矢量符号，并允许用户从指定的文件添加新矢量符号。要创建符号集的定制版本，需要使用 `Symbol Maker MapBasic` 应用程序。矢量符号集更改后，使用 `Reload()` 方法将符号集读取到库中。“C:\MyData” 目录下 `FNT` 文件中的示例代码演示了 `SimpleVectorPointStyles` 集的重新加载。

```
using MapInfo.Styles;

StyleRepository styleRepository = Session.Current.StyleRepository;
VectorPointStyleRepository vecPointStyleRepository =
    styleRepository.VectorPointStyleRepository;
vecPointStyleRepository.Reload("C:\MyData\MyVectorSymbols.FNT");
```

GmlFeatureCollection

`GmlFeatureCollection` 类用于从 `XML` 文件导入图元。可以通过创建 `FeatureCollection` 对象并构造 `FeatureObjects` 加入创建的对象的方法，将 `Features` 添加到地图。然后可以将 `FeatureCollection` 插入到地图中。

WorkSpacePersistence 和 WorkSpaceLoader

`WorkSpaceLoader` 可以被子类化，以便使用的永久文件可以保存从应用程序保存的附加信息。开发人员可以使用应用程序写入任何需要保存到永久 `XML` 文件中的信息，做法是通过在 `UserData` 节点下添加信息。`WorkSpacePersistence` 类写入持续文件且该节点中有某些内容时，该内容将自动写入文件。`WorkSpaceLoader` 类或该类的某些派生类读取的数据包含 `UserData` 节点的子节点。如下代码是它的示例。

```
using MapInfo.Persistence;

class myWorkSpaceLoader : WorkSpaceLoader
{
    public myWorkSpaceLoader(string fileName) : base(fileName)
    {
        // empty
    }
    override public void Load()
    {
        // called parent Load() method.
        base.Load();

        XmlNode userData = this.UserDataXmlNode;
        foreach (XmlNode childNode in userData.ChildNodes)
        {
            string text = childNode.Name;
            text = childNode.InnerText;
            // user can do their own load stuff here
        }
    }
}
```

Workspace 管理器扩展

MapXtreme 的工作空间创建工具— Workspace 管理器可进行扩展，以添加新的菜单命令、工具和选项卡窗口，这样可以简化创建工作空间的过程。

有关 Workspace 管理器功能和用户界面的详情，请参阅第 24 章： **Workspace 管理器**。

Workspace 管理器扩展为 .NET 程序集，需进行写操作才能扩展 Workspace 管理器的功能。一般来说，您很可能需要创建 MapXtreme API，以通过 Workspace 管理器不显示的对象模型添加功能。

例如，可添加多个菜单项、工具、选项卡窗口或进行响应以更改事件。“图层控制”也可以扩展，因此可以使用同样的方式将新功能添加到其菜单。

只要扩展可以使用，就只需在 Workspace 管理器中已更新的菜单条上单击“扩展”菜单以加载扩展。Workspace 管理器启动时，也可以自动加载以供使用。

我们提供了一些您可能需要创建的 Workspace 管理器扩展的示例，它们包括具有排序功能的表浏览器、自定义主题图模板或创建/编辑对象的新工具。

创建 Workspace 扩展

MapXtreme 提供 `MapInfo.WorkspaceManager.Extension.dll`，它是一个程序集，可定义界面以帮助您将自己的功能添加到 Workspace 管理器。

```
public interface IWorkspaceManagerExtension
{
    string Name
    {
        get;
    }
    string Version
    {
        get;
    }
    bool HasPropertiesDialog
    {
        get;
    }
    void ShowPropertiesDialog(IWin32Window owner);

    IWorkspaceManagerNotifications Load(IWorkspaceManager callback);

    void Unload();
}
```

要创建扩展，请创建一个至少具有一个类的程序集，以执行 `IWorkspaceManagerExtension` 界面。此类应有一个不含参数的构造函数。

加载方法

IWorkspaceManagerExtension 上的主要方法为：

```
IWorkspaceManagerNotifications Load(IWorkspaceManager callback);
```

此方法在扩展类已构造后才能调用。传递它的对象实例可执行 IWorkspaceManager，使类能够调用返回 Workspace Manager 以访问用户界面的元素（例如菜单、toolstrips、abcontrol）并执行命令（例如加载和保存工作空间）。

```
public interface IWorkspaceManager
{
    bool NewWorkspace();
    bool LoadWorkspace(string filename);
    void SaveWorkspace(string filename);
    void MarkWorkspaceChanged();
    bool HasWorkspaceChanged();

    string LoadedWorkspaceFileName{get;}

    int AddMapTab(MapInfo.Mapping.Map map);
    int FindMapTab(MapInfo.Mapping.Map map);
    int AddLegendTab(MapInfo.Mapping.Legends.Legend legend);
    int FindLegendTab(MapInfo.Mapping.Legends.Legend legend);
    void RemoveTab(int tab);

    Microsoft.Win32.RegistryKey GetRegistryKey();

    MapInfo.Windows.Controls.MapControl MapControl{get;}

    MapInfo.Windows.Controls.MapControl GetMapControlFromTab(int tab);
    MapInfo.Windows.Controls.LegendControl GetLegendControlFromTab(int tab);
    MapInfo.Windows.Controls.LayerControl LayerControl{get;}

    System.Windows.Forms.Form MainForm{get;}

    System.Windows.Forms.MenuStrip Menu{get;}
    System.Windows.Forms.ToolStrip FileToolStrip{get;}
    System.Windows.Forms.ToolStrip MapToolStrip{get;}
    System.Windows.Forms.ToolStrip ToolsToolStrip{get;}
    System.Windows.Forms.StatusStrip StatusStrip{get;}
    System.Windows.Forms.TabControl MainTabControl{get;}

    ContextMenuStrip MapContextMenu{get;}
    ContextMenuStrip LegendContextMenu{get;}
    ContextMenuStrip GenericTabContextMenu{get;}
}
```

事件处理

如果扩展需要接收来自于 **Workspace** 管理器的事件通知，请添加可执行 **IWorkspaceManagerNotifications** 界面并能从 **Load()** 方法返回的类。创建、加载或保存工作空间时，**Workspace Manager** 将调用此类。

```
public interface IWorkspaceManagerNotifications
{
    void OnNewWorkspace();
    void OnWorkspaceLoaded(WorkSpaceLoader workspaceLoader, string
filename);
    void OnWorkspaceSaving(WorkSpacePersistence workspacePersistence,
string filename);
    void OnWorkspaceSaved(string filename);
    void OnLayerControlDialog(MapInfo.Windows.Dialogs.LayerControlDlg
dlg);
}
```

加载扩展

加载 **Workspace** 管理器扩展时，可通过“扩展”>“加载扩展”菜单命令。“打开”菜单出现，可使用它导航到程序集的位置。

使用提供的“扩展管理器”对话框，以显示已加载和未加载的扩展。可通过“扩展”>“管理扩展”命令访问。它可以调用一个属性对话框（如果有任何一个可用）。

从内存中程序集加载扩展

扩展也可以自动加载。启动时，**Workspace** 管理器会查找扩展，将它们加载到两个位置。

第一，它扫描所有已加载到当前 **AppDomain** 中的程序集，以查看是否有任何类型可以执行 **IWorkspaceManager**。这是一种高级方案；只有在 **SessionEventHandler**（请参阅 **ISessionEventHandlers**）也正好包含 **Workspace** 管理器扩展时才可能使用它。

第二，**Workspace** 管理器进入名为 **Extensions** 的文件夹，在 **WorkspaceManager.exe** 的示例位置查找。程序集的名称必须为 **<my_extension>.WorkspaceManagerExtension.dll**。此位置中的所有扩展程序集将加载到和 **Workspace** 管理器一样的 **AppDomain** 中。

加载扩展的命令行参数

也可通过命令行控制，使 **Workspace** 管理器查找要加载的扩展：

/LoadExtensions=[All|Folder|None]

其中：

All 表示在已加载的程序集查找，然后在 **Extensions** 文件夹中查找

Folder 表示仅在 **Extensions** 文件夹中查找，跳过已加载的程序集

None 表示不自动加载任何扩展

卸载扩展

用户可卸载扩展，或者在退出 **Workspace** 管理器时也可卸载扩展。经过设计，扩展可删除任何添加的菜单、工具条或选项卡窗口等等，以尽量释放资源。**Workspace** 管理器关闭时，将从 AppDomain 卸载扩展程序集。

示例扩展

以下是一个 **Workspace** 管理器的示例扩展，它在启动时加载以前使用的工作空间。

```
using MapInfo.WorkspaceManager.Extension;
using System.Windows.Forms;
using System;

public class LoadLastWorkspaceExtension : IWorkspaceManagerExtension
{
    private string _name;
    private IWorkspaceManager _callback;

    private LoadLastWorkspaceExtension()
    {
        _name = "Load Last Workspace Extension";
    }

    public string Name
    {
        get { return _name; }
    }

    public string Version
    {
        get { return "0.57"; }
    }

    // this get called when extension is first loaded
    // use it to hook up and UI like menu items, toolbars, etc
    public IWorkspaceManagerNotifications Load(IWorkspaceManager
callback)
    {
        _callback = callback;

        Microsoft.Win32.RegistryKey key = _callback.GetRegistryKey();
        string s = (string)key.GetValue("RecentFiles");
        if (s != null && s.Length > 0)
        {
            try
            {
                string[] files = s.Split('|');
                callback.LoadWorkspace(files[0]);
            }
            catch { }
        }
    }
}
```

```
        }

        key.Close();

        return null; // returning because we don't need the
notifications
    }

    public void Unload()
{
    // nothing to clean up
}
public bool HasPropertiesDialog
{
    get { return false; }
}

public void ShowPropertiesDialog(IWin32Window owner)
{
    throw new NotImplementedException();
}

}
```

应用程序数据文件的位置

任何 MapXtreme 应用程序都使用以下文件中存储的数据。

文件类型	文件名
缩写文件	MAPINFO.ABB
画笔文件	MAPINFO.PEN
投影文件	MapInfoCoordinateSystemSet.xml
矢量符号文件	MapInfo.fnt
定制符号目录	CustSymb
Nadcon 文件	*.las、*.los
jgd2000 文件	jgd2000.*

默认情况下，MapXtreme 应用程序在以下目录寻找数据文件：

- C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x — 这是 MapXtreme 安装程序存放这些文件的目录。

- 用户的应用程序存放的目录。对于 Windows 应用程序，这是存放 .exe 文件的目录。对于 web 应用程序，这是存放 Web.config 文件的目录。
- MapInfo.CoreEngine 程序集文件夹 — 该目录是应用程序寻找数据文件的最后位置。

对以上位置的目录列表可以通过 `ISessions.AppDataPaths` 属性获得，该属性将列表以字符串数组的形式返回。当不能通过编程方式添加到该列表时，可以通过在应用程序的配置文件中定义定制部分的方法添加到列表。以下示例代码说明了该方法。注意，附加路径定义的顺序确定它们搜索的顺序，并且所有定制路径在以上列出的默认位置之前搜索。

下面的示例显示了 `<SpecialPath>` 标记：

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
        <section name="MapInfo.CoreEngine"
            type="MapInfo.Engine.ConfigSectionHandler,
            MapInfo.CoreEngine, Version=6.8.0.536, Culture=neutral,
            PublicKeyToken=93e298a0f6b95eb1" />
    </configSections>
    <MapInfo.CoreEngine>
        <ApplicationDataPaths>
            <SpecialPath>
                <LocalApplicationData>YourCorporation\Your
Application<\LocalApplicationData>
                <\SpecialPath>
            </ApplicationDataPaths>
        </MapInfo.CoreEngine>
    </configuration>
```

在该示例中，MapXtreme 将搜索类似如下文件夹中的自定义符号：C:\Documents and Settings\username\Application Data\Your Corporation\Your Application\CustSymb。请注意 `<SpecialPath>` 元素不应包括 CustSymb 文件夹名称。如果该元素包含了文件夹名称，则路径将会被解释为 ...\\Your Application\\CustMapXtreme 2004\\CustSymb\\CustSymb。重复的 CustSymb 是不正确的。

如果应用程序数据存储在相对于 .NET Framework 特殊系统文件夹的位置，则使用 `<SpecialPath>` 语法。例如，如果应用程序数据存储在“我的文档”目录下名为 `MyAppData` 的目录中，则配置文件中的条目可能为：

```
<Personal>MyAppData</Personal>
```

其中 "Personal" 是表示“我的文档”目录的 .NET Framework 枚举 `Environment.SpecialFolder` 的值。

以上配置元素中的每个元素按以下方式定义：

`<configSections>`

这是声明定制部分句柄的位置。这种情况下，程序集 `MapInfo.CoreEngine` 中定义的 `ConfigSectionHandler` 类负责分析配置文件的 `MapInfo.CoreEngine` 部分。

`<MapInfo.CoreEngine>`

该部分包含 `MapInfo.CoreEngine` 程序集的设置。特定地，应用程序数据搜索路径。

<ApplicationDataPaths>

该部分定义 MapXtreme 在搜索应用程序数据文件时使用的附加路径。

<Path>

如果应用程序数据的完整路径已知则要使用的元素。

<SpecialPath>

如果应用程序数据存储在众所周知的系统文件夹的相对位置，则使用该语法。例如，如果应用程序数据存储在“我的文档”目录下名为 **MyAppData** 的文件夹中，则配置文件的条目名称将采取 `<Personal>MyAppData</Personal>` 的形式，此处“Personal”是 `Environment.SpecialFolder` 枚举的值，表示“我的文档”目录。任何 `Environment.SpecialFolder` 中的枚举值都可以用于在配置文件中定义应用程序数据路径。

查找缩写文件

可以定制 `Abbreviation` 文件以匹配用户数据，使查找操作更有效率。有关详细信息，请参阅[微调查找过程](#)。

C

理解 MapInfo 工作空间

MapXtreme 提供了工作空间格式，可以移植和互操作并使用 MapInfo codespace 定义。本附录介绍工作空间的定义、功能和使用。关于 MapInfo codespace 定义的详细信息，请参阅[附录 G: 定义 MapInfo Codespace](#)。

在本附录中：

- ◆ 什么是 **MapInfo** 工作空间? 467
- ◆ 通过 **.GST** 以编程方式创建 **.MWS** 工作空间 471
- ◆ 工作空间的结构 467

什么是 MapInfo 工作空间？

使用 **MapXtreme**，用户可以将创建的地图作为基于 XML 的工作空间保存（使用 .MWS 扩展名），工作空间可以移植和互操作。然后可以和其他任何使用 **MapInfo** 产品的用户共享这些地图，无论他们处于何种工作环境。

如果用户熟悉 **MapXtreme Java**、**MapX** 或 **MapInfo Professional**，则会了解到目前用户使用不同的文件格式保存地图：

- **MapXtreme Java** 地图保存为 **Map Definition Files (.MDF 文件)**
- **MapX** 地图保存为 **geosets (.GST 文件)**
- **MapInfo Professional** 地图保存为工作空间 (.WOR 和 .MWS 文件)

上述文件已成为独立的 **MapInfo** 产品用户共享地图的方式。**MapInfo Professional** 工作空间也可以包含浏览器的设置并布局窗口、图形、图例甚至打印机的设置。

MapInfo Workspace 格式支持创建命名资源以便于访问移植到连接、地图定义、地图图层、数据源定义以及样式。主工作空间架构名为 **MXP_Workspace_1_5.xsd**。其位于产品介质中。其它支持的架构包括用于地图定义的 **MXP_MapDefinition_1_5.xsd** 以及用于命名图层、连接、数据源定义和样式。

本公司的 XML 文档使用文本标识符，并且尽可能的使用了 XML 标准化组织定义的标识符。不过，XML 是可扩展的，任何文档作者都可以添加标识符以阐明文档使用的数据的含义。为确保这些标识符按照 **Pitney Bowes Software** 定义的规范准确标注，本公司在名为 **MapInfo Codespace** 的文档中对它们作了详细说明。要查看 **MapInfo Codespace** 标识符，请参阅 [附录 G：定义 MapInfo Codespace](#)。

关于创建工作空间和使用 **Workspace** 管理器的详细信息，请参阅 [第 24 章：Workspace 管理器](#)。



不能将在 **MapInfo Professional** 中创建的 .WOR 工作空间用于 **MapXtreme**。**MapXtreme** 可读取 **MapX geosets**。

工作空间的结构

要理解工作空间的结构，可以通过一个具体的工作空间及其相关的 XML 代码。下面将以 **MapXtreme** 安装路径的 **Sample** 文件夹下的 **World.MWS** 工作空间作为示例。下文描述了工作空间中五种数据类型中的四种。

1. **Header** 包含了基本文件信息，包括版本类型和创建日期。
2. **Connection** 部分定义了工作空间中包含的数据库、文件和 WMS 连接。这里是命名连接信息存储的位置。
3. **DataSourceDefinition** 部分列出了数据的定义及其所在的位置。如果有命名数据源定义，则会在此处包含这些定义。

4. **MapDefinition** 部分包含了和工作空间关联的图层、主题和标注图元的定义，例如标注的详细信息、缩放范围、使用的颜色等。此处将包含地图、图层、主题、样式的命名定义。

架构中的附加部分适用于用户定义的数据。这使应用程序可以永久化自身的数据。用户数据是“通配符”元素，因为不会针对构架对其进行验证，所以可以输入任意需要的内容。

有关 MapXtreme 中命名资源支持的信息，请参阅[打开和保存包含命名资源的工作空间](#)。

Header 部分

文件以必需的 XML 文件头信息开始，该文件头信息描述了文件的 XML 版本，编码描述和指示文件是否独立的字段。文件中的最高级元素是 **Workspace Element**，包含文件版本、日期和 XML 命名空间定义的属性。

 MapXtreme 工作空间永久性支持的编码方式是 UTF-8。

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<WorkSpace name="" version="MXP_WorkSpace_1_5"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.mapinfo.com/mxp">
```

Connection 部分

ConnectionString 部分定义了使用工作空间必需的数据库连接和文件连接。目前支持的数据库连接包括 MS Windows ODBC 连接、Oracle (OCI) 连接和 JDBC 驱动程序连接。文件连接标识文件的路径。以下是这些连接的一些示例。

```
<ConnectionString>
  <DBConnection dbType="sqlserver">
    <ConnectionName>JCD_SS2005_MIGS_BOUNDARY2</ConnectionName>
    <ODBCConnectionString>driver={SQL Native Client};server=JODEMPSE-W3\MIGS;database=MIUS06_BOUNDARY2;uid=MIGS;</ODBCConnectionString>
  </DBConnection>
</ConnectionString>

<ConnectionString>
  <FileConnection dbType="file">
    <ConnectionName>MapStorage1_US</ConnectionName>
    <FilePath>c:\data\maps</FilePath>
  </FileConnection>
</ConnectionString>
```

DataSource Definition 部分

DataSource Definition 部分定义了在每个连接位置检索的数据文件（例如，TAB 文件）和数据库表。它也可以包括可选的元数据以描述要检索的数据。此示例显示了命名数据源定义、数据库数据源定义以及 TAB 数据源定义。

```
<DataSourceDefinitionSet>
    <NamedDataSourceDefinitionRef resourceId="MyDataSource"/>
    <DBDataSourceDefinition id="id1" readOnly="false">
        <DataSourceName>alias_Line_ontario_oracle_9i_rel
        </DataSourceName>
        <ConnectionMember>
            <ConnectionName>ontario_oracle_9i_release_1</ConnectionName>
        </ConnectionMember>
        <DBQuery>
            <Query>select * from us_hiway_extra</Query>
        </DBQuery>
        <DBDataSourceMetadata>
            <FeatureGeometryAttribute srsName="mapinfo:coordsys 1,62">
                OBJECT
            </FeatureGeometryAttribute>
            KeyAttributes/
        </DBDataSourceMetadata>
    </DBDataSourceDefinition>
    <TABFileDataSourceDefinition id="id2" readOnly="false">
        <DataSourceName>alias_Line_us_hiway_extra</DataSourceName>
        <FileName>FeatureLayerBuilder\us_hiway_extra.tab</FileName>
    </TABFileDataSourceDefinition>
</DataSourceDefinitionSet> ...
```

Map Definition 部分

Map Definition 部分定义一张或多张地图和组成它们的图层。每张地图都具有显示条件，包括图像大小、缩放和中心设置、渲染地图的坐标系。此示例显示了命名地图定义引用和地图定义。在 **LayerList** 下方有两个命名图层、三个定义的图层以及标注图层的命名样式引用。

```
<MapDefinitionSet>
    <NamedMapDefinitionRef resourceId="MyWorldMap"/>
    <MapDefinition id="id7" name="Map1" alias="Map1" uniqueId="4adeb0e9-7c77-4957-a3fb-a1a0677756ef">
        <DisplayConditions>
            <MapSize uom="mapinfo:imagesize pixel">
                <ImageWidth>600</ImageWidth>
                <ImageHeight>400</ImageHeight>
            </MapSize>
            <ZoomAndCenter>
                <MapZoom uom="mapinfo:length mi">3000</MapZoom>
                <gml:Point srsName="EPSG:4269">
                    <gml:coordinates>-79.771366,38.003251000000006
                    </gml:coordinates>
                </gml:Point>
            </ZoomAndCenter>
        </DisplayConditions>
    </MapDefinition>
</MapDefinitionSet>
```

```
</ZoomAndCenter>
<DisplayCoordSys>
    <SRSName>EPSG:4269</SRSName>
</DisplayCoordSys>
</DisplayConditions>
<LayerList>
    <NamedLayerRef resourceID="NamedLayer_World 25 Major Cities"/>
    <NamedLayerRef resourceID="NamedLayer_World Capitals"/>
    <FeatureLayer id="id8" name="World25Cities"
        alias="World25Cities"
        volatile="unknown">
        <DataSourceRef ref="id4" />
    </FeatureLayer>
    <FeatureLayer id="id9" name="WorldCapitals"
        alias="WorldCapitals"
        volatile="unknown">
        <DataSourceRef ref="id2" />
    </FeatureLayer>
    <FeatureLayer id="id10" name="Regions"
        alias="Regions"
        volatile="unknown">
        <DataSourceRef ref="id6" />
    </FeatureLayer>
<LabelSource maxLabels="2147483647" name="World 25 Major Cities">
    <Visibility visible="true">
        <VisibleRange enabled="true">
            <ZoomRange uom="mapinfo:length mi" minInclusive="true"
maxInclusive="true">0 8000</ZoomRange>
        </VisibleRange>
    </Visibility>
    <DataSourceRef ref="id3" />
    <BaseLabelProperties>
        <LabelProperties>
            <LabelVisibility visible="true">
                <VisibleRange enabled="false">
                    <ZoomRange uom="mapinfo:length mi"
minInclusive="true" maxInclusive="true">0 8000</ZoomRange>
                </VisibleRange>
            </LabelVisibility>
            <LabelText justification="right">
                <BaseLabelStyle>
                    <NamedStyleRef resourceID="Layer_id7" id="Layer_id7"/>
                    </BaseLabelStyle>
                <StringTokenList>
                    <StringToken>
                        <StringValueExpression>
                            <AttributeName>Place_Name</AttributeName>
                        </StringValueExpression>
                    </StringToken>
                </StringTokenList>
                <LabelCharacterLimit
spacing="1">2147483647</LabelCharacterLimit>
```

```
</LabelText>
<LabelCallout visible="true" end="none">
    <LineStyle stroke="black" width="1" width-
unit="mapinfo:imagesize pixel">
        <Pen>mapinfo:pen 2</Pen>
    </LineStyle>
</LabelCallout>
<LabelReferencePosition recalculate="outOfView">
<LineLabelPositionInterval>0</LineLabelPositionInterval>
</LabelReferencePosition>
<LabelLayout>
    <Alignment>
        <HorizontalAlignment>right</HorizontalAlignment>
        <VerticalAlignment>center</VerticalAlignment>
    </Alignment>
    <Offset uom="mapinfo:imagesize pixel">
        <XOffset>-2</XOffset>
        <YOffset>0</YOffset>
    </Offset>
<LabelRelativeOrientation>parallel</LabelRelativeOrientation>
    </LabelLayout>
    <LabelBaseSize useScale="false">
        <MapScale>0</MapScale>
    </LabelBaseSize>
    </LabelProperties>
</BaseLabelProperties>
<LabelThemeList />
</LabelSource>
</LayerList>
</MapDefinition>
</MapDefinitionSet>
```

该图层列表显示了为该地图设置的标注属性。这些属性包括标注所在的图层、标注的文件源、标注的可视性规则（包括标注可视的范围，如果适用）、标注的字体和大小属性、标注字符限制和标注线（如果适用）、标注位置设置和对齐方式、标注的原点（关于标注的点）。

通过 .GST 以编程方式创建 .MWS 工作空间

在 MapX 中，曾经使用 `Map.SaveMapAsGeoset` 方法从现有的地图创建 geoset。用户也可以通过编程方式保存他们的工作空间，以后在 `Workspace` 管理器中通过使用如下所示的代码片断查看这些工作空间。

① 用户需要打开一张包含一个或多个图层的地图，以将其作为工作空间保存。

```
// Reads a MapX geoset, writes a MapXtreme workspace.
using MapInfo.Persistence;
using MapInfo.Mapping;
。
```

```
。
。
MapLoader MapLoader = MapLoader.CreateFromFile("my.gst");
MapExport MapExport = new MapExport();
MapExport.Map.Load(MapLoader);
WorkSpacePersistence wsp = new WorkSpacePersistence();
wsp.Save("c:\\temp\\newwork.mws");
```

可扩展数据提供方

此附录介绍 **MapXtreme** 的可扩展数据提供方体系结构，并介绍如何构建为 **MapXtreme** 在其他情况下不提供访问方法的空间数据构造这样的体系结构。

在本附录中：

◆ 简介	474
◆ 可扩展数据提供方概述	474
◆ 入门	476
◆ 所需组件	477
◆ 可选构建块：基类、帮助程序和实用程序	479
◆ 示例：COTW (Center of the World) 数据提供方	480
◆ 可选接口	481
◆ 构建和测试数据提供方	482
◆ Spatialite 示例数据提供方	484
◆ 高级主题 / 重点考虑事项	485

简介

MapXtreme 提供可扩展数据提供方体系结构模型，可以实现此模型来访问 **MapXtreme** 不支持的数据格式。此模型由一系列必选及可选接口、抽象基类构建块和实用程序组成。

扩展 **MapXtreme** 的数据提供方模型的工作很困难，需要投入大量的开发和测试资源。大部分 **MapXtreme** 用户发现，用于空间数据格式的现有 **MapXtreme** 数据提供方或用于非空间数据的 Microsoft ADO.NET 完全不足以满足他们的需求。有关 **MapXtreme** 的数据访问选项的完整讨论，请参阅第 8 章：使用数据。

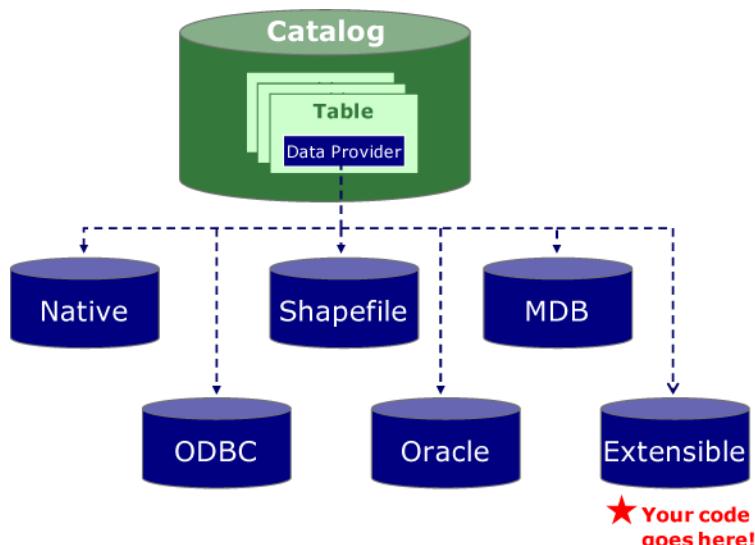
如果您有通过 **MapXtreme** 数据模型无法满足的数据访问要求，请考虑使用这里提供的接口和指南对其进行扩展。

MapXtreme 的可扩展数据提供方接口组织在 **MapXtreme** 对象模型的 *MapInfo.Data.Provider* 命名空间下。为这些接口提供支持的是 *MapInfo.Data* 和 *MapInfo.Data.Common* 命名空间中的类。

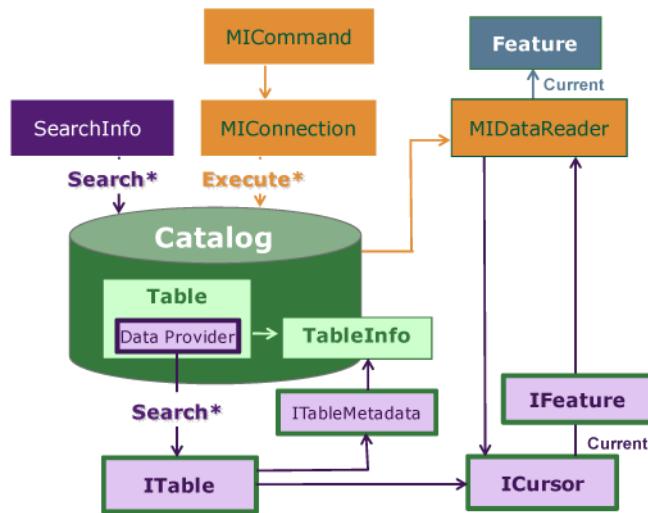
MapXtreme 可扩展数据提供方目前支持打开表、读取表内容及关联的元数据集、使用多个方法搜索表内容以及通过插入、更新和删除操作修改表内容。

可扩展数据提供方概述

MapXtreme 中的目录使用内部数据提供方体系结构构建。可扩展数据提供方模型进行了改编，将此内部体系结构通过一组 .NET 接口公开，如下图中所示。



要了解数据提供方如何工作以及如何创建数据提供方，有一些核心概念要了解。下图显示了一些关键接口和 **MapXtreme** 数据访问引擎的现有组件间的关系。可扩展数据提供方的组件在粗线框中以淡紫色显示。



数据提供方

数据提供方是应用程序和数据源之间的桥梁，提供用于访问数据的机制，以便在应用程序中使用这些数据。

MapXtreme 可扩展数据提供方是一组接口，通过这些接口，您可以访问来自任意数据源、采取任意数据格式的数据。**MapXtreme** 数据提供方提供数据和 **MapXtreme** 功能（如显示、查询、编辑和分析）之间的连接，而可扩展数据提供方对其进行扩展。

“数据提供方”这一术语用于指代现有接口的具体实现。例如，**MapXtreme** 包括用于访问 SpatiaLite（基于 SQLite）数据库的数据提供方实现。可扩展接口的数据提供方集合中有一个名为 **IDataProvider** 接口，此接口是数据提供方的基础。

数据源

数据源是公开数据和数据访问功能（如数据的描述、查询、操作和相关操作）的数据库管理系统、Web 服务或其他引擎或软件 API。

例如，将地图数据作为图元集合返回的 WFS 服务器就是一个数据源。WFS 服务器是可从中将图元类型作为表格公开的数据源。服务的 URL 以及可能存在的其他身份验证属性一起定义数据提供方将如何访问数据源。此信息称为数据源定义。

大部分数据提供方实现都将包含一个数据源，不过此概念并非必不可少。采用基于文件的格式的数据提供方可能仅包含表模型。例如，**MapInfo** 表和 **ESRI Shapefiles** 都是不需要连接到数据源的数据提供方。

表

表是具有相同架构（或列集）的一组图元。表不一定必须具有几何体属性；不过，如果构建数据提供方，那么可能就会在一些表上有几何体属性。**MapXtreme** 将几何体数据作为表上的列处理，就像包含字符串或数字之类较为简单的数据类型的列一样。

表公开用于描述所包含的数据的元数据，并提供对数据查询和编辑的访问权限。表中的每个图元都必须由数据提供方使用键进行唯一的标识。**MapXtreme** 使用键进行选择、结果集和编辑操作。

光标

光标是用于枚举一组图元的对象。**MapXtreme** 将通过对表调用搜索方法之一从数据提供方请求数据。表将返回光标，**MapXtreme** 可通过光标枚举满足指定标准的图元。通过光标返回的图元预期为瞬态的。这意味着从当前图元获取的数据假定仅在光标打开并定位在这个当前记录时有效。一旦光标位置更改（移动到下一图元或关闭光标），则该图元不再可访问。稳定且可扩展的数据提供方实现应该能够使用光标模型的瞬态特征来重用内存，特别是用于返回 **FeatureGeometry** 对象时。

图元访问器是一种特殊的光标，用于使用键访问图元。任何时候，只要 **MapXtreme** 需要使用键访问一个或多个图元，都将请求一个图元访问器，然后从访问器请求图元。图元访问器也具有与光标相同的瞬态性。

入门

既然您已决定开始着手对 **MapXtreme** 的数据提供方模型进行扩展，那么您的第一个问题可能就是，“我从哪里开始呢？”

从哪里开始？

我们建议您首先完整地阅读本附录和参考实现，以熟悉相关概念及这些概念彼此间的关系。准备好开始时，首先构建一个非常简单的数据提供方，以便充分了解此工作涉及的需求。即使最终的数据提供方需求涉及范围非常广，仅使用必需的元素构建基本数据提供方也能让您很好地了解此数据模型。

“[所需组件](#)”各小节中给出了最基本的元素。请学习“[可扩展数据提供方概述](#)”的内容，以了解各个接口如何彼此相关，以及这些接口与 **MapXtreme** 的数据访问模型的关系。回顾所提供的“[示例：COTW \(Center of the World\) 数据提供方](#)”，以了解我们如何实现基本数据提供方，并复习随 **MapXtreme** 示例源代码安装的全功能数据提供方“[SpatialLite 示例数据提供方](#)”。

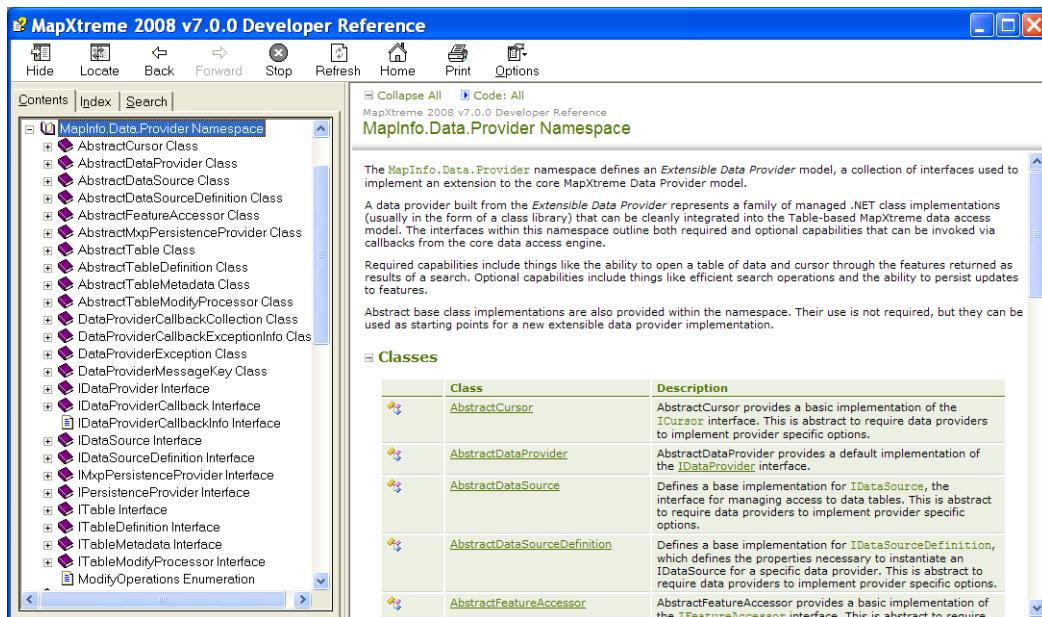
我们还建议学习 **SDK** 中提供的可选构建块，以了解已经实现了哪些内容。请参阅[可选构建块：基类、帮助程序和实用程序](#)。

创建数据提供方的前期工作之一是，确定数据提供方是否有数据源。并非所有数据提供方具有或需要这种模型，数据提供方可能采用基于表的方式。数据源是与之通信、从而通过相同连接对可能存在的多组图元（或表）进行访问的任何服务或数据库。数据库服务器始终作为数据源公开，而数据库中的表作为表公开。**WFS** 服务器将建模为数据源，而 **WFS** 服务器公开的 **FeatureType** 将建模为表。**ADO.NET DataSet** 可以作为数据源公开，而其中包含的 **DataTable** 则将作为表公开。**Excel** 电子表格可以作为数据源公开，而选择范围则作为各类表公开。如果采用更为简单的数据模型，例如

ASCII 文本文件或仅包含单个图元集的其他文件格式，则可能不需要数据源，可以仅使用表模型直接对数据提供方进行建模。数据可能虽然非常复杂但仍然不需要数据源模型。例如，有些基于文件的数据格式可以将图元拆分存储在多个文件中，可能按州或省或其他规则划分。这将仍然是基于表的数据提供方，将不需要数据源。

可以在哪里获得更为详细的帮助信息？

组成可扩展数据提供方体系结构的接口在 `MapInfo.Data.Provider` 和 `MapInfo.Data` 命名空间中定义。我们建议您通过《[MapXtreme 参考指南](#)》熟悉 `MapInfo.Data.Provider` 命名空间。



所需组件

要构建基本数据提供方，必须实现以下组件。这些类支持基本数据访问操作，包括打开表、读取表内容及关联的元数据和搜索表内容。

请将 [SpatiaLite 示例数据提供方](#) 作为完整且可靠的提供方加以参考，以了解关于如何实现这些接口的技巧和指南。

IProvider 接口

`IProvider` 接口提供自定义数据提供方实现的基础，公开用于打开此提供方所提供的表格式的功能。它还公开与管理这些表的数据源相关的功能。实现 `IDataSource` 接口的任务是可选的，因为可以无需通过数据源构建数据提供方。该类通常定义为单态实例。

ITableDefinition 接口

ITableDefinition 接口提供可扩展数据提供方和 **MapXtreme** 公共 API 表模型之间的主要链接。它提供对特定数据提供方的 **ITable** 进行实例化所必要的属性。实现此接口的类确定 **MapXtreme** 要打开表时需要提供的信息。**MapXtreme Catalog** 提供了 **OpenTable** 方法的一些重载，允许您和数据提供方的其他用户从 **ITable** 实例化表。调用了 **OpenTable** 后，**ITableMetadata**、**ICursor**、**IFeature** 等所有其他接口都对用户隐藏，供 **MapXtreme** 在内部使用。

此接口通常还是会希望写出到工作空间文件的信息。有关详细信息，请参阅[永久性提供方](#)。

ITable 接口

ITable 接口定义了表的接口，表是 **MapXtreme** 可以读取的详细的基本容器。在地图中查询和显示。**ITable** 上的方法包括 **SearchAll**、**SearchByEnvelope**（重点区域）和 **SearchByKey**。表代表单个图元集合，所有图元必须包含相同的结构（或架构）。

ITableMetaData 接口

ITableMetaData 接口用于传递关于打开的表的属性和所支持的功能的重要信息，如列及其数据类型、表中包含的几何体类型以及是否支持编辑操作。

表元数据将表的架构作为列定义集合公开。列通过 **IColumn** 和 **IGeometryColumn** 接口公开。任何返回 **FeatureGeometry** 对象的列都必须公开为 **IGeometryColumn**。**IGeometryColumn** 还允许提供方指示在插入和更新操作期间返回和接受的几何体值中是否支持 Z 和 M 尺寸。大部分数据提供方都将采用 **IColumn** 和 **IGeometryColumn** 的自定义实现，以帮助在 **MapXtreme** 搜索请求和底层提供方实现之间进行信息通信。

ICursor 接口

ICursor 接口是返回 **IFeature** 对象的 **IEnumerator**。当对 **ITable** 发出搜索请求时，所得到的是一个 **ICursor**，此对象提供对图元的访问。从光标返回的图元可能为瞬态的，即图元仅预计在光标状态未发生变化时有效。将光标前进到下一记录（图元）可能会返回相同的对象，此对象可能已经使用当前记录的新值进行了更新。强健的数据提供方实现应该利用此行为来重用内存。

搜索请求可能会请求表公开的属性（列）子集。从光标返回的图元必须包含搜索请求的列的结构。

IFeatureAccessor 接口

IFeatureAccessor 用于使用键从表获取图元。它公开了一个主方法，名为 **FetchByKey**，此方法返回由给定的键标识的图元。数据提供方返回的每个图元都必须包含键。键的定义和内容由数据提供方确定。当 **MapXtreme** 向数据提供方给出键，以进行 **FetchByKey** 请求或编辑操作时，数据提供方将使用键值标识正确的记录。**IFeatureAccessor** 接口提供了让 **MapXtreme** 对一系列调用进行组合的方法，因此在需要的情况下，数据提供方可以准备一个命令，然后直接对每个 **FetchByKey** 调用绑定键值并执行此命令。

可选构建块：基类、帮助程序和实用程序

MapXtreme 可扩展数据提供方是接口的集合，可以实现这些接口来访问采用当前不支持的格式的数据。**MapXtreme** 提供了一组实用程序、示例和指南，能帮助您快速入门，着手创建您的实现。这些是大多数人都希望使用的公共实现。它们涉及以下区域：

- ◆ 抽象基类
- ◆ 实用程序类
- ◆ 键实现

有必要在开始构建甚至最为简单的实现前了解一下这里提供了哪些部分，因为有些已经构建了。

抽象基类

提供了很多接口的抽象基类。抽象基类用于处理接口的默认实现，因而您只需要实现所需的部分即可。

例如，**IDataProvider** 提供了一个 **OpenDataSource** 方法和两个不同风格的 **OpenTable** 方法（一个使用数据源，一个不使用数据源）。如果使用数据源，则必须实现 **OpenDataSource** 和基于数据源的 **OpenTable**。如果没有数据源，则只需要实现不需要数据源的 **OpenTable**。

IDataProvider 的抽象基类对于每个方法都有默认实现，即这些方法不受支持。这些方法标记有“*virtual*”字样，因此在选择实现其中的方法时，仅需要提供所需的部分的覆盖实现即可。

实用程序类

简单图元

数据提供方的基本目标是访问图元。图元通常为表中的行。空间图元由它们的几何体、样式、键和属性描述。

我们提供了名为 **SimpleFeature** 的实用程序类，该类实现了 **IFeature** 接口。它实现定义元数据的关系 (**SimpleFeatureMetadata**) 和图元的属性列表 (**AttributeValues**) 以及标识键的值 (**IKey**)。大部分数据提供方都将希望使用 **SimpleFeature** 类。

SimpleFeature 还实现 **FeatureChangedEventHandler** 事件管理。

SimpleFeature 记录在 **MapInfo.Data.Common** 命名空间下。

几何体和坐标系的 OGC 转换

MapXtreme 可扩展数据提供方包括帮助进行在 OGC 几何体（**OpenGIS®** 简单图元接口标准）和 **MapXtreme** 几何体之间进行转换的过程。这包括 OGC 所规定并被大量空间数据系统（如 **PostGIS**、**MySQL** 和 **SQL Server 2008**）采用的已知文本和已知二进制格式。已知文本格式指空间参考系统的标准文本表示形式。已知二进制格式是几何体的标准二进制表示形式。

这些实用程序包括在 **MapInfo.OGC** 命名空间中。它实现已知文本和已知二进制格式的数据读取器和写入器，同时支持从大到小和从小到大的字节顺序。

键实现

对于要由 MapXtreme 数据提供方访问的数据，要求每个图元都具有一个键。我们在此 SDK 中提供两个常见的键实现：整数键和字符串键。

如果您的数据基于整数或字符串，则可以跳过 IKey 接口的键实现工作，使用提供的现有实现。

示例：COTW (Center of the World) 数据提供方

“Center of the World” 数据提供方是一个非常简单、专门设计的示例，用于演示如何编写数据提供方实现。此数据提供方管理一个表，表中具有单个行，其中包含使用经度/维度 (0,0) 表示的空间点。

要开始实现新可扩展数据提供方，一个合适的着手点是从实现 IDatasProvider 类开始。该类要求作为单态实例实现，以便对其的所有引用（例如，从 ITableDefinition 和 IDatasourceDefinition 引用）唯一地解析到相同的实例。以下是示例实现的开始部分：

```
using System;
using MapInfo.Engine;
using MapInfo.Data.Provider;

namespace COTW
{
    public sealed class COTWDataProvider :
    {
        private static string PROVIDER_NAME = "Center of the World
Sample Extensible Data Provider for MapXtreme";
        private static COTWDataProvider m_singleton = null;

        private COTWDataProvider(string name)
            : base(name)
        {
        }

        public static COTWDataProvider GetInstance()
        {
            if (m_singleton == null)
                m_singleton = new COTWDataProvider(PROVIDER_NAME);
            return m_singleton;
        }
    }
}
```

注意，该类从 AbstractDataProvider（而不是 IDatasProvider 本身）扩展。正如“[可选构建块：基类、帮助程序和实用程序](#)”所讨论的，我们提供了抽象基类作为构建块来尽可能帮助提供合适的默认实现。对于 IDatasProvider，此抽象类管理 name 属性和其 OpenDataSource 和 OpenTable 方法的默认实现，以免抛出未实现异常。通过这样，我们可以决定哪些部分相关，并仅为这些提供实现，从而保持实现的整洁有序。

由于我们所定义的表具有固定结构，因此 **ITableDefinition** 实现不需要太多的工作。为了让这个示例有所意义，我们将让表定义接受字符串值，可以将此值作为在外部为点指定的标签。完整的实现将与以下所示类似：

```
using System;
using MapInfo.Data.Provider;

namespace COTW
{
    public sealed class COTWTableDefinition : AbstractTableDefinition
    {
        private string m_label;

        public COTWTableDefinition(string label)
            : base()
        {
            if (label == null)
                throw new ArgumentNullException("label");
            m_label = label;
        }

        public override IDataProvider DataProvider
        {
            get { return COTWDataProvider.GetInstance(); }
        }

        public string Label
        {
            get
            {
                return m_label;
            }
        }
    }
}
```

我们没有直接使用接口，而是再次使用了抽象基类。在这种情况下，这样做允许忽略任何相关部分，接受接口上的 **DataSourceDefinition** 和 **CustomMetadata** 属性的默认值。请注意 **DataProvider** 属性如何引用前面实现的单态 **COTWDataProvider** 实例。此示例还引入了我们自己的一个新属性。

可选接口

MapXtreme 可扩展数据提供方模型包括可选接口，用于在访问数据时提供更多的功能，包括：

- **IDataSource**
- **IDataSourceDefinition**
- **ITableModifyProcessor**

IDataSource

IDataSource 接口在 **Catalog.OpenTable** 中用于将新表与特定数据源关联。此接口是可选的，因为可以不使用数据源构建数据提供方。

数据源在打开 **ITableDefinition** 时实例化（**ITableDefinition** 中包含关联的 **IDataSourceDefinition** 或直接通过 **Catalog.DataSources.OpenDataSource** 方法进行）。将提取的 **IDataSourceDefinition** 提供给 **IDataProvider.OpenDataSource**，以连接到数据源并返回关联的 **IDataSource** 实例。

IDataSourceDefinition

DataSourceDefinition 仅在使用数据源访问数据时才为必需的。它包括用户为了能实例化数据源而预计提供的信息。如果您未使用数据源，则只需要提供 **Tabledefinition** 就能打开表。

在打开表的过程中，将从 **DataSourceDefinition** 属性提取 **IDataSourceDefinition**，并将其传入 **OpenDataSource(IDataSourceDefinition, CustomProperties)**，以尝试建立到数据源的连接。

实现 **DataSourceDefinition** 时，应该提供 **System.Object.Equals(object)** 有意义的覆盖实现。

ITableModifyProcessor

表元数据指示是否支持 **Insert**、**Update** 或 **Delete** 操作。如果支持其中的任意操作，则必须提供 **ITableModifyProcessor** 的实现。必须根据对应的各个元数据属性提供特定插入、更新和删除操作的实现。例如，有些数据提供方可能仅支持插入操作。例如，随 **MapXtreme** 提供的 **ASCII** 数据提供方就仅支持插入操作，而不支持更新或删除操作。

注意，**ITable** 接口还包括名为 **ReadOnly** 的属性。此属性提供额外的控制级别，可用于指示表是否为只读。即使您的 提供方提供完整实现的修改处理器，某个特定的表也可以因为其他原因而为只读状态，例如访问权限不足或数据文件位于只读介质上。您的数据提供方可能事先对此进行检查，并将整个表标记为 **ReadOnly**，或通过各个表元数据属性进行设置。

构建和测试数据提供方

构建数据提供方是一项复杂的任务，而且由于大部分数据提供方调用都是由 **MapXtreme** 发出的，因此很难进行调试。您将希望分阶段构建数据提供方，并在此过程中仅测试所实现的部分。为了更好地对此加以控制，您可能会希望从示例开始着手。因为在示例中，您可以进行练习，仅编写要实现的代码，而且还可以对示例进行扩展，以扩展数据提供方。不应该直接将这包括在数据提供方项目中；不过，**Visual Studio** 允许在单个解决方案中创建多个项目。

如果在数据提供方模型中有数据源，则将要从那里着手。创建 **IDataSourceDefinition** 和 **IDataSource** 的实现，并尝试使用 **Catalog.DataSources.OpenDataSource** 方法来确定 **MapXtreme** 是否调用您的数据提供方以及 **DataSources** 集合中的数据源对象。

表模型的第一个重要里程碑是能够打开表并显示表的元数据。您可以立即开发和测试此功能，而无需编写任何复杂得多的搜索或光标代码。您将需要创建 `ITableDefinition`、`ITable` 和 `ITableMetadata` 的实现。最初您可以将 `search` 和 `modify` 接口实现为抛出 `NotImplementedException`。在测试应用程序中，您应该能够创建和填充表定义类的示例，通过将此定义提供给对 `Catalog.OpenTable` 的调用来自实例化表，然后检查表的 `TableInfo` 属性提供的列定义。列定义应该准确地反映您的表元数据向 `MapXtreme` 提供的信息，包括正确的几何体列坐标系、数据绑定和默认视图。

构建并测试提供方的下一步是添加非常简单的搜索并提供光标实现。在您的测试应用程序中，可以添加与以下所示类似的简单代码块：

```
Table table = catalog.OpenTable(myTableDefinition, "MyTable");
foreach (Feature feature in table)
{
    // inspect the feature, write out values to the console, etc...
}
```

为了使此代码工作，您将需要实现 `ITable.SearchAll` 方法，并提供光标实现。您还将需要提供键实现，在上面的代码示例中返回的代码应该反映键的相应值（不过值序列化为字符串）。

这些基础级别的代码工作正常之后，您就可以开始在地图中对数据提供方进行测试，以便通过可视的结果来确定几何体是否按照预期的方式返回、是否具有正确的样式，等等。在此前，您将需要添加 `SearchByEnvelope` 的实现。您可以首先忽略 `areaOfInterest` 参数，返回全部数据来了解数据的情况。由于每个请求都将检索所有图元，因此性能将较差。您应该关闭图层的 `InfoTip` 属性，从而在地图上移动鼠标时阻止 `MapXtreme` 尝试显示信息提示。`InfoTip` 使用鼠标所在的点周围的小范围重点区域调用数据提供方，如果返回所有的表图元，性能将非常差。在实现图元访问器之前，也不能测试选择或结果集功能。

继续下一步工作之前，您应该添加对 `SearchByEnvelope` 方法的支持。这将有助于了解数据提供方的性能，因为您现在能够仅将渲染地图所需的图元发送回 `MapXtreme`。您还可以重新打开 `InfoTip` 设置，并将 `InfoTip` 表达式设置为各种值，以确定是否能将正确的数据传递给 `MapXtreme`。如果性能令人不能接受，那么在增加复杂性之前，此时就是评估设计并对其进行调整的好时机。

实现图元访问器接口将是下一个主要里程碑。对此接口进行测试的最简单方法是，创建结果集图元集合，然后枚举图元。在内部，结果集图元集合将缓存满足搜索条件的记录，然后只要访问此结果集就使用图元访问器。

只要让所有这些部分正确工作，就完成了数据提供方构建工作中最难部分的绝大部分任务。您将很可能希望将重点放在构建永久性提供方，以便将定义永久保存到 `MapXtreme` 工作空间文件中。此时可以考察新 `WorkspaceManager` 可扩展性功能，并考虑添加 `WorkspaceManager` 扩展，以提供用于定义表的用户界面，并允许挂钩到永久性提供方扩展中，以便设置和创建工作空间，并使用数据提供方永久性内容读取工作空间。

此时，您应该将着重进行在简单桌面应用程序环境中测试数据提供方的工作。实现了永久性机制后，则可以开始考虑基于 Web 的部署。这将需要序列化支持，以便跨用户请求对表或整个目录或会话进行序列化。

Spatialite 示例数据提供方

可扩展数据提供方 SDK 中包括了数据提供方的示例实现，其中重点演示了可扩展数据提供方体系结构的很多功能。MapInfo.Spatialite.sln 是 Visual Studio 解决方案，使用了 SQLite3 Ado.Net 数据提供方建立到底层 SQLite3 数据库的连接，并在 MapXtreme 应用程序中打开表。

这个示例实现还对 MapXtreme 的其他可扩展方面进行了重点介绍，包括会话初始化时的自动加载自定义代码和 Workspace Manager 的扩展。此示例位于您的 MapXtreme 安装目录的 \Samples\DataProviders 文件夹中。

i 与编译并提供可运行应用程序的 MapXtreme 其他示例应用程序不同，此示例实现生成 .NET 程序集。此示例的开发重点是展示 MapXtreme 中的可扩展选项，而不是介绍能正常工作的应用程序。

MapInfo.Spatialite.sln 包含两个项目：

- Spatialite Data Provider
- Spatialite Workspace Manager Extension

Spatialite Data Provider Extension

此项目中的代码显示如何对 MapXtreme 可扩展数据提供方类和接口进行扩展来创建 Spatialite 数据提供方，以便打开 Spatialite 表、插入/更新/删除记录、搜索表、更改样式和将新表导入到 SQLite3 数据库中。抽象基类用于帮助更为高效地进行代码编写。请参阅随示例提供的 Readme.rtf，以了解哪些类和接口进行了扩展的详情。

编译时，Spatialite Data Provider 项目创建了名为 MapInfo.Spatialite.DataProvider.SessionEventHandler.dll 的程序集（作为主数据提供方实现使用）和会话事件处理程序（便于在 MapXtreme 启动时将该数据提供方用户自动加载）。

Spatialite 数据提供方程序集还包含永久性提供方的实现，这个是数据提供方的可选组件，但非常有用。这样就可以使用 MapXtreme 不以本地方式读取/写入 MapInfo 工作空间 (.MWS) 的 Spatialite 表和数据源定义。请参阅 [永久性提供方](#)。

Spatialite 数据提供方中的 ISessionEventHandler 实现是另一个有用的组件，也是可选的。将此组件包含在数据提供方程序集中，以便 MapXtreme 启动时自动加载数据提供方。这意味着 Spatialite 表定义可以在 MapXtreme 会话初始化完成之后立即加载并在 MapXtreme 应用程序中可用。

SessionEventHandler 并不特定于数据提供方可扩展性。可以将其用于加载您在会话启动时所需的任何自定义代码。我们利用此功能来对自定义 Spatialite 表和数据源定义进行永久化。为了将这个用于其他自定义代码，程序集必须包含后缀 SessionEventHandler.dll，并置于 \Common Files\MapInfo\MapXtreme\7.x.x\SessionEventHandlers 中，MapXtreme 才能对其予以识别。有关此主题的详细信息，请参阅 [ISessionEventHandlers](#) Spatialite Workspace 管理器扩展

对 MapXtreme 的另一个主要增强是，能够扩展 Workspace Manager 的功能。SpatiaLite Data Provider 示例中包含的 SpatiaLite Workspace Manager Extension 项目就是这方面的一个例子。此项目对 Workspace Manager 进行扩展，以提供用于连接到数据库和打开表的 SpatiaLite 菜单、对话框和代码。

编译此项目时，它会生成带 WorkspaceManagerExtension DLL 后缀的程序集。此程序集依赖于提供 SpatiaLite 数据源的表定义的 SpatiaLite 数据提供方。

Workspace Manager 扩展可以包含在创建和保存工作空间时可能需要的任何功能。有关这些功能的讨论，请参阅[附录 B：自定义 MapXtreme](#)。

高级主题 / 重点考虑事项

本节包括在设计和构建可扩展数据提供方时值得考虑的各个主题：

- ◆ [创建几何体](#)
- ◆ [坐标系](#)
- ◆ [样式](#)
- ◆ [异常处理](#)
- ◆ [永久性提供方](#)
- ◆ [序列化](#)
- ◆ [身份验证](#)
- ◆ [线程安全](#)

创建几何体

实现了第三方空间格式支持的可扩展数据提供方必须在原始格式和 MapXtreme FeatureGeometry 格式之间对空间数据进行转换。FeatureGeometry 对象作为光标和图元访问器中包含的 IFeature 对象的值返回。图元和其中包含的几何体对象都为瞬态的。MapXtreme 将仅假定这些对象在光标打开且定位在当前图元上时有效。因此，最为强健和可扩展的数据提供方实现将尝试在光标或图元访问器的生命周期内重用相同的图元和图元几何体。

如果几何体在创建之后针对当前图元持续更新，则将其称为“瞬态”几何体。MapXtreme 的几何体模型包含构造函数和方法签名，方便以这种方式使用几何体。在 MultiPoint、MultiCurve 和 MultiPolygon 对象的初始构造之后，可以在这些现有实例上使用几何体编辑器接口来针对新图元进行更改。对于 MultiCurve 和 MultiPolygon 对象，可以使用 Clear 方法清空几何体，然后可以分别使用 AddCurve 和 AddPolygon 来重新定义几何体。对于 MultiPoint 对象，可以使用 ReplaceAll 方法进行重新定义。而且，使用输入数组的构造函数签名和 Add/Replace 方法支持可选的大小指派，从而还能高效地重用单个数组实例。

OGC 转换代码就以这种方式工作。用于传递已知二进制格式和已知文本的内部字节数组会被重用并根据需要增长。当数据提供给相应的几何体方法（如 AddPolygon）时，将使用接受数组大小的格式，因为输入数组可能比所提供的数据长。

MultiPolygon 对象由多个多边形组成，这些多边形通常由 **MapXtreme** 进行检查，以确定是否有任意多边形彼此间存在内/外关系；例如，环状的 **MultiPolygon** 实例可能由一个外部多边形组成，而另一个内部多边形被视为剪切方，二者共同在逻辑上表示单个几何体。几何体代码默认情况下并不进行任何假设，编辑完成时，它将分析所有成员多边形，以搜索是否存在这些关系。如果所转换的第三方空间格式已经知道这些关系，并可以按照正确的顺序添加多边形（外部多边形紧跟在任何关联的内部多边形后添加），则 **MapXtreme** 不需要投入大量开销对所有多边形进行重新分析。此操作可以使用 **MultiPolygon** 类上重载的 **EditingComplete** 方法并向 **withinSpecified** 参数传入“true”来禁用。

此处理时间可能会很长。在移动单个节点可能会让内部多边形突然变成外部多边形的情况下，有必要提供交互式编辑操作。数据提供方通常已经知道其数据已经进行了格式化，并以恰当的方式存储，不需要每次构建几何体时都进行这个开销巨大的处理。

对于大量几何体和/或包括大量节点的复杂几何体，为了实现高性能数据提供方，了解并利用这些几何体选项至关重要。

坐标系

MapXtreme 中的几何体列必须采用相同的坐标系返回 **FeatureGeometry** 对象。不同的列和不同的表可能使用不同的坐标系，但从表中的单个列返回的所有几何体值必须使用相同的坐标系，而且必须与表的元数据通过 **IGeometryColumn.CoordSys** 属性提供的坐标系统匹配。

大部分数据提供方都将通过随数据一起存储和管理的某些元数据来确定坐标系。不过，如果此信息不可用，则数据提供方请求在表的定义中提供坐标系。

可以通过 **CoordSysFactory** 实例构造坐标系，此实例通过 **Session.CoordSysFactory** 属性提供。不需要为数据提供方返回的每个 **FeatureGeometry** 对象提供新坐标系对象。**IGeometryColumn** 接口返回的坐标系对象可以直接由每个 **FeatureGeometry** 对象引用。向数据提供方提供的用于插入和更新操作的 **FeatureGeometry** 对象将会由 **MapXtreme** 转换为列的坐标系，然后再传递给数据提供方。

样式

MapXtreme 的可扩展数据提供方模型以两种方式支持为图元几何体使用样式：

- 作为应用到表中的所有图元的样式属性
- 作为样式属性列（其中包含按图元指定的样式信息）

从可扩展数据提供方表打开的 **MapInfo.Data.Table** 将具有样式列，前提是至少有一个几何体列（可绘制地图的表）。该列的数据来自可扩展数据提供方表的样式列，或者，如果没有这样的列，则来自数据提供方表中的几何体列等等 **IGeometryColumn.DefaultStyle** 属性上指定的默认样式。

ITableMetadata.Columns

样式信息通过 **MapInfo.Data.Provider.ITableMetadata** 接口中提供的列定义传递。样式属性列是 **ITableMetadata.Columns** 的 **IColumn** 实例，其 **DataType** 属性分配的值为 **MIDbType.Style**。样式列仅在表中两个或更多图元可能包括不同的样式值时才有必要使用。

I`GeometryColumn.DefaultStyle`

如果希望在表的一个图元几何体列之一中渲染对象时统一地应用单个样式，则不需要样式属性列。为此，可以直接将样式对象实例作为图元几何体列的 `DefaultStyle` 属性进行管理 (`IGeometryColumn.DefaultStyle`)。

始终建议提供默认样式，即使存在样式列是也建议如此。默认样式用于在样式列为空或没有值时填充缺失的样式属性值。对于缺少显式提供的默认样式的情况，**MapXtreme** 提供了内部缺省样式，可以在需要时应用。

如果仅在图元几何体上使用默认样式，样式列仍然将显示在与 **Catalog** 内打开的结果 **Table** 实例关联的列集合中。此列将可以为空和只读，所有值均默认设置为指定的默认样式实例值。

虽然没有明确禁止在可扩展数据提供方实现中定义多个几何体列和/或多个样式列数据配置，但元数据模型并不支持将样式列与几何体列显式关联。渲染地图图层时，有个隐式的假设，即用于该图层的表包含单个几何体列，而且最多有一个邻近的样式列。与此类似，模型还允许可扩展数据提供方定义包含多个样式列但没有相邻几何体列的表，这些表当前不能缓存，不能导出为其他格式。

异常处理

数据提供方中的异常处理非常重要。数据提供方通常将处理各种系统资源，如文件句柄或数据库连接。如果这些资源未在正常和异常代码路径中清理，则可能会泄漏，导致应用程序行为异常或长时间积累而导致内存泄漏。这些问题通常在利用会话池技术的 **Web** 应用程序中变得较为突出，内存泄漏可能最终会导致系统关闭或多个进场挂起以等待释放泄漏的连接。

抛出的异常应该遵循 Microsoft 在《[异常处理最佳实践](#)》中规定的标准 .NET 实践。抛出自定义异常时，请使用 `MapInfo.Data.Provider.DataProviderException` 类。该类可能会派生子类以在必要的情况下提供其他行为。

异常处理也是考虑将数据提供方外部化的好时机。通过恰当地将资源（如字符串、位图等）捕获在文件中，可以针对其他文化对您的数据提供方进行本地化。**Spatialite Data Provider** 实例提供了用于处理资源字符串的异常处理参考实现（请参考项目中的 `Resources.cs` 文件）。

永久性提供方

MapXtreme 提供了将数据访问信息（即 `ITableDefinitions` 和 `IDataSourceDefinitions`）保存到基于 XML 的 **MapXtreme** 工作空间 (.MWS) 的支持。这是可扩展数据提供方的可选组件，如果需要与其他人共享工作空间或直接在以后重用此信息，则可能会希望使用此组件。

为了向可扩展数据提供方添加持久性支持，最简单的方式是，编写一个实现 `MapInfo.Data.Provider.IMxpPersistenceProvider` 接口的永久性提供方类。

对于更为高级的永久性提供方，请考虑提供架构来支持对所得到的工作空间文件进行 XML 验证。仅有在明确的需要时才有必要这样做，因为 **MapXtreme** 并不自动对工作空间 XML 进行验证。

您还可以考虑将永久性提供方单独作为程序集编写。提供方可以独立于其他可扩展数据提供方组件进行预载。提供方程序集将推迟其他组件的加载，控制仅在明确需要时实时加载数据提供方。

如何使用永久性提供方？

永久性提供方在 Catalog 上的 PersistenceProviderCollection 中进行管理。在 MapXtreme 搜索默认位置中的程序集来寻找可能包含的持久提供方时，此集合可以在会话启动期间自动初始化。默认位置为 \Program Files\Common Files\MapInfo\MapXtreme7.x.x\SessionEventHandlers。还可以采用编程方式使用 Add 和 AddFromFolder 方法将提供方添加到 PersistenceProviderCollection 中。

通过 MapInfo.Persistence.WorkspacePersistence 类将会话上下文保存到 MapInfo 工作空间文件 (.MWS) 时，都会查询 PersistenceProviderCollection，以确定是否有从可扩展数据提供方打开的目录中的任何表。如果为该数据提供方标识了永久性提供方，则将其用于把可扩展表信息保存到工作空间。类似地，当加载工作空间时，将搜索可用永久性提供方集合，以确定数据提供方是否能够理解与可扩展数据提供方表关联的工作空间文件中的自定义标记。

实现永久性提供方

为了让 PersistenceProviderCollection.AddFromFolder 方法加载您的永久性提供方类，其中必须包含一个零参数公共构造函数签名。在 MapXtreme 的会话初始化期间内部使用的也是这个方法。可以提供额外的构造函数签名来支持以编程方式使用 PersistenceProviderCollection.Add 方法。

MapXtreme 提供了名为 AbstractMxpPersistenceProvider 的抽象基类，您可以将其作为实现永久性提供方类的着手点。

必须在永久性提供方中提供的东西之一是数个方法，用于检查是否有您的永久性提供方的受支持数据提供方、受支持的实体名称（用于 XML 标记）和受支持的架构命名空间。此工作通过实现 IMxpPersistenceProvider 接口中包括的三个查询方法来完成。

SupportsDataProvider() 用于标识负责给定 IDataProvider 的 ITableDefinition 和 IDataSourceDefinition 构造。通常，永久性提供方编写为支持单个数据提供方，因此这个方法的实现通常非常简单，只需要确定所提供的 IDataProvider 是您的数据提供方类的实例即可。

SupportsSchemaNamespace 和 SupportsEntityName 在工作空间永久化期间一起用于确定支持为可扩展数据提供方内容标识的命名空间和标记名称的永久性提供方。例如，在读取工作空间文件期间，遇到了可扩展数据提供方的以下内容：

```
<sample:SampleTableDefinition xmlns:sample="http://sample/sample">
    <sample:TableName>test</sample:TableName>
</sample:SampleTableDefinition>
```

对于外部标记，获得了一个 XmlNode 对象引用。向 SupportsSchemaNamespace 提供了 NamespaceURI 属性 ("sample")，向 SupportsEntityName 提供了 LocalName 属性 ("SampleTableDefinition")，以标识同时支持二者的永久性提供方。通过抽象类可用的构造函数签名也支持这些方法的默认实现。

AbstractMxpPersistenceProvider 包含构造函数签名，此签名与 IMxpPersistenceProvider.Schema 的默认实现结合使用，用于返回对程序集中以嵌入资源的形式附加的 .xsd 架构文件的 XMLSchema 引用。详细信息，请参阅《开发人员参考》。

IMxpPersistenceProvider 上剩下的方法是数据源和表定义的读/写方法对。对于不使用数据源的数据提供方，仅需要实现表定义的读/写方法。如果采用数据源，则需要将数据源定义永久化。MapXtreme 将为您进行此工作。您可以只重点关注您的表都有的那些表定义属性。

读方法包含可扩展内容的 XML 标记的 `XmlElement` 句柄。数据源和表定义的组件作为该元素内的子节点管理。对于上面的 `SampleTableDefinition` 示例，实现代码可能与以下所示相似：

```
ITableDefinition td = null;
if (node.LocalName.Equals("SampleTableDefinition") &&
node.NamespaceURI("http://sample/sample"))
{
    string tableName = null;
    foreach (XmlNode childNode in node.ChildNodes)
    {
        if (childNode.LocalName.Equals("TableName") &&
childNode.NamespaceURI.Equals("http://sample/sample"))
        {
            tableName = childNode.InnerText;
        }
    }
    td = new SampleTableDefinition(tableName);
}
```

写方法必须创建插入到工作空间 XML 的 `XmlElement`。元素绑定到 XML 文档实例，因此随可扩展数据提供方提供的 `XmlDocument` 需要永久化。注意，对于 `DataSourceDefinition` 写方法， `XmlDocument` 直接作为方法参数提供，而对于 `TableDefinition` 写方法， `XmlDocument` 通过 `IMxpPersistenceServices` 参数的 `Document` 成员提供。

```
XmlElement outerTag = xmlDocument.CreateElement("sample",
"SampleTableDefinition", "http://sample/sample");
XmlElement innerTag = xmlDocument.CreateElement("sample", "TableName",
"http://sample/sample");
innerTag.InnerText = "test";
outerTag.AppendChild(elNode);
```

有关其他信息和示例，请参考《开发人员参考指南》文档、[SpatiaLite 示例数据提供方](#)和[《MapXtreme 代码交流》](#)中的其他数据提供方示例。

序列化

序列化是将对象转换为数据流的过程，以便在有用的时期内采用永久形式或在内存中加以保持。此过程是在 **MapXtreme Web** 应用程序和多线程桌面应用程序中维护对象的重要部分。

如果不进行序列化，将需要经常重建对象，例如在会话期间每次出现请求该对象的 Web 请求时。

序列化对象被请求时，首先反序列化对象（或从数据流重建），然后修改对象。**MapXtreme** 的序列化算法不生成对象的副本（与其他序列化算法相同），所以反序列化的对象仅创建一次。

为了对 Web 应用程序和多线程桌面应用程序进行恰当的状态管理，应用程序开发人员经常需要直接对 **MapXtreme Table** 实例进行序列化，或者通过对包含所有表的集合的 **MapXtreme Catalog** 进行序列化来实现此目的。当表类型不受支持时，将需要由应用程序开发人员确定如何显式地管理这些表及其 **Catalog** 的总体状态，以围绕此限制进行操作。为此，鼓励可扩展数据提供方支持其提供方的序列化，以恰当地集成到 **MapXtreme** 的 **Table** 序列化工作流中。

必须对提供方实现的哪些组件进行序列化？

MapXtreme 将表区分为永久表和临时表。永久表只需要将其定义/结构序列化，就能在反序列化时恰当地重新构造。临时表还需要负责其数据的序列化。可扩展数据提供方表被视为永久表，因而能够简化整体序列化责任，不过仍然必须对多个提供方类进行序列化。

对于支持数据源概念的提供方，**IDataSourceDefinition** 和 **IDataSource** 接口的类实现都必须可序列化。无论是否支持数据源，每个提供方还都必须支持实现 **ITableDefinition** 和 **ITable** 接口的类的序列化。**ITableDefinition** 和 **IDataSourceDefinition** 接口还必须提供有意义的 **Equals** 方法重载，以保证反序列化正常工作。

实现 **ITableMetadata** 的类并不明确地需要可序列化，不过这样做有一些较为实际的原因。如果不可序列化，相应的表和数据源实例的反序列化可能会需要对其进行重新构造，而重新获得此元数据在性能方面可能开销极大。

为了帮助您使用，对于其中的每个接口，可扩展数据提供方 API 都提供了可序列化的抽象基类。还为由默认实现关联的相关属性提供了序列化支持。

如何序列化类？

在 .NET 术语中，通过实现来自 **System.Runtime.Serialization** 命名空间的 **ISerializable** 接口并/或向类定义应用 **[Serializable]** 属性来将类变成可序列化的。在提供实现时，提供方开发人员可以自由地遵循关于 .NET 序列化的一般指南；不过，我们强烈建议同时应用二者，特别是从所提供的抽象基类派生您的类时更应如此。

如果决定不支持序列化会如何呢？

我们建议在可扩展数据提供方中提供序列化支持，以便为将要在其解决方案中采用您的提供方的应用程序开发人员提供更好、更广泛的支持。不过，并不严格要求提供方实现必须支持序列化。事实上，在某些情况下，由于法律方面的原因，有时候提供此类支持基本上很难或不可靠。

应用程序开发人员可以使用 .NET 反射 API 来发现您的提供方是否支持序列化；不过，这仍然不能提供所需的全部信息，不便于了解如何在具有状态管理需求的应用程序中恰当地采用您的提供方。我们非常鼓励提供相关文档，帮助了解关于您的提供方实现能否/如何作为此类解决方案的组件使用。

序列化和依赖关系

MapXtreme 的可扩展数据提供方包含很多存在于需要和建议使用的类之间的依赖关系。**ITableDefinition** 包括 **IDataSourceDefinition** 属性。**IDataSource** 还包含 **IDataSourceDefinition**，**ITable** 包含 **IDataSource**、**ITableMetadata** 和 **ITableDefinition** 属性。

一般来说，您可能不需要对引用可能会共享的任何属性进行序列化。例如，**ITableDefinition** 上的 **IDataSourceDefinition** 属性可能代表重用于打开多个表的数据源定义。您可能不需要在类实现 **ITableDefinition** 中包括此属性。对于 **ITable** 中的 **IDataSource** 和 **ITableDefinition** 属性也是如此。

我们建议在序列化时包括定义属性，即便稍后被覆盖来使用共享引用也应如此。

ITable 中的 **IDataSource** 引用不应序列化，因为 **MapXtreme** 将负责自动对表的数据源进行序列化。

共享对象引用通过 **MapXtreme** 提供的序列化逻辑重新建立。可扩展数据提供方 API 包含核心 **MapXtreme** 数据访问引擎完成此工作所必需的挂钩。具体的示例包括：

- **IDataProvider** 接口上的 **ReAssociate** 方法
- **ITableDefinition** 接口上可设置的 **DataSourceDefinition** 属性

实现序列化

类通过添加 **[Serializable]** 属性标记或实现 **System.Runtime.Serialization.ISerializable** 接口进行序列化。我们建议同时采用这两个方式。**ISerializable.GetObjectData** 方法必须实现，而且必须为公共方法：

```
public void GetObjectData(SerializationInfo info, StreamingContext context)
```

对于从抽象基类派生的类，此签名还应该包含覆盖键，而且实现的第一行代码应该委托给基类，以便对所管理的组件进行序列化。例如：

```
base.GetObjectData(info, context);
```

接口中剩下的可序列化成员将通过向 **AddValue** 方法提供字符串键和成员值序列化为名为 **info** 的 **SerializationInfo** 参数。例如：

```
info.AddValue("TableName", _tableName);
```

实现反序列化

为了支持反序列化，需要提供受保护的反序列化构造函数，其参数与上面的 **GetObjectData** 上的参数匹配。例如：

```
protected COTWTableDefinition(SerializationInfo info, StreamingContext ctxt)
```

对于从抽象基类派生的类，此构造函数应该委托给基类，以便对所管理组件进行反序列化。例如：

```
protected COTWTableDefinition(SerializationInfo info, StreamingContext ctxt) : base(info, ctxt)
```

在此构造函数内，类实例通过使用名为 **info** 的 **SerializationInfo** 参数上提供的“**get**”方法检索值，从而分配其成员变量。例如，要对上面的 **GetObjectData** 示例中序列化的 **TableName** 值进行反序列化，相应的代码可能与以下所示类似：

```
_tableName = info.GetString("TableName");
```

对于属于序列化的方法，根据实现类是否密封（至少为决定因素之一），有不同的关于覆盖支持、方法属性等的建议。通过使用 **FxCop** 之类的代码分析工具，可以提供非常有价值的辅助，从而提供适当的建议。

有关其他信息和示例，请参阅《**MapXtreme** 开发人员参考》（联机帮助）、本附录中提供的“**SpatialLite** 示例数据提供方”和《**MapXtreme** 代码交流》中的其他数据提供方示例。

身份验证

很多数据提供方在打开数据源或表是要求进行身份验证，其中的身份验证详细信息安全保存，不能从可公开访问的共享工作空间文件获得。

为了支持 `OpenDataSource` 和 `OpenTable` 处理的运行时身份验证，`MapXtreme` 可扩展数据提供方模型提供了名为 `IDataProviderCallback` 的新接口。此接口支持用户定义的回调方法，供可扩展数据提供方用于解析数据源和表定义，但这些定义不足以用于打开数据源或表，因而起到保护作用。

对于最简单的实现，自定义数据提供方的客户端代码将包含一个类，该类实现 `IDataProviderCallback` 接口，并提供 `IDataSourceDefinition` 和/或 `ITableDefinition` 回调方法的实现，可供直接打开数据源或表。

不过，对于大部分用户，将有必要加载包含 `IDataProviderCallback` 初始化的独立程序集且将其此程序集作为 `MapXtreme` 会话初始化的一部分加载。请参阅 [ISessionEventHandlers](#)。当您在加载默认工作空间时如果尝试解析数据源或表定义，将要求这样做。例如，基于 Web 的应用程序将要求为回调提供会话初始化支持。

如何使用数据提供方回调？

回调方法如何调用由数据提供方实现处理；`MapXtreme` 管理回调方法集合并将其通过定义的接口传递给数据提供方。它并不调用任何 `IDataProviderCallback`。

`IDataProviderCallback` 实例由 `MapXtreme` Session Catalog 上的 `DataProviderCallbacksCollection` 管理。此回调集合将回调对象集中为一个 `System.Type`，并且每个类型关联支持多个回调。可以通过多种方式管理 `DataProviderCallbacksCollection`：会话初始化期间（请参阅 [SessionEventHandler/DataProvider 加载章节](#)）、应用程序初始化期间，以及最为简单的方式，在紧靠从客户端应用程序进行显式 `OpenTable` 调用前的位置。

`MapXtreme` 查询 Session Catalog 获得与 `OpenDataSource` 或 `OpenTable` 方法关联的回调，并将对集合的引用提供给 `DataProvider`。`IDataProvider` 接口支持对 `IEnumerator` 的引用，后者用于在 `OpenDataSource` 和 `OpenTable` 操作期间遍历 `IDataProviderCallback`。如果可以通过修改输入 `DataSourceDefinition` 或 `TableDefinition` 来处理故障，则 `DataProvider` 类可以使用 `IEnumerator` 引用的任何回调来实现解决 `OpenDataSource` 或 `OpenTable` 故障所需的相应逻辑。或者，`DataProvider` 可以实现用于完成设计不足的定义对象的已知策略。

回调的一个典型示例是，在桌面应用程序环境中，`OpenDataSource` 回调向用户显示对话框，要求提供对数据提供方有效的用户名/密码组合，从而解决凭据身份验证错误。

实现数据提供方回调

`IDataProviderCallback` 接口指定用于修改 `IDataSourceDefinition` 和 `ITableDefinition` 数据提供方定义类型的数据的方法签名。实现 `IDataProviderCallback` 的类必须提供相应的方法实现，以便实时更新定义对象，从而满足 `OpenDataSource/OpenTable` 要求。

此外，还可以在回调方法上指定 `IDataProviderCallbackInfo` 接口，以向回调提供其他的运行时状态信息，方便确定正确的操作。`IDataProviderCallbackInfo` 仅是标记接口，并不指定向回调提供的信息的性质。典型的实现可以包括运行时异常或状态枚举属性。

`MapInfo.Data.Provider` 命名空间包含可供数据提供方和/或客户端应用程序使用的规范实现。`DataProviderCallback` 类实现基于委托的回调机制，允许客户端代码在可扩展数据提供方程序集外分配回调方法。`DataProviderCallbackExceptionInfo` 类将 `IDataProviderCallbackInfo` 作为 `System.Exception` 引用的承载者实现，而 `DataProviderCallbackCollection` 实现 `IDataProviderCallback` 的 `ICollection`。

以下是如何实现凭据解析回调方案的示例。

```
// Create a method to handle the OpenDataSource InvalidCredentials state
// Implements IDataProviderCallback.Callback signature
public static IDatasourceDefinition InvalidCredentialsCb(
IDatasourceDefinition dsd, IDataProviderCallbackInfo info)
{
    // Implementation details are specific to the Data Provider,
    // specifically, to the appropriate IDatasourceDefinition implementation
    DataProviderCallbackExceptionInfo pcinfo
        = info as DataProviderCallbackExceptionInfo;
    if (pcinfo != null)
    {
        if (pcinfo.Exception == /* invalid credentials */)
        {
            // Example: present user dialog
            string newpwd = GetPasswordFromUser();

            // Example: create new EDP implementation specific DataSourceDefinition
            // with updated password credentials
            return new EDPDataSourceDefinition(newpwd);
        }
    }
}

// MXT/EDP client code
{
    // Create a callback harness and assign the callback method
    DataProviderCallback cb = new DataProviderCallback();
    cb.DataSourceDefinitionCallback = InvalidCredentialsCb;

    // Associate the IDataProviderCallback with the EDP definition data type
    Session.Current.Catalog.DataProviderCallbacksCollection.AddProviderCallback(
        typeof(EDPDataSourceDefinition), cb);

    // Create an initially invalid Data Source Definition
    EDPDataSourceDefinition dsd = new
        EDPDataSourceDefinition("badpassword");

    // open table
```

```
EDPTableDefinition tableDef = new EDPTableDefinition (dsd);
Table t = Session.Current.Catalog.OpenTable(tableDef, "TESTTABLE");
}
```

Implementing `IDataProviderCallback` usage in the Data Provider

`IDataProvider` 上的 `OpenDataSource` 和 `OpenTable` 方法包括 `IDataProviderCallback` 实例的枚举符。此枚举符由 `Catalog` 自动提供。

`DataProvider` 实现负责迭代，并在有必要尝试完成 `OpenDataSource|OpenTable` operation 操作时调用回调。

示例 `DataProvider` 实现可能会继续加入以下代码行，从而确保构造了有效的开放数据源：

```
public override IDatasource OpenDataSource(
IDatasourceDefinition definition,
CustomProperties customProperties,
IEnumerator<IDataProviderCallback> callbacks)
{
    EDPDataSourceDefinition dsDef = definition as EDPDataSourceDefinition;
    if (dsDef == null)
    {
        // invalid definition
        // throw exception
    }

    EDPDataSourceDefinition tmpDsDef = dsDef;
    EDPDataSource ds = null;
    bool needDS = true;
    while (needDS)
    {
        needDS = false;
        DataProviderCallbackExceptionInfo cbinfo = null;
        try
        {
            // Data Source ctor will fail if data source definition is invalid,
            // e.g., invalid credentials
            ds = new EDPDataSource (tmpDsDef, customProperties);
        }
        catch (Exception e)
        {
            cbinfo = new DataProviderCallbackExceptionInfo(e);
        }
        if (ds == null
        && callbacks != null)
        {
            while (!needDS
            && callbacks.MoveNext())
            {
                // callback will return null if it is unable to modify the
                // the definition appropriate for a retry attempt
                // Example: User selected cancel from a credential input dialog
            }
        }
    }
}
```

```
        tmpDsDef = callbacks.Current.Callback(dsDef, cbinfo) as
EDPDataSourceDefinition;
        if (tmpDsDef != null)
{
    needDS = true;
}
} // while (!needDS and callbacks.MoveNext())

} // if (ds == null && callbacks != null)

if (!needDS
&& cbinfo != null)
{
throw cbinfo.Exception;
}

} // while(needDS)

return ds;
}
```

线程安全

MapXtreme 是线程安全的，即不同的线程可以并发访问不同的会话、目录、表、地图等，而不会产生任何意外的副作用。**MapXtreme** 对象不支持多线程，即地图、表、目录等的相同示例不能同时从多个线程进行访问。只要可能会在 Web 环境中部署数据提供方，则数据提供方实现必须至少遵循此模型。

通常，只要不依赖于任何全局变量、单态对象等，代码就应该是线程安全的。另外，还不应依赖于线程本地存储的使用，因为 **ASP.NET** 可能会实际导致执行您请求的线程发生变更。如果您的数据提供方遵循这些指南，则不应在整个代码中要求使用太多（如果有）的同步锁定，因为这可能会方案解决方案的可扩展性。不过，您访问的实际数据可能会暗示同步无法避免或是首选方式。例如，如果您的数据提供方公开来自 **Excel** 电子表格的数据，则在不同的线程（或完全不同的进程）中打开进行读取操作时，电子表格可能锁定，无法进行编辑操作。对于这些情况，务必仔细考虑如何设计数据提供方的体系结构。光标、图元访问器和修改处理器全部都意味着在需要数据时要对数据进行某种类型的锁定。不过，如果没有活动光标、访问器或修改处理器，底层数据文件（如果有）就应该保持开放状态。

IDataProvider 是一个例外，我们建议将其作为单态对象实现。此接口旨在作为工厂接口使用，完全不应该包含任何状态（即除了静态 **INSTANCE** 属性外不应该有任何类成员变量）。定义对象必须引用其数据提供方，**MapXtreme** 才能够对其进行调用来打开数据源或表。对于此机制，尽管非常适合采用单态方式，但需要一个可实例化的类。

您可能倾向于缓存连接，甚至可能缓存数据源或表对象，并在调用 **OpenTable** 或 **OpenDataSource** 时进行重用。我们强烈建议不要这样做。**MapXtreme** 在 **Catalog** 上提供了数据源和表的集合，将依赖于您在 **ITableDefinition** 和 **IDataSourceDefinition** 类中提供的 **Equals** 逻辑来查找和重用正确的实例。这样将防止出现线程交叉、多线程的情况。您的底层数据访问技术（如 **ADO.NET** 数据提供方）可能会缓存和重用数据库连接。由于能够在多线程环境中确保正确的行为，因此这是可以接受的体系结构。

从 MapXtreme 应用程序 进行打印

本附录将指导您从 MapXtreme 开发项目中打印出最好的地图图像。首先会介绍打印功能概述和一些有用的技巧和方法，然后帮助您解决在使用 MapXtreme 应用程序打印时出现的一些问题。

在本附录中：

◆ 概述	497
◆ 了解 MapXtreme 中的打印选项	497
◆ 在应用程序中实现打印功能	501
◆ 常规打印技巧与方法	502
◆ 已知打印问题的解决方案	506

概述

从 MapXtreme 开发的应用程序通常可以直接打印。但是随着打印设备多样性持续扩展，将会产生特定于某些设备的问题。我们为开发人员提供了多种功能以便自定义打印过程。这些选项设计为可以根据将要打印的地图及使用的设备来自定义打印。这些不同设置用于满足多种打印机和绘图仪的需要和它们之间的细微差别。

MapXtreme 支持：

- 打印地图 - 使用 `MapPrintDocument` 类。
- 打印图例 - 使用 `LegendPrintDocument` 类。
- 直接打印到设备或通过使用增强元文件 (Enhanced Metafile, EMF) 进行打印。
- 打印不同大小的地图
- 打印具备半透明样式颜色和图层的地图

有关信息，请参阅 [GDI+ 半透明度和抗锯齿](#)。

- 打印半透明光栅图像

只有启用了 `EnableTranslucency` 才能打印半透明光栅图像。

MapXtreme 不支持：

- 使用打印选项对话框。
- 打印布局。

打印地图中的图例时布局是非常重要的。有关如何处理该限制，请参阅示例[打印地图中的图例](#)。

- 打印多页面地图。

如果地图的幅面为多个页面，则只打印第一页。

- 通过编程方式打印到文件。

需要使用 `Microsoft.NET 2.0 Framework System.Printing.PrinterSettings` 类来实现此功能。

在 MapXtreme 应用程序中处理打印的类是从 `Microsoft.NET 2.0 Framework System.Drawing.PrintDocument` 类派生而来并继承了该功能。设备输出控制由 `Microsoft.NET 2.0 Framework System.Printing.PrinterSettings` 和 `System.Printing.PageSettings` 类管理。

`MapInfo.Printing.MapPrintDocument` 类专门用于打印地图，`MapInfo.Printing.LegendPrintDocument` 则专门用于打印图例。

了解 MapXtreme 中的打印选项

从 MapXtreme 应用程序打印地图或图例时，可以向用户提供以下打印选项。为了使用户可以控制这些设置，需要将此功能构建在应用程序的打印对话框中。如果不是特定地允许用户更改这些选项，则使用默认设置。也可以通过编程方式调节这些设置。

打印大小

MapXtreme 提供这些选项用于打印不同大小的地图。要改变地图打印的大小，需要将 `MapPrintDocument.MapPrintSize` 属性设置为 `MapPrintSize` 枚举中的一个值。这些值如下所示：

适合于页面

该选项为默认选项，打印的地图保持其纵横比，但缩放至适合于页面。

```
MapPrintDocument.MapPrintSize=MapPrintSize.FitPage
```

填充页面

使用此选项打印的地图填充整个页面。此方法不保持地图原来的纵横比，打印的地图可能会变形。

```
MapPrintDocument.MapPrintSize=MapPrintSize.FillPage
```

当前地图大小

此选项以地图的原始大小打印地图。此方法可能会正合适，但是如果地图比所在页面大，则打印时会丢失超出页面的部分。

```
MapPrintDocument.MapPrintSize=MapPrintSize.MapSize
```



以下属性通用于 `MapInfo.Printing.MapPrintDocument` 和 `MapInfo.Printing.LegendPrintDocument` 类。

特殊透明光栅处理

用于将应用程序设置为内部管理透明像素显示和光栅图像的打印。在屏幕上，透明图像被渲染为使用光栅操作 (ROP) 来处理透明像素。打印时此方法可能起作用，也可能不起作用，因为 MapXtreme 使用不同的方法来确定打印的透明方法。必须确定特殊的打印驱动程序是否可以正确处理 ROP，并相应将其设置为 `true` 或 `false`。大多数的打印情况下，此值应设置为 `true`。当绘制到屏幕上时，此值通常设置为 `false`。使用 `DrawingAttributes.SpecialTransparentRasterHandling` 来设置此值。此值设置为 `true`（默认）使应用程序可以内部管理打印过程。



打印矢量图层时此设置不起作用。

特殊透明矢量处理

用于将应用程序设置为打印时内部管理透明矢量填充图案。使用 `DrawingAttributes.SpecialTransparentVectorHandling` 来启用或禁用此功能。将此功能设置为 `true` 使 MapXtreme 在打印透明填充图案或透明位图符号时执行特殊的处理。此设置通常用于打印。将此功能设置为 `false` 让打印设备来处理如何打印透明填充图案和透明位图符号。请参阅[使用填充图案时需要速度吗？](#)详细了解何种填充图案是位图，何种填充图案是矢量图。

尽可能地以真彩色显示光栅

某些打印机不支持 24 位（真彩色）图像。MapXtreme 具有内部功能可以对此进行调节。使用 `DrawingAttributes.TrueColorRaster` 来启用或禁用此功能。

将此功能设置为 `true`（默认）来使用 24 位（真彩色）打印光栅和网格图像。如果图像是 24 位，同时打印机支持 256 色（8 位）以上，就可能实现此功能。如果打印机不支持 24 位图像，可将此属性设置为 `false`，仅使用 256 色用 `DitherMethod` 中指定的抖动来渲染光栅图像。将此值设置为 `false` 打印 24 位图像时，打印出来的图像将会严重恶化。将此属性设置为 `false` 时，同时需要设置抖动方法（参见下文）。

GDI+ 半透明度和抗锯齿

通过 MapXtreme 中的 GDI+ 渲染可创建半透明线、标注和图层，还可应用抗锯齿，以便在以较低分辨率呈现高清晰度转换时对线、曲线的锯齿状边缘和填充区域的边缘进行平滑处理。

用 `MapPrintDocument` 或 `LegendPrintDocument` 打印地图或图例时，会分别用 `Map.DrawingAttributes` 或 `Legend.DrawingAttributes` 中的值，自动设置 `DrawingAttributes.EnableTranslucency` 和 `DrawingAttributes.SmoothingMode` 这两个属性的值。初始化后，用户可以用自己的值设置覆盖这些属性。借此可在不必明确设置这两个显示选项的情况下，将其在地图或图例中传递。

EnableTranslucency: 用于获取或设置是否在屏幕、打印机或文件导出中绘制地图时向样式颜色和/或图层中添加半透明值。此属性为 `false` 时会忽略样式（如 `Color.A`）和图层（如 `MapLayer.Alpha`）中的半透明度值。使用值 255 替代。该属性对光栅半透明度没有影响（例如，`RasterStyle.Alpha`）；但只有启用了 `EnableTranslucency` 才能打印半透明光栅图像。这对屏幕显示或导出没有必要。如果将该属性设置为 `false`，那么 `DrawingAttributes.SmoothingMode` 属性将会自动设置为 `SmoothingMode.None`。换句话说，不启用半透明度，就不能使用抗锯齿。

SmoothingMode: 获取或设置地图的渲染质量。此属性用于指定 `MapInfo.Mapping.SmoothingMode` 枚举成员。默认值为 `SmoothingMode.None`。平滑模式指定直线、曲线和填充区域边界是否使用平滑（也称为抗锯齿）。如果将该属性设置为 `SmoothingMode.AntiAlias`，那么 `MapInfo.Mapping.DrawingAttributes.EnableTranslucency` 属性将会自动设置为 `true`。换句话说，当使用抗锯齿时始终允许半透明度。

如果将 `EnableTranslucency` 设置为 `false`，则会将 `SmoothingMode` 自动设置为 `None`（如果尚未对其作此设置）。如果将 `SmoothingMode` 设置为 `AntiAlias`，则会将 `EnableTranslucency` 自动设置为 `true`（如果尚未对其作此设置）。`EnableTranslucency` 为 `true` 时，可在不失去半透明度效果的情况下在 `SmoothingMode` 的 `None` 设置和 `AntiAlias` 设置之间进行切换。

MapStyleControl 类

`MapStyleControl` (`MapInfo.Windows.Controls` 命名空间) 可在桌面 `LayerControl` 中的选项卡上显示，允许用户设置地图样式和渲染选项（如抗锯齿和半透明）。

抖动方法

抖动是一种电脑混合像素的技术，当颜色深度减小时可以保持图像的外观。将 24 位图像转换为 256 色时选择一种抖动方法。

- **半色调抖动**计算图像中高对比度元素之间颜色的一系列半色调的差异，以此来创建颜色的平滑过渡。此选项在显示、打印和导出选项中默认选择。
- **误差扩散抖动**计算对比颜色之间的中间色，并将周围像素逐渐均匀混合为中间色。

通过使用 `DrawingAttributes.RasterDitherMethod` 属性设置此值。可以将此值设置为 `DitherMethod` 枚举中的一个值。

示例：

```
mapPrintDocument.DrawingAttributes.RasterDitherMethod =  
MapInfo.Mapping.DitherMethod.HalfTone;
```

特殊多边形孔处理

MapXtreme 应用程序可以使用多个单独的块并将它们混合在一起绘制复杂的多边形 (Polygon) 对象。当多边形带有孔或岛状物打印时，这些块可能无法很好地匹配，可能会叠加或留有缝隙。有一些内部编程可以调节此错误行为，以确保这些多边形正常显示和打印。要利用此功能，需要将 `DrawingAttributes.SpecialPolygonHoleHandling` 属性的值设置为 `true`。当绘制到屏幕上时，此值应设置为 `false`。

缩放图案

此设置即可以打印不透明位图填充图案，正如屏幕上所看到的一样，还允许打印机驱动程序额外对渲染该填充图案进行控制。用 `DrawingAttributes.ScaleBitmapPatterns` 设置此值。如果此值为 `true`，则位图填充图案将缩放到可以补偿打印机和屏幕分辨率的差异。此值为 `False` 则不会缩放图案，而是依靠打印机驱动程序来缩放该填充图案。如果打印机驱动程序可以缩放填充图案，则应当将此值设置为 `false`。如果打印机驱动程序无法缩放填充图案，则应当将此值设置为 `true` 以便防止打印输出显示得太小，且显示为接近实心填充图案。请参阅[使用填充图案时需要速度吗？](#)详细了解何种填充图案是位图，何种填充图案是矢量图。

如果此属性设置为 `true`，则打印预览和打印出来的文档看起来可能会不同，因为打印机驱动程序缩放了图案，而显示时并没有缩放图案。即使显示不正常，也可以正常打印。



如果填充图案有背景颜色，则认为图案是不透明的，而没有背景颜色的图案认为是透明填充图案，无论其属性值是什么，都始终被缩放。

直接打印到设备

使用此选项可以将图像直接打印到打印机。用 `PrintMethod.Direct` 设置此值。

使用增强元文件 (EMF) 打印

使用此选项生成图像的增强元文件，然后将其发送到打印机。此选项利用当前打印机技术来缩小假脱机大小，并且可以在不牺牲质量的前提下更快地打印文件。用 `PrintMethod.Emf` 设置此值。

在应用程序中实现打印功能

MapXtreme 中的打印命名空间包含一组类，可以帮助您打印地图。使用这些类可以向任何连接的打印设备上处理地图的基本打印。

使用 **MapPrinting** 类来访问对话框和打印机控制对话框。为应用程序分配单独的 **MapPrinting** 对象，并使用相同的例子来用于打印/打印预览/页面设置，这一点非常重要。

设置 **MapPrinting** 并分配地图：

```
this.mapPrinting = new MapPrinting();
this.mapPrinting.Map = this.mapControl1.Map;
```

以下是用于打印地图的示例代码：

```
this.mapPrinting.ShowDialog = true;
this.mapPrinting.Print();
```

以下是用于显示地图打印预览的示例代码：

```
this.mapPrinting.PrintPreview();
```

以下是用于显示“页面设置”对话框的示例代码：

```
this.mapPrinting.PageSettingsDialog();
```

可以通过打印文档访问打印机设置，打印文档是主要打印类的一部分。因为要向支持打印需要的派生类中添加设置和选项，所以必须从地图打印对象中将该类去除，再重新将其类型转换到派生类中。

```
mapPrinting = new MapPrinting();
mapPrinting.Map = mapControl1.Map;
MapPrintDocument mapPrintDocument = mapPrinting.PrintDocument
as MapPrintDocument;
if (mapPrintDocument != null) {
    // here are some examples of how to set print options
    // Set these based on your needs
    mapPrintDocument.DrawingAttributes.SpecialTransparentRasterHandling
= true;
    mapPrintDocument.PrintMethod = PrintMethod.Direct,
    // and set other properties of mapPrintDocument
}
```

我们不提供可用于更改地图打印选项的 UI。这些对话框仅用于用户更改系统打印机设置。如果想要提供给用户附加选项，则必须提供自己的 UI 来设置 **MapPrintDocument**（例如 **DrawingAttributes**、**PrintMethod** 和 **PrintSize**）的属性。

在打印文档中还可以访问打印进程，还可以添加打印事件发生时的回调。例如，您也许想要向每个打印的页面上添加徽标或其它图形。

C# 示例

```
mapPrinting = new MapPrinting();
mapPrinting.Map = mapControl1.Map;
mapPrinting.PrintDocument.PrintPage += new
PrintPageEventHandler(mapPrintDocument1_PrintPage);

private void mapPrintDocument1_PrintPage(object sender,
PrintPageEventArgs e)
{
    // add customization for each page (ie; title, page #, etc.)
    Graphics g = e.Graphics;
}
```

VB 示例

```
mapPrinting = New MapPrinting()
mapPrinting.Map = mapControl1.Map
AddHandler mapPrinting.PrintDocument.PrintPage, AddressOf
mapPrintDocument1_PrintPage

Private Sub mapPrintDocument1_PrintPage(ByVal sender As Object, ByVal e As
PrintPageEventArgs)
    ' add customization for each page (ie; title, page #, etc.)
    Dim g As Graphics = e.Graphics
End Sub
```

使用此方法可以访问图形对象，并可以使用任何可用图形程序来打印附加图形或文本。

常规打印技巧与方法

i **MapInfo** 不会推荐某种打印机/绘图仪优于其它种类的打印机/绘图仪，也不会特别证明某种打印机可以永久正常运行。有太多种因素可以影响到输出，从而我们无法做出推荐。

用户在打印时碰到了一些特定的常见问题并提供了解决方案，以下技巧和建议就是根据这些信息而得出的。

解决打印故障的良好的第一步是要确保您已经下载和安装了该 **MapXtreme** 版本的补丁（如果有补丁），并且要确保正使用的打印机驱动程序是您的打印机/操作系统驱动程序的最新版本。此文档中对一些异常已作出说明。

如何重载打印页面事件

.NET 打印 API 提供了一种重载打印页面事件的机制。因此可以添加对每个打印页面的自定义设置。例如，可以添加标题、页码、徽标等。

需要实现 **PrintPageEventHandler** 才能完成此功能。示例：

```
this.mapPrinting = new MapPrinting();
this.mapPrinting.Map = this.mapControl1.Map;
this.mapPrinting.PrintDocument.PrintPage += new
System.Drawing.Printing.PrintPageEventHandler(this.mapPrintDocument1_Print
tPage);

private void mapPrintDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    // TODO - add your code here to alter the printed page
}
```

打印地图中的图例

正如本节的开始部分所述，**MapXtreme** 不支持布局的打印。如果想要打印地图中的图例，则需要将该图例作为修饰添加到地图中。下面是一些代码示例演示此操作如何执行：

```
//create cartographic legend
MapInfo.Mapping.Legends.Legend legend =
mapControl1.Map.Legends.CreateLegend(new System.Drawing.Size(5, 5));
legend.Border = true;
MapInfo.Mapping.LayerType[] normalLyr = new MapInfo.Mapping.LayerType[1];
normalLyr[0] = MapInfo.Mapping.LayerType.Normal;
MapInfo.Mapping.IMapLayerFilter filter =
MapInfo.Mapping.MapLayerFilterFactory.FilterByLayerType(normalLyr);
MapInfo.Mapping.Legends.LegendFrame frame;

foreach(MapInfo.Mapping.FeatureLayer ftrLayer in
mapControl2.Map.Layers.GetMapLayerEnumerator(filter))
{
    frame =
MapInfo.Mapping.Legends.LegendFrameFactory.CreateCartographicLegendFrame(
ftrLayer);
    legend.Frames.Append(frame);
}

//set legend location on the map
System.Drawing.Point pt = new System.Drawing.Point(220, 200);
pt.X = mapControl2.Size.Width - legend.Size.Width;
pt.Y = mapControl2.Size.Height - legend.Size.Height;
legend.Location = pt;
```

```
//append legend as map adornment  
mapControl1.Map.Adornments.Append(legend);
```

```
mapPrinting.print();
```

旧的驱动程序能用，新的驱动程序不能用

有疑问时，如果旧的驱动程序能用而新的不能用，那么请使用旧的驱动程序。

有关 HP 755 驱动程序的建议

如果使用此型号的绘图仪时打印有困难，请尝试用 HP 650C (C2859B) 的绘图仪驱动程序来代替。多数情况下，如果使用某种打印机或绘图仪型号时有问题，还会有另一种相似的型号，您可以用相似型号的驱动程序来代替当前的驱动程序，效果会非常好。例如，您可以将 HP8550 color LaserJet 彩色激光打印机驱动程序用作 HP 8500 DN Color LaserJet 彩色激光打印机的驱动程序。

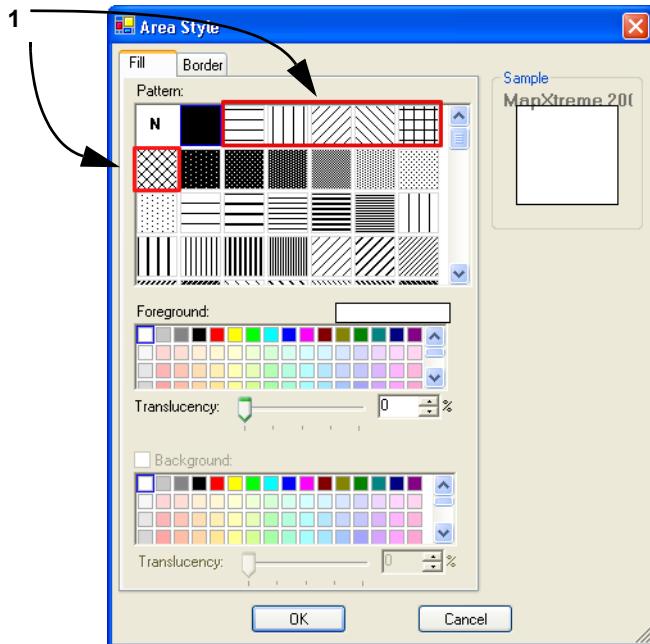
有效缩放图案的建议

打印时，了解您正在使用的打印机驱动程序的类型。许多 PCL6 和某些 HPGL 驱动程序可以进行填充图案的缩放，并提供此功能的控制。关闭其缩放可能会造成打印结果和监视器上显示的有所不同。建议尝试关闭驱动程序的缩放选项而先试用我们提供的缩放功能，因为我们已经增强了更好地打印地图和图例的方法。打开我们的图案缩放选项，启用应用程序中的“缩放图案”（请参阅[缩放图案](#)）。再将缩放功能关闭，看一看哪一种效果更好。测试显示使用我们的缩放功能所产生的颜色输出与屏幕上的显示更接近。

我们发现，如果使用 LanguageLevel 2 或 3 打印到 PostScript 驱动程序，某些 Microsoft 驱动程序不支持图案缩放。那么我们的缩放方法可能就起不到作用了。Microsoft 建议将 PostScript 驱动程序的语言级别重新设置到 LanguageLevel 1 来除去此限制。此情况下出现过异常。在 Windows 2000 和 Windows NT 上，使用我们的缩放选项，一些 HP Laser Jet 激光打印机和 Color Laser Jet 彩色激光打印机的 PostScript 驱动程序可以正常打印。

使用填充图案时需要速度吗？

请注意 **AreaStyle** 对话框中的前六项填充图案（实心填充之后）是 Windows 标准，有助于更快速地打印。这些填充图案是基于矢量的。MapXtreme 附带的其它图案是位图。选择填充图案时可能需要考虑这些。



1 使用这些填充图案以实现快速打印。

速度仍然不理想吗？

如果要提高打印机速度，而打印机有“快速”、“正常”和“最佳”三个打印质量选项，则建议选择“快速”。但是这样也会降低输出的分辨率。

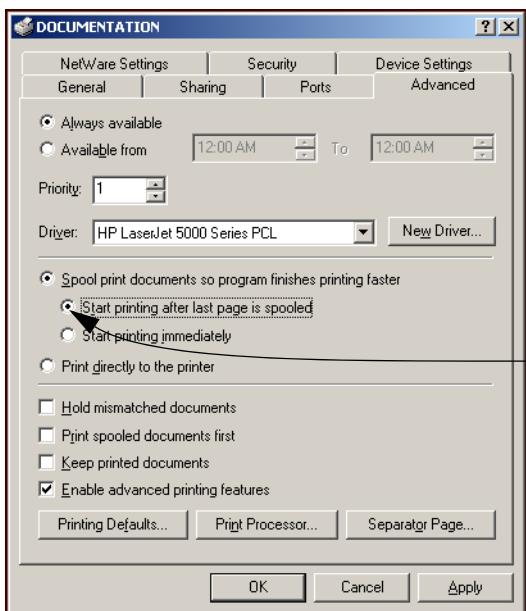
磁盘空间不足吗？

确保有足够的临时磁盘空间，特别是在使用“使用增强元文件打印”选项时。系统会尝试在磁盘上本地创建一个带图层的位图。

全局打印？本地假脱机！

尝试在本地假脱机¹打印作业，而不是在绘图仪上。这样计算机可以将输出光栅化，而不是由打印机来做此项工作，这样会更有效。

1. 假脱机是存储数据的过程，在内存中或文件中组成将被打印的文档，直到打印机准备好对其进行处理。



1 单击假脱机打印文档和最后一页假脱机后开始打印按钮。

1. 要设置本地假脱机打印作业，请选取开始 > 设置 > 控制面板 > 打印机。
2. 右键单击打印机，然后选择菜单中的“属性”，以显示打印机的属性。
3. 单击“高级”选项卡，显示高级属性选项。

(i) 如果您没有计算机的管理员权限，可能无法使用假脱机选项。如果想要进行更改，请与 IT 部门联系以获取他们的支持。

4. 单击确定保存所作更改。

已知打印问题的解决方案

有许多种因素影响应用程序的打印和绘图，MapXtreme 应用程序也不例外。MapXtreme 不提供打印机驱动程序，而是使用当前 Windows 操作系统中现有的驱动程序。本节介绍用户及合作伙伴没有涉及的特定的打印机/绘图仪问题。我们提供这些信息的目的是支持您，使得您可以支持您的用户。

平台独立问题

如果没有特别注明，任何操作系统或硬件/软件都会发生平台独立问题。在下文中查找您正要解决的问题，并参阅解决方案。

阴影线图案打印为黑色多边形

打印机/绘图仪：HP Designjet 系列绘图仪

问题：打印带阴影线图案的地图时，一些区域显示为黑色实心。

解决方案：依据 Hewlett Packard 支持部门，新的 HP 打印机驱动程序 (4.63) 可以正确处理非 Windows 标准的阴影图案。我们建议您下载并安装新的 HP 4.63 驱动程序来解决此问题。

打印时阴影图案压缩了

操作系统：WindowsNT 4.0/2000

打印机/绘图仪：HP LaserJet 系列，用 HP LaserJet 4050 系列 PCL 6 驱动程序测试

问题：使用 MapInfo 阴影图案（非前六种图案 AreaStyle 对话框的其中一种）时，可能会注意到该图案大大地压缩了。这是因为这些图案是位图，是以较高的打印机分辨率绘制而成的。标准 Windows 图案不会出现此问题。打印机分辨率会影响压缩量。

解决方案：一些打印机驱动程序具有一个新的输出设置称为“缩放图案 (WYSIWYG)”，使图案可以按正确的比例缩放。在测试的 HP 打印机中，可通过选择打印首选项 > 打印质量 > 详细情况选项来使用此选项。

用户应当关闭驱动程序的缩放功能或关闭我们的缩放功能，因为两个缩放同时工作会将图案填充两次。尝试两种缩放类型，选出更好用的一种。

特定平台问题

如果没有特别注明，这些问题出现在特殊的操作系统和/或硬件。这些问题根据操作系统分组。

打印光栅时，如果使用 HP 500, 800, 5000 DesignJet 绘图仪，每 2-4 厘米将重复打印输出
有两种建议可用于解决此问题。首先，尝试从驱动程序级别或打印机级别使打印机本地假脱机。



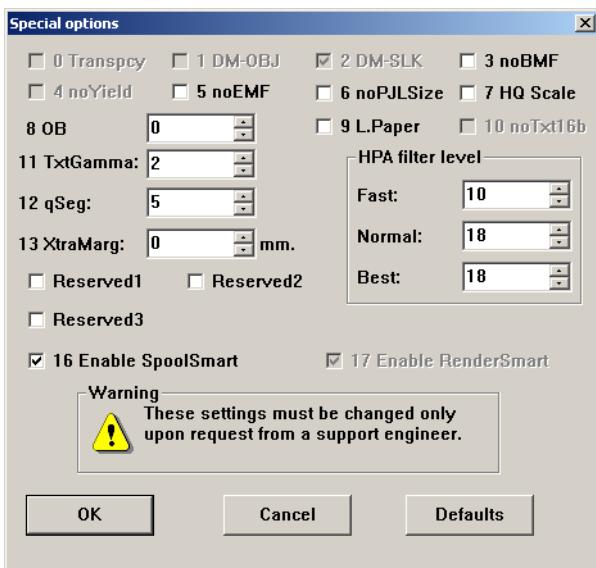
必须具备打印机和打印机驱动程序的读/写权限才能解决此问题。

要解决此问题，可以在驱动程序属性中更改一个高级隐藏设置。

要避免在打印矢量覆盖的光栅图时打印机每 2-4 厘米 重复自己：

1. 在“控制面板”中找到打印机驱动程序的属性。要获取此设置，右键单击打印机图标并选取“属性”。
2. 单击“高级”选项卡。选择以下选项：
 - 选择避免内存不足选项。
 - 如果使用 Windows 2000 或 XP 操作系统，则请选择缩放因子为 100%。
3. 单击此对话框中的打印默认值按钮。
4. 在此对话框中，选择关于按钮。关于<驱动程序名称>对话框显示。

5. 按下键盘上的 **F8** 键，单击确定按钮。“Special options”对话框显示。



6. 清除 Enable SpoolSmart 复选框，然后单击确定：

(i) 修改此驱动程序选项框中的设置时，一定要留意原始设置，以备在需要时改回原始设置。这些是专门为 HP 支持工程师设计的，普通公众一般不使用。

可以将这些设置保存为“快速设置”，以确保每次打印光栅文件时都起作用。

1. 浏览至 HP 打印机驱动程序，访问驱动程序首选项。使用这些方法中适合您的操作系统的一种：
 - 使用 **Windows NT 4.0** — 从 Windows 桌面选择开始 > 设置 > 打印机。然后，右键单击合适的绘图仪驱动程序图标，并选取“文档默认值”。
 - 使用 **Windows 2000/XP (经典视图)** — 从 Windows 桌面选择开始 > 设置 > 打印机。然后，右键单击合适的绘图仪驱动程序图标，并选取“打印首选项”。
2. 在隐藏菜单中按照您所想要的方式设置首选项。
3. 在“快速设置”框中，键入所选设置的名称（例如，“光栅打印设置”），并单击保存。当前所有的驱动程序设置（纸张类型、隐藏菜单设置等）都保存在“快速设置”名称下。打印机驱动程序会记住这些设置，可以用于将来的打印作业。

要在稍后以相同设置打印光栅地图：

1. 选取“打印”命令来显示“打印”对话框。
2. 选取属性按钮。“属性”对话框显示。
3. 确保输入的名称（如示例中的“光栅打印设置”）显示在“快速设置”下拉列表中，然后单击确定。“打印”对话框重新显示。

4. 单击确定以打印。

i 如果“快速设置”中不需要此特殊设置，请记住要将此设置改回到默认机器设置。

不能使用 HP Designjet 打印机打印（驱动程序 5.31 或 5.32）。

当尝试使用这些驱动程序时会发生以下一种情况：

- 如果选择最高性能选项，则会显示内存出错信息，打印机仅打印部分图像。
- 如果选择避免内存不足选项，则计算机会崩溃，并显示“硬盘空间不足”和“虚拟内存不足”信息。

此问题的解决方案与前述问题的解决方案相似。“特殊选项”对话框为同一个对话框。

1. 编程设置以下打印选项：

```
MapPrintDocument mapPrintDocument = this.mapPrinting.PrintDocument as  
MapPrintDocument;  
mapPrintDocument.PrintMethod = PrintMethod.Direct;  
mapPrintDocument.DrawingAttributes.SpecialTransparentVectorHandling =  
true;  
mapPrintDocument.DrawingAttributes.SpecialTransparentRasterHandling =  
false;  
mapPrintDocument.DrawingAttributes.TrueColorRaster = true;.
```

2. 从“控制面板”选择“打印机”或“打印机和传真”，找出您用于打印的打印机。
3. 右键单击该打印机并选择“属性”选项。“属性”对话框显示。
4. 单击关于按钮来显示关于<打印机驱动程序>对话框。
5. 按下键盘上的 F8 键，单击确定按钮。“特殊选项”对话框显示。
6. 取消选取“启用 SpoolSmart”复选框。

F

样式查找

本附录包含支持样式的查找表，包括填充图案、直线样式、矢量符号和自定义位图符号。有关样式说明和如何使用的详细信息，请参阅[第 15 章：为地图设置样式](#)。

在本附录中：

- ◆ 填充图案..... 511
- ◆ 直线样式..... 525
- ◆ 矢量符号..... 525
- ◆ 定制符号..... 530
- ◆ **MapXtreme 图标**..... 532

填充图案

下表概述了 MapXtreme 填充图案（也称为 MapXtreme 编程 API 的内部样式）。每个填充图案都有一个关联的“索引编号”（用于对 `InteriorStyleRepository` 对象的可编程访问）和一个“图案值”（填充图案的内部描述名）。

具体如下：

填充图案

自身的图形填充图案。

索引编号

基于 0 的索引，用于检索表示 `InteriorStyleRepository` 中填充图案的内部样式。

图案值

填充图案的数字标识符，可用于构造内部样式对象。这只是一个内部值，不能通过编程方式访问 `InteriorStyleRepository`。不过，这些图案值可用于指示 MapXtreme 工作空间文件 (.mws) 中的填充图案。

了解索引编号结构

如“填充图案”表中所见，填充图案索引（列于“索引编号”列）的范围从 1 至 172。基于 1 的索引结构用于其它所有 MapInfo 产品（如 MapInfo Professional 和 MapBasic）。不过，在 MapXtreme API 中，填充图案（也称为内部样式对象）通过 `InteriorStyleRepository` 对象（该对象使用基于 0 的索引）获得。

具体而言，`InteriorStyleRepository` 对象允许用户通过基于 0 的索引结构（范围从 0 至 172）在所有可用内部样式对象上重复。因此，实际上有 173 个内部样式对象可供选择，不过，基于 0 和 1 的内部样式对象是等价的。基于以下原因，需要位于元素 0 和 1 的等价对象：

首先，我们必须提供 `InteriorStyleRepository` 类自身的基于 0 的内部索引功能。即，`InteriorStyleRepository` 类只允许基于 0 的索引访问，用于从其 `IList` 检索元素。（`IList` 是 `InteriorStyleRepository` 类实现的标准 .NET 接口。）

其次，基于 0 的索引结构必须与“图案值”列中显示的基于 1 的索引结构保持同步。因此，只有 172 个独立的填充图案，其索引范围从 1 至 172。已添加位于索引 0 的可编程填充图案，以给出该索引元素的定义。

i 您可能会注意到前八个填充图案都具有相同的“索引编号”和“图案值”。随后，“图案值”通常比对应的“索引编号”大 3（因此，“图案值”序列达到 175，而“索引编号”以 172 结束）。这不是出错。仍然有 172 个唯一的填充图案。表中未丢失任何内容。

填充图案及相关索引编号和图案值

填充图案	索引编号	图案值
 (None)	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	12
	10	13
	11	14
	12	15
	13	16

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	14	17
	15	19
	16	18
	17	20
	18	21
	19	22
	20	23
	21	24
	22	25
	23	26
	24	27
	25	28
	26	29

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	27	30
	28	31
	29	32
	30	33
	31	34
	32	35
	33	36
	34	37
	35	38
	36	39
	37	40
	38	41
	39	42
	40	43

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	41	44
	42	45
	43	46
	44	47
	45	48
	46	49
	47	50
	48	51
	49	52
	50	53
	51	54
	52	55
	53	56

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	54	57
	55	58
	56	59
	57	60
	58	61
	59	62
	60	63
	61	64
	62	65
	63	66
	64	67
	65	68
	66	69
	67	70

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	68	71
	69	72
	70	73
	71	74
	72	75
	73	76
	74	77
	75	78
	76	79
	77	80
	78	81
	79	82
	80	83
	81	84

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	82	85
	83	86
	84	87
	85	88
	86	89
	87	90
	88	91
	89	92
	90	93
	91	94
	92	95
	93	96
	94	97

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	95	98
	96	99
	97	100
	98	101
	99	102
	100	103
	101	104
	102	105
	103	106
	104	107
	105	108
	106	109
	107	110

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	108	111
	109	112
	110	113
△▽ ▽△	111	114
~~ ~ ~	112	115
▲ ▲	113	116
/~\~ ~\~\~	114	117
	115	118
	116	119
-○— —○-	117	120
○/ \○	118	121
○ ○	119	122
	120	123

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	121	124
	122	125
	123	126
	124	127
	125	128
	126	129
	127	130
	128	131
	129	132
	130	133
	131	134
	132	135
	133	136

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
■ ■	134	137
— — — —	135	138
	136	139
	137	140
	138	141
	139	142
× ×	140	143
	141	144
	142	145
	143	146
	144	147
	145	148
	146	149

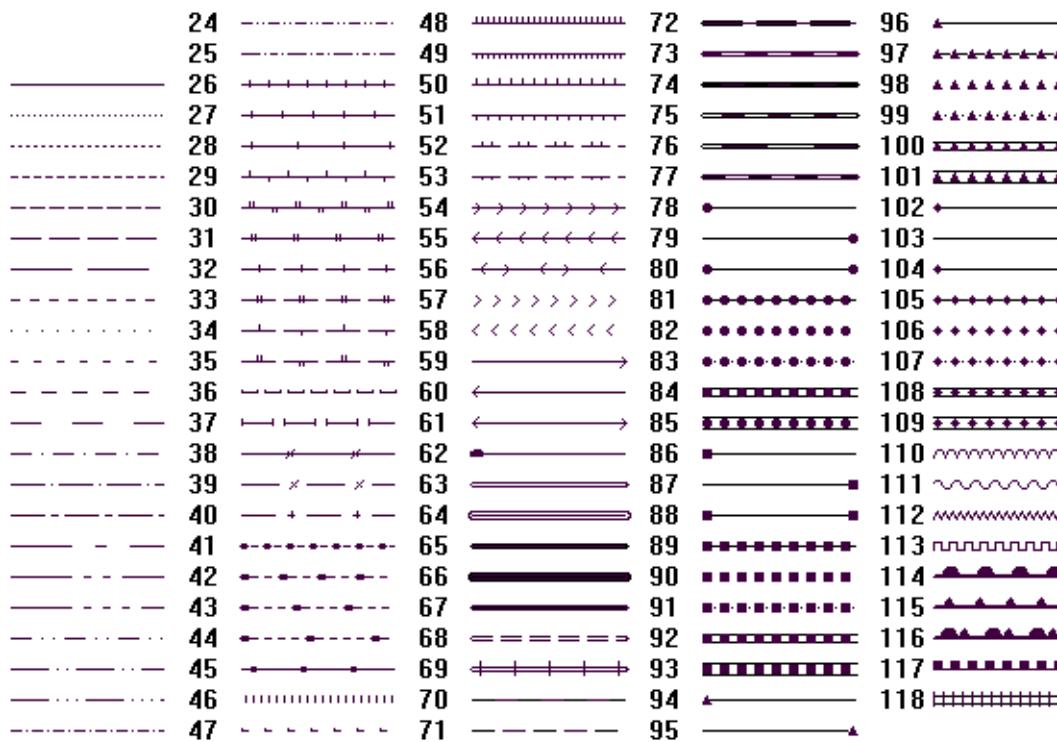
填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	147	150
	148	151
	149	152
	150	153
	151	154
	152	155
	153	156
	154	157
	155	158
	156	159
	157	160
	158	161
	159	162

填充图案及相关索引编号和图案值（续）

填充图案	索引编号	图案值
	160	163
	161	164
	162	165
	163	166
	164	167
	165	168
	166	169
	167	170
	168	171
	169	172
	170	173
	171	174
	172	175

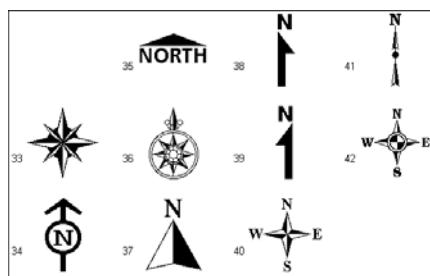
直线样式



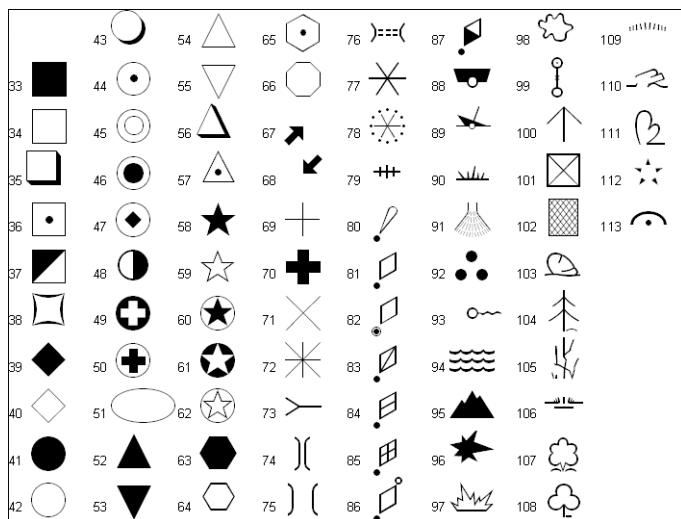
矢量符号

MapXtreme 在其安装过程中自动安装 10 种 MapInfo 特定的 TrueType 字体。这些字体为用户提供了字形符号选择，范围涉及天气、房地产、交通等。字形编号为 Unicode 字符值，由于这些编号位于第一个 Unicode 字符代码块范围内，因此，与 ASCII 字符集兼容。

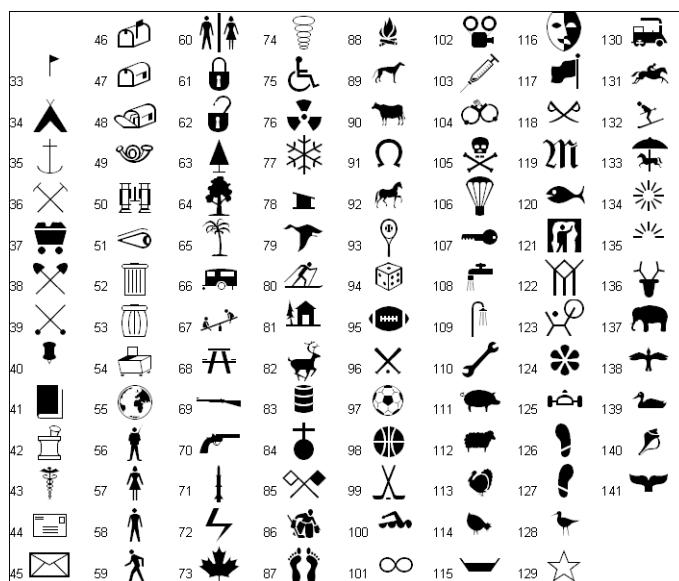
MapInfo 箭头



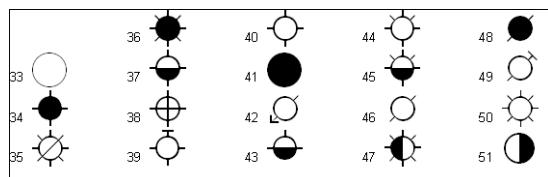
MapInfo 制图



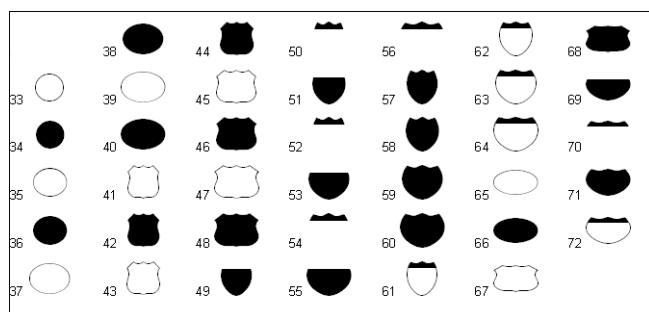
MapInfo 杂类



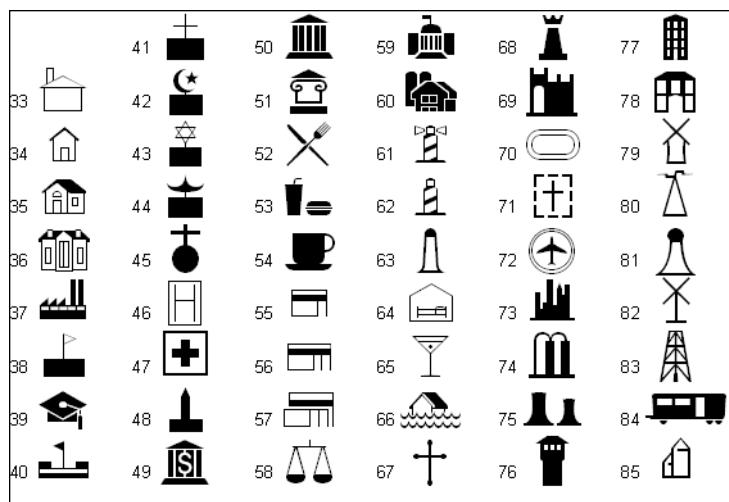
MapInfo 油气



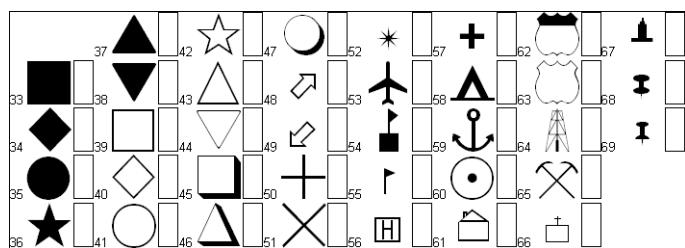
MapInfo 遮罩



MapInfo 房地产

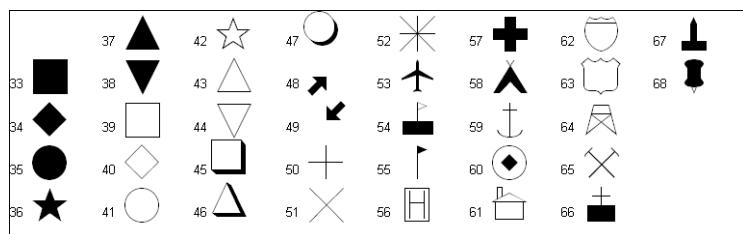


地图符号

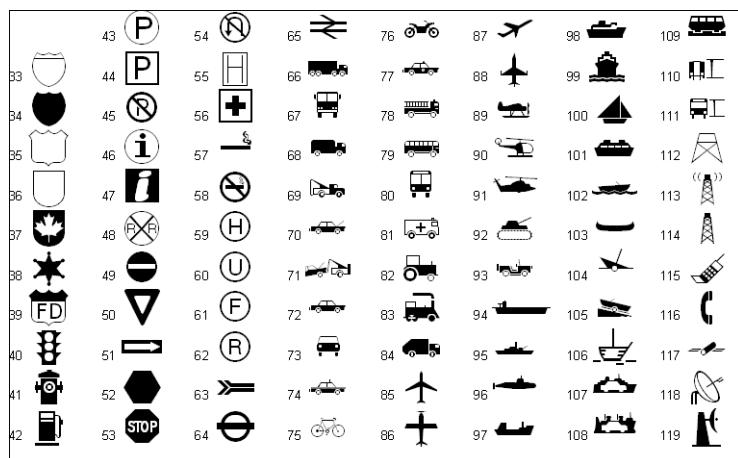


MapInfo 符号

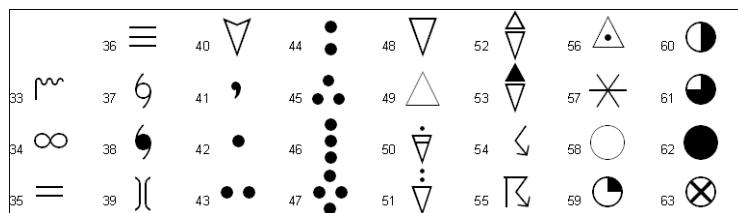
MapInfo 3.0 兼容符号



MapInfo 交通



MapInfo 天气

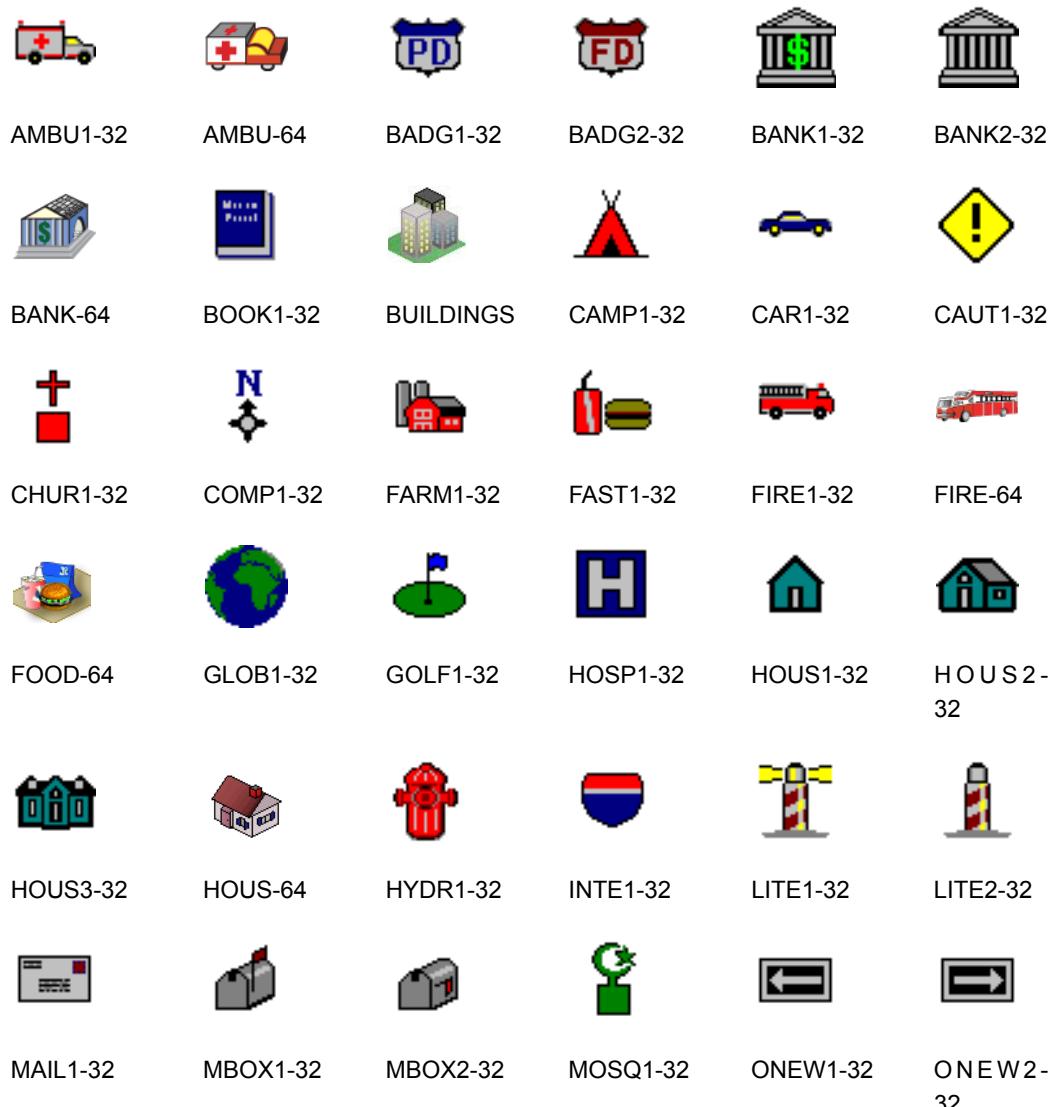


定制符号

以下符号位于 C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x\CustSymb。每个图像的文件扩展名都是 .BMP。

可以用编程方式通过 MapInfo.Styles 命名空间中的 BitmapPointStyleRepository 集合类访问这些符号。

可以创建自己的位图图像并将其添加到 CustSymb 目录。尽管事实上对创建的图像没有大小限制，不过 MapXtreme 显示图像的能力取决于可用的内存。图像不一定必须是方形，而且还可以具有最多 24 位颜色深度。要确保图像以其高度和宽度显示，则必须在各自图像的 BitmapPointStyle 对象中将 Boolean "NativeSize" 属性设置为 true。





PENC1-32



PIN1-32

(青色)



PIN2-32

(红色)



PIN1-32

(黄色)



PIN4-32

(绿色)



PIN5-32

(蓝色)



PIN6-32

(紫色)



PINB-64

(蓝色)



PING-64

(绿色)



PINGY-64

(灰色)



PINR-64

(红色)



POLI1-32



RAIL1-32



RAIL2-32



RAIL3-32



RAIL-64



RED-CAR



REST1-32



STAT1-32



STOP1-32



SYNA1-32



TARG1-32



TAXI1-32



TEMP1-32



TOWE1-32



TOWE2-32



TRAF1-32



TRUC1-32



TRUC2-32



TRUC-64



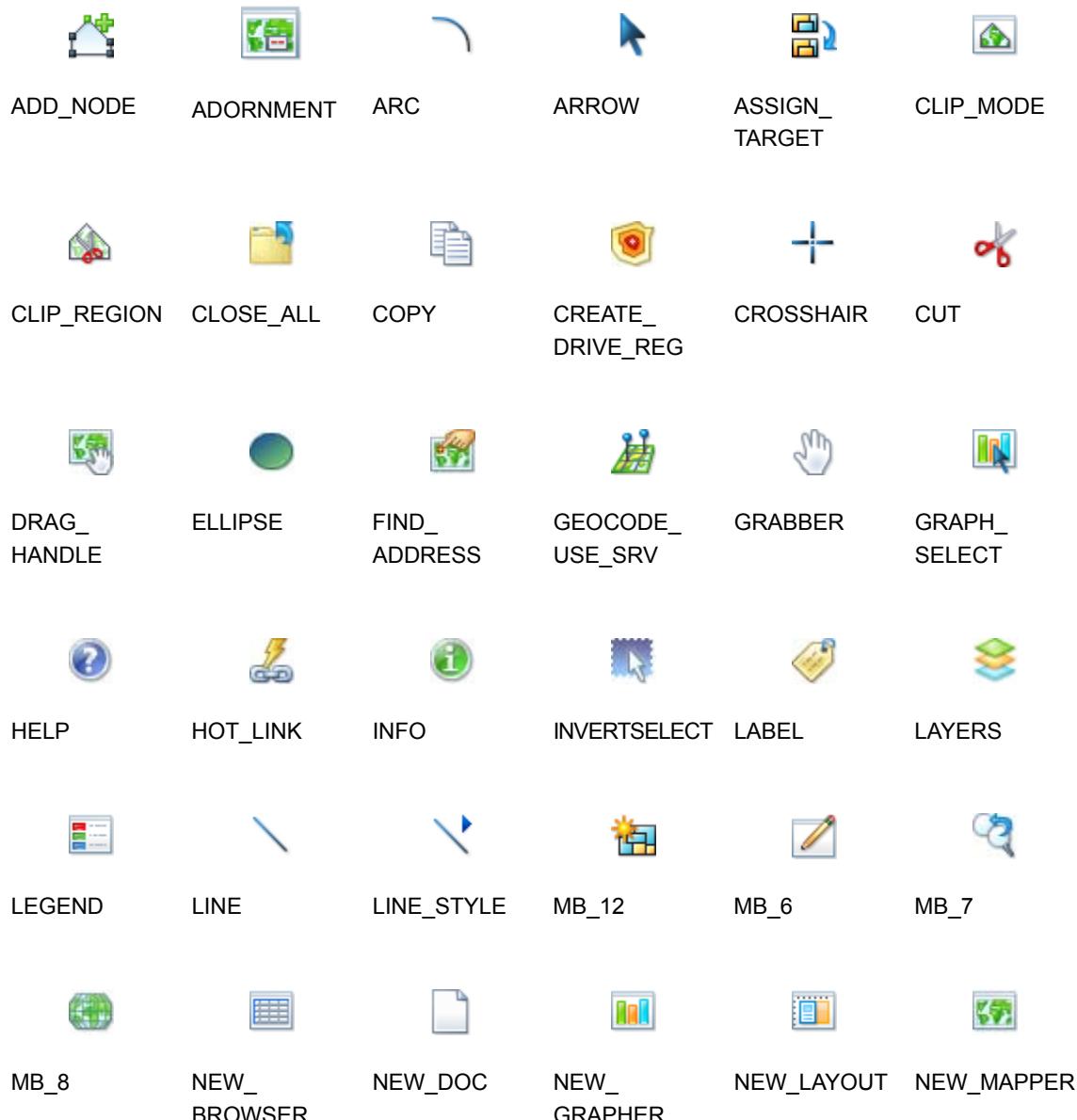
YIEL1-32

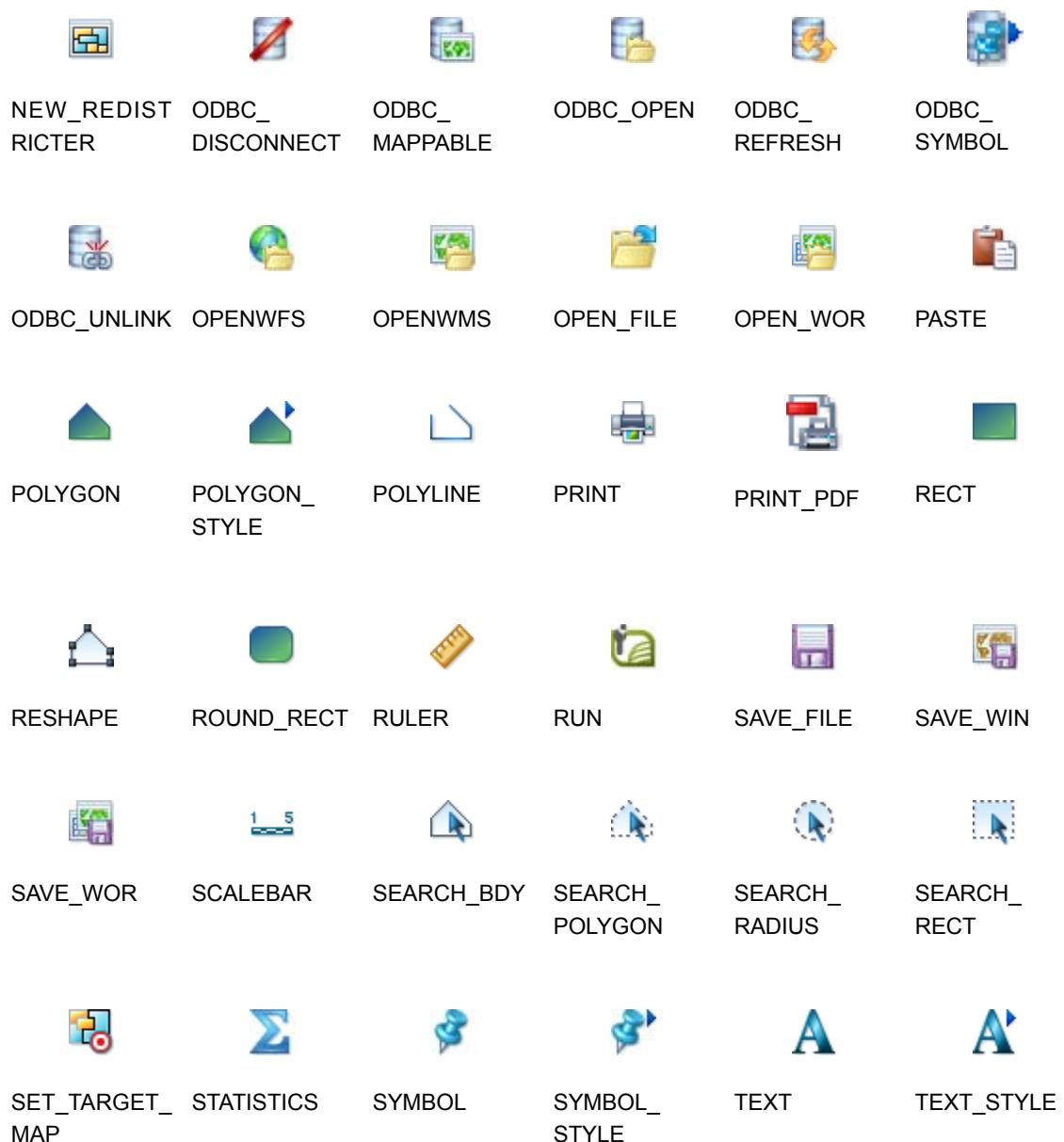


YIEL2-32

MapXtreme 图标

此缩图表表示可用于基于 MapXtreme 应用程序的工具栏图标集合。这些图标安装在 MapXtreme 安装文件夹下的 \Samples\Icons 文件夹。每个 .PNG 都有两种格式：小 .PNG (16x16 像素) 和大 .PNG (24x24 像素)。







定义 MapInfo Codespace

MapInfo Codespace 是定义和标准的列表，常用于创建 MapInfo 地图和工作区。可以参考这些定义以帮助使用 MapXtreme。您可能想要将本公司的 codespace 定义与另一种常用的 codespace - European Petroleum Survey Group (EPSG) 作比较，该文档可在其网站上找到：www.epsg.org/。

在本附录中：

- ◆ 定义 MapInfo Codespace 536

定义 MapInfo Codespace

在下表中，将说明目前的 MapInfo Codespace。这些定义将允许基于 XML 的地图文档使用本公司的 codespace 并参考公共值，例如 `srsName`。点几何体定义的示例可以如下所示：

```
<gml:Point srsName="mapinfo:coordsys 1,74"> ...</gml:Point> 或  
<gml:Point srsName="epsg:4269"> ...</gml:Point>
```

MapInfo Codespace 定义

类别	位置 1	位置 2	位置 3	位置 4	解释与示例
coordsys	CoordSys 文件*				坐标参考系。 示例: mapinfo:coordsys 1,74
pen	[1-255]				简单 Mapinfo 画笔图案； 编号 [1-255] 基于默认的 MapInfo MapInfo.pen 定义。遵循所有已建立的规则。 示例: mapinfo:pen 46
brush	[1-8, 12-175]				简单 Mapinfo 画刷图案编号 [1-8, 12-175]。（注：9-11 均保留）基于默认的 MapInfo 位图定义。 示例: mapinfo:brush 17
length	in ft yd mi mm cm m km sft nmi li ch rd pt twip pica deg pt= 1/72 in pica = 12 pt twip = 1/20 pt 1 deg = (pi/180) * 6370997 meter = 69.09329 mi (基于大圆周)				长度单位。 支持 inch foot yard mile 和 millimeter centimeter meter kilometer 和 survey foot nautical mile 和 link chain rod point twip pica degree. 示例: mapinfo:length m mapinfo:length pt mapinfo:length deg
imagesize	pixel				图像尺寸的非线性单位。 示例: mapinfo:imagesize pixel

MapInfo Codespace 定义（续）

类别	位置 1	位置 2	位置 3	位置 4	解释与示例
type					数据类型。
	boolean				示例： mapinfo:type boolean
	byte				示例： mapinfo:type byte
	date				示例： mapinfo:type date
	datetime				示例： mapinfo:type datetime
	time				示例： mapinfo:type time
	decimal	[(n n)]	n.m n 是总的位数 m 小数点后保留的位数， m<=n		最小值可选 最大值可选 精度可选 示例： mapinfo:type decimal (100 -- 所有 d>100 mapinfo:type decimal 100) -- 所有 d<100 mapinfo:type decimal [200 300] -- 所有 200<=d<=300 mapinfo:type decimal 10.7 - 所有总长度不超过 10 位的小数，小数点后不超过 7 位，那么整数部分最多有 3 位。 mapinfo:type decimal [200 300] 10.7 -- 前两种示例的结合。
	double	[(n n)]			示例： mapinfo:type double (0 150)
	float	[(n n)]			示例： mapinfo:type float 30.25]
	int	[(n n)]			示例： mapinfo:type int [0 180]
	short	[(n n)]			示例： mapinfo:type short
	string	n			最大长度 n 可选。 示例： mapinfo:type string 256
	char				示例： mapinfo:type char

MapInfo Codespace 定义（续）

类别	位置 1	位置 2	位置 3	位置 4	解释与示例
运算符					标量属性比较运算符或几何体运算符或布尔运算符。
	eq neq lt gt lteq gteq				二进制标量值运算符。 示例：mapinfo:op eq
	in not_in				标量值 in/not_in，标量值的枚举集，例如 value in {2, 50, 88, 95} value not_in {“NY”, “NJ”}。 示例： mapinfo:op in mapinfo:op not_in
	between not_between				标量值是在两个基于该标量值类型的顺序相关标量值之间，例如，value between {5, 25} value not_between {1may2000, 30may2000} 示例： mapinfo:op between mapinfo:op not_between
	like not_like				字符串值匹配 RHS 图案。RHS 图案的格式可以指定为数据源，例如，lastname like “Jo%”。（使用 Oracle 通配字符 ‘%’）。 示例： mapinfo:op like mapinfo:op not_like
	intersects				图元几何体 fg 和给定的 Polygon 相交。 示例：mapinfo:op intersects

MapInfo Codespace 定义（续）

类别	位置 1	位置 2	位置 3	位置 4	解释与示例
运算符 (续)	mbr_intersects				图元几何体 (fg) 的 mbr (最小绑定矩形) 和给定的 Polygon 的 mbr 相交。 示例: mapinfo:op mbr_intersects
	contains				图元几何体 fg 包含给定的 Point。给定的 Polygon 包含图元几何体 fg。 示例: mapinfo:op contains
	contains_centroid				图元几何体 fg 包含给定的 Polygon 的质心。给定的 polygon 包含图元几何体 (fg) 的质心 示例: mapinfo:op contains_centroid
和					布尔 and。 示例: mapinfo:op and
或					布尔 or。 示例: mapinfo:op or

MapInfo Codespace 定义（续）

类别	位置 1	位置 2	位置 3	位置 4	解释与示例
area					面积单位。 示例：mapinfo:area sq mi
	sq	in ft yd mi mm cm m km sft nmi li ch rd pt twip pica degree			平方线性单位。 示例：mapinfo:area sq mi
	acre a ha perch rood a = are (10m x10m 的面积) ha = hectare (100m x 100m 的面积) perch = 272.25 sq ft (1 平方 rod 16.5 ft) rood = 40 perch (1/4 acre)				预定义面积单位。 示例：mapinfo:area acre
time	msec sec min hr day week month year				时间单位。 Millisecond Second Minute, Hour Day Week Month Year 示例：mapinfo:time hr
angle	deg rad				角度度量单位。 degree radian 示例：mapinfo:angle rad
temp	K F C				温度单位。 Kelvin Fahrenheit Celcius 示例：mapinfo:temp C

* MapXtreme 的坐标系信息在 MapInfoCoordinateSystemSet.xml 文件中，该文件位于 C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x 目录下。

H

坐标系的元素

本附录详细介绍了坐标系的元素，包括支持的基准面、椭圆和变换。

在本附录中：

- ◆ 投影及其参数 542
- ◆ 投影基准面 546
- ◆ 基准面转换 557
- ◆ 定制基准面 558
- ◆ National Transformation v.2 (NTv2) 563
- ◆ 关于坐标系和投影的信息 565

投影及其参数

下表指示了适用于每个投影的参数，它们以出现在 MapInfoCoordinateSystemSet.xml 中的相关坐标系行中的顺序列出。该文档位于 C:\Program Files\Common Files\MapInfo\MapXtreme\7.x.x 目录下。

	基准面	单位	原点, 经度	原点, 纬度	标准纬线 1	标准纬线 2	方位角	比例因子	东伪偏移	北伪偏移	范围
Albers Equal-Area Conic	X	X	X	X	X	X			X	X	
Azimuthal Equidistant	X	X	X	X							X
Cassini-Soldner	X	X	X	X					X	X	
Cylindrical Equal Area	X	X	X		X						
球面双像	X	X	X	X				X	X	X	
埃克脱 IV	X	X	X								
Eckert VI	X	X	X								
Equidistant Conic	X	X	X	X	X	X			X	X	
Gall	X	X	X								
Hotine Oblique Mercator	X	X	X	X			X	X	X	X	
Lambert Azimuthal Equal-Area	X	X	X	X							X
Lambert Conformal Conic	X	X	X	X	X	X			X	X	
Longitude-Latitude	X										
Mercator	X	X	X								
Miller	X	X	X								
Mollweide	X	X	X								
New Zealand Map Grid	X	X	X	X					X	X	
Polyconic	X	X	X	X					X	X	
Regional Mercator	X	X	X		X						
Robinson	X	X	X								
Sinusoidal	X	X	X								

	基准面	单位	原点, 经度	原点, 纬度	标准纬线 1	标准纬线 2	方位角	比例因子	东伪偏移	北伪偏移	范围
Stereographic	X	X	X	X				X	X	X	
Swiss Oblique Mercator	X	X	X	X					X	X	
Transverse Mercator	X	X	X	X				X	X	X	

投影 (Projection)

投影是坐标系使用的等式或等式集。下表命名了 MapInfo 使用的投影并给出了用于标识 MapInfoCoordinateSystemSet 文件中投影的编号：

编号	投影 (Projection)
9	Albers Equal-Area Conic
28	Azimuthal Equidistant (all origin latitudes)
5	Azimuthal Equidistant (polar aspect only)
30	Cassini-Soldner
2	Cylindrical Equal-Area
31	球面双像
14	埃克脱 IV
15	Eckert VI
6	Equidistant Conic, 也称作 Simple Conic
17	Gall
7	Hotine Oblique Mercator
4	Lambert Azimuthal Equal-Area (polar aspect only)
29	Lambert Azimuthal Equal-Area
3	Lambert Conformal Conic
19	Lambert Conformal Conic (modified for Belgium 1972)
1	经度/纬度 (Longitude/Latitude)

编号	投影 (Projection)
10	Mercator
11	Miller Cylindrical
13	Mollweide
18	New Zealand Map Grid
27	Polyconic
26	Regional Mercator
12	Robinson
16	Sinusoidal
20	Stereographic
25	Swiss Oblique Mercator
8	Transverse Mercator, (也称做 Gauss-Kruger)
21	Transverse Mercator, (modified for Danish System 34 Jylland-Fyn)
22	Transverse Mercator, (modified for Danish System 34 Sjaelland)
23	Transverse Mercator, (modified for Danish System 34/45 Bornholm)
24	Transverse Mercator, (modified for Finnish KKJ)

例如，Longitude/Latitude 投影在 MapInfoCoordinateSystemSet.xml 中列出如下：

```
<GeographicCRS>
    <srsName>经度 / 纬度 (Porto Santo 1936)</srsName>
    <srsID>
        <code>coordsys 1,94</code>
        <codeSpace>mapinfo</codeSpace>
        <remarks>经度 / 纬度 (v 6.0 和更高版本投影) </remarks>
    </srsID>
    <usesEllipsoidalCS/>
    <usesGeodeticDatum/>
</GeographicCRS>
```

上述 MapInfoCoordinateSystemSet.xml 中的投影编号，可以通过将常数值添加到 Projection 表中所列基数的方法修改。有效值及其含义列表以下：

常数	含义	参数
1000	系统具有仿射变换	仿射单位指定符和系数在系统常规参数之后出现。
2000	系统具有显式边界	边界在系统常规参数之后出现。
3000	系统既有仿射又有边界	仿射参数跟随系统参数；边界跟随仿射参数。

示例：

假定要以简单系统工作，该系统基于 Transverse Mercator 投影并使用 NAD 1983 基准面。可以采用一行代码，例如 MapInfoCoordinateSystemSet.xml 文件中的以下示例：

```
<ProjectedCRS>
    <srsName>UTM 区域 1 (NAD 83)</srsName>
    <srsID>
        <code>coordsys 8,33,7,-177,0,0.9996,500000,0</code>
        <codeSpace>mapinfo</codeSpace>
        <remarks>通用横墨卡托</remarks>
    </srsID>
    <baseCRS/>
    <definedByConversion/>
    <usesCartesianCS/>
</ProjectedCRS>
```

需要基于以上的系统，但是要有以下参数指定的仿射变换：

```
Units=meters; A=0.5; B=-0.866; C=0; D=0.866; E=0.5; and F=0.
```

需要的 MapInfoCoordinateSystemSet.xml 文件中的代码是：

```
<ProjectedCRS>
    <srsName>UTM 区域 1 (NAD 83) - 旋转 60 度</srsName>
    <srsID>
        <code>coordsys 8,33,7,-177,0,0.9996,500000,0, 7, 0.5, -
            0.866, 0, 0.866, 0.5, 0 </code>
        <codeSpace>mapinfo</codeSpace>
        <remarks>通用横墨卡托</remarks>
    </srsID>
    <baseCRS/>
    <definedByConversion/>
    <usesCartesianCS/>
</ProjectedCRS>
```

或者，如果要将系统边界设置为 $(x_1, y_1, x_2, y_2)=(-500000, 0, 500000, 1000000)$ ，需要的代码是：

```
<ProjectedCRS>
    <srsName>UTM 区域 1 (NAD 83) - 有界的</srsName>
    <srsID>
        <code>coordsys 8,33,7,-177,0,0.9996,500000,0, -500000, 0,
            500000, 1000000
        </code>
    </srsID>
```

```
<codeSpace>mapinfo</codeSpace>
<remarks>通用横墨卡托 </remarks>
</srsID>
<baseCRS/>
<definedByConversion/>
<usesCartesianCS/>
</ProjectedCRS>
```

要使用以上的修改将系统定制，代码如下：

```
<ProjectedCRS>
  <srsName>UTM 区域 1 (NAD 83)</srsName>
  <srsID>
    <code>coordsys 8,33,7,-177,0,0.9996,500000,0, 7, 0.5, -
          0.866, 0, 0.866, 0.5, 0, -500000, 0, 500000,
          1000000 </code>
    <codeSpace>mapinfo</codeSpace>
    <remarks>通用横墨卡托 </remarks>
  </srsID>
  <baseCRS/>
  <definedByConversion/>
  <usesCartesianCS/>
</ProjectedCRS>
```

投影基准面

基准面是通过将参考椭面关联到地球上特定点的方法建立的。下表列出了每个基准面的详细信息：

- MapInfoCoordinateSystemSet.xml 文件中用于定义基准面的数字。
- 基准面的名称
- 通常使用基准面的地图
- 基准面的参考椭面

编号	基准面	区域地图	椭面
1	Adindan	埃塞俄比亚、马里、塞内加尔、苏丹	Clarke 1880
2	Afgooye	索马里	Krassovsky
1007	AGD 66, 7 参数	澳大利亚、A.C.T.	Australian National
1008	AGD 66, 7 参数	澳大利亚、塔斯马尼亚	Australian National
1009	AGD 66, 7 参数	澳大利亚、维多利亚/NSW	Australian National
1006	AGD 84, 7 parameter	澳大利亚	Australian National
3	Ain el Abd 1970	巴林群岛	International

编号	基准面	区域地图	椭面
118	American Samoa	美属萨摩亚群岛	Clarke 1866
4	Anna 1 Astro 1965	可可斯群岛	Australian National
119	Antigua Island Astro 1943	安提瓜、顺风群岛	Clarke 1880
5	Arc 1950	博茨瓦纳、莱索托、马拉维、斯威士兰、扎伊尔、赞比亚、津巴布韦	Clarke 1880
6	Arc 1960	肯尼亚、坦桑尼亚	Clarke 1880
7	Ascension Island 1958	阿森松岛	International
9	Astro B4 Sorol Atoll	燕鸥岛	International
8	Astro Beacon “E”	硫磺岛	International
10	Astro DOS 71/4	圣海伦岛	International
11	Astronomic Station 1952	马库斯岛	International
151	ATS77 (Average Terrestrial System 1977)	加拿大	ATS77
12	Australian Geodetic 1966 (AGD 66 - 3 param)	澳大利亚和塔斯马尼亚岛	Australian National
13	Australian Geodetic 1984 (AGD 84 - 3 param)	澳大利亚和塔斯马尼亚岛	Australian National
1006	Australia-National-AGD84 7 param	澳大利亚	Australian National
1007	Australia (A.C.T.AGD66 7-param)	澳大利亚	Australian National
1008	Australia (Tasmania AGD66 7-param)	塔斯马尼亚	Australian National
1009	Australia (Victoria/NSW AGD66 7-param)	维多利亚、NSW	Australian National
151	Average Terrestrial System 1977 (ATS77)		
120	Ayabelle Lighthouse	吉布提	Clarke 1880
110	Belgium	比利时	International

编号	基准面	区域地图	椭面
14	Bellevue (IGN)	埃法特和埃罗蒙哥群岛	International
15	Bermuda 1957	百慕大群岛	Clarke 1866
16	Bogota Observatory	哥伦比亚	International
121	Bukit Rimpah	邦加和勿里洞群岛 (印度尼西亚)	Bessel 1841
17	Campo Inchauspe	阿根廷	International
18	Canton Astro 1966	凤凰群岛	International
19	Cape	南非	Clarke 1880
20	Cape Canaveral	佛罗里达和巴哈马群岛	Clarke 1866
1005	Cape, 7 parameter	南非	WGS 84
21	Carthage	突尼斯	Clarke 1880
22	Chatham 1971	查塔姆岛 (新西兰)	International
23	Chua Astro	巴拉圭	International
122	Co-Ordinate System 1937 of Estonia	爱沙尼亚	Bessel 1841
24	Corrego Alegre	巴西	International
123	Dabola	几内亚	Clarke 1880
124	Deception Island	迷幻岛、南极洲	Clarke 1880
1000	Deutsches Hauptdreiecksnetz (DHDN)	德国	Bessel
25	Djakarta (Batavia)	苏门达腊岛 (印度尼西亚)	Bessel 1841
26	DOS 1968	Gizo 岛 (新乔治亚群岛)	International
27	Easter Island 1967	复活节岛	International
115	EUREF 89	欧洲	GRS 80
28	European 1950 (ED 50)	奥地利、比利时、丹麦、芬兰、法国、德国、直布罗陀、希腊、意大利、卢森堡、荷兰、挪威、葡萄牙、西班牙、瑞典、瑞士	International

编号	基准面	区域地图	椭面
29	European 1979 (ED 79)	奥地利、芬兰、荷兰、挪威、西班牙、瑞典、瑞士	International
108	European 1987 (ED 87)	欧洲	International
125	Fort Thomas 1955	尼维斯，圣克里托斯 利华德群岛	Clarke 1880
30	Gandajika Base	马尔代夫共和国	International
116	GDA 94	澳大利亚	GRS 80
32	Geodetic Reference System 1967 (GRS 67)	全球	GRS 67
33	Geodetic Reference System 1980 (GRS 80)	全球	GRS 80
126	Graciosa Base SW 1948	Faial、Graciosa、Pico、Sao Jorge 和 Terceira 群岛（亚速尔群岛）	International 1924
34	Guam 1963	关岛	Clarke 1866
35	GUX 1 Astro	瓜达康纳尔岛	International
150	Hartbeesthoek 94	南非	WGS 84
127	Herat North	阿富汗	International 1924
128	Hermannskogel	南斯拉夫（1990 年解体）、斯诺文尼亚、克罗地亚、波斯尼亚、黑塞哥维那、塞尔维亚	Bessel 1841
36	Hito XVIII 1963	南智利（靠近 53×S）	International
37	Hjorsey 1955	冰岛	International
38	Hong Kong 1963	中国香港特别行政区	International
1004	Hungarian Datum (HD 72)	匈牙利	GRS 67
39	Hu-Tzu-Shan	中国台湾地区	International
40	Indian	泰国和越南	Everest (India 1830)

编号	基准面	区域地图	椭面
41	Indian	孟加拉、印度、尼泊尔	Everest (India 1830)
129	Indian	巴基斯坦	Everest (Pakistan)
130	Indian 1954	泰国	Everest (India 1830)
131	Indian 1960	越南	Everest (India 1830)
132	Indian 1975	泰国	Everest (India 1830)
133	Indonesian 1974	印度尼西亚	Indonesian 1974
42	Ireland 1965	爱尔兰	Modified Airy
134	ISTS 061 Astro 1968	南乔治亚岛	International 1924
43	ISTS 073 Astro 1969	Diego Garcia	International
152	Japanese Geodetic Datum 2000 (JGD2000)	日本	GRS80
44	Johnston Island 1961	约翰斯顿岛	International
45	Kandawala	斯里兰卡	Everest (India 1830)
46	Kerguelen Island	凯尔盖朗岛	International
47	Kertau 1948	西马来西亚和新加坡	Everest (W.Malaysia 和 Singapore 1948)
135	Kusaie Astro 1951	卡罗琳群岛、密克罗尼西亚联邦	International 1924
48	L.C.5 Astro	大开曼群岛	Clarke 1866
136	Leigon	加纳	Clarke 1880
49	Liberia 1964	利比里亚	Clarke 1880
113	Lisboa (DLx)	葡萄牙	International
50	Luzon	菲律宾(不含棉兰老岛)	Clarke 1866

编号	基准面	区域地图	椭面
51	Luzon	棉兰老岛	Clarke 1866
52	Mahe 1971	马希岛	Clarke 1880
53	Marco Astro	Salvage 群岛	International
54	Massawa	厄立特里亚（埃塞俄比亚）	Bessel 1841
114	Melrica 1973 (D73)	葡萄牙	International
55	Merchich	摩洛哥	Clarke 1880
56	Midway Astro 1961	中途岛	International
57	Minna	尼日利亚	Clarke 1880
137	Montserrat Island Astro 1958	蒙特塞拉特、顺风群岛	Clarke 1880
138	M' Poraloko	加蓬	Clarke 1880
58	Nahrwan	Masirah 岛（阿曼）	Clarke 1880
59	Nahrwan	阿拉伯联合酋长国	Clarke 1880
60	Nahrwan	沙特阿拉伯	Clarke 1880
61	Naparima, BWI	特立尼达和多巴哥	International
109	荷兰	荷兰	Bessel
117	New Zealand Geodetic Datum 2000 (NZGD2000)	新西兰	GRS 80
31	New Zealand Geodetic Datum 1949 (NZGD 49)	新西兰	International
1010	(New Zealand (NZGD49 7-parameter))	新西兰	International
62	North American 1927 (NAD 27)	美国大陆	Clarke 1866
63	North American 1927 (NAD 27)	阿拉斯加	Clarke 1866
64	North American 1927 (NAD 27)	巴哈马（不含圣萨尔瓦多岛）	Clarke 1866

编号	基准面	区域地图	椭面
65	North American 1927 (NAD 27)	圣萨尔瓦多岛	Clarke 1866
66	North American 1927 (NAD 27)	加拿大（包括纽芬兰岛）	Clarke 1866
67	North American 1927 (NAD 27)	运河区	Clarke 1866
68	North American 1927 (NAD 27)	加勒比（特克斯和凯科斯群岛）	Clarke 1866
69	North American 1927 (NAD 27)	中美洲（伯利兹、哥斯达黎加、萨尔瓦多、危地马拉、洪都拉斯、尼加拉瓜）	Clarke 1866
70	North American 1927 (NAD 27)	古巴	Clarke 1866
71	North American 1927 (NAD 27)	格陵兰 (Hayes Peninsula)	Clarke 1866
72	North American 1927 (NAD 27)	墨西哥	Clarke 1866
73	North American 1927 (NAD 27)	密歇根州（只用于 State Plane Coordinate System 1927）	Modified Clarke 1866
74	North American 1983 (NAD 83)	阿拉斯加、加拿大、中美洲、美国大陆、墨西哥	GRS 80
139	North Sahara 1959	阿尔及利亚	Clarke 1880
107	Nouvelle Triangulation Francaise (NTF) Greenwich Prime Meridian	法国	Modified Clarke 1880
1002	Nouvelle Triangulation Francaise (NTF) Paris Prime Meridian	法国	Modified Clarke 1880
111	NWGL 10	全球	WGS 72
75	Observatorio 1966	Corvo and Flores 群岛（亚速尔群岛）	International

编号	基准面	区域地图	椭面
140	Observatorio Meteorologico 1939	Corvo and Flores 群岛（亚速尔群岛）	International 1924
76	Old Egyptian	埃及	Helmert 1906
77	Old Hawaiian	夏威夷	Clarke 1866
97	OldTokyo	日本、韩国、冲绳	Bessel 1841
78	Oman	阿曼	Clarke 1880
79	Ordnance Survey of Great Britain 1936	英格兰、怀特岛、苏格兰、设得兰群岛、威尔士	Airy
80	Pico de las Nieves	加拿利群岛	International
81	Pitcairn Astro 1967	皮特克恩岛	International
141	Point 58	布基纳法索和尼日尔	Clarke 1880
142	Pointe Noire 1948	刚果	Clarke 1880
157	Popular Visualization	全球	Popular Visualization
143	Porto Santo 1936	波尔图桑塔和马德拉群岛	International 1924
1000	Potsdam	德国	Bessel
82	Provisional South American 1956	玻利维亚、智利、哥伦比亚、厄瓜多尔、圭亚那、秘鲁、委内瑞拉	International
36	Provisional South Chilean 1963	南智利（靠近 53°S）	International
83	Puerto Rico	波多黎各和维京群岛	Clarke 1866
1001	Pulkovo 1942	德国	Krassovsky
1012	PZ90	俄罗斯	PZ90
84	Qatar National	卡塔尔	International
85	Qornoq	南格陵兰岛	International
1000	Rauenberg	德国	Bessel
86	Reunion	马斯克林岛	International

编号	基准面	区域地图	椭面
112	Rikets Triangulering 1990 (RT 90)	瑞典	Bessel
1011	Rikets Triangulering 1990 (RT 90), 7 parameter	瑞典	Bessel
87	Rome 1940	撒丁尼亚岛	International
88	Santo (DOS)	圣埃斯皮里托岛	International
89	So Braz	So Miguel, Santa Maria 群岛 (亚速尔群岛)	International
90	Sapper Hill 1943	东福克兰岛	International
91	Schwarzeck	纳米比亚	Modified Bessel 1841
144	Selvagem Grande 1938	Salvage 群岛	International 1924
145	Sierra Leone 1960	塞拉里昂	Clarke 1880
146	S-JTSK	捷克共和国	Bessel 1841
1013	SK42	俄罗斯	PZ90
1024	SK95	俄罗斯	PZ90
92	South American 1969	阿根廷、玻利维亚、巴西、智利、哥伦比亚、厄瓜多尔、圭亚那、圭亚那、秘鲁、委内瑞拉、特立尼达和多巴哥	South American 1969
93	South Asia	新加坡	Modified Fischer 1960
94	Southeast Base	Porto Santo and Madeira 群岛	International
95	Southwest Base	Faial、Graciosa、Pico、Sao Jorge、Terceira 群岛 (亚速尔群岛)	International
1003	Switzerland (CH 1903)	瑞士	Bessel
147	Tananarive Observatory 1925	马达加斯加	International 1924
96	Timbalai 1948	文莱和东马来西亚 (沙捞越和沙巴)	Everest (India 1830)

编号	基准面	区域地图	椭面
1015	Tokyo	日本	Bessel 1841
98	Tristan Astro 1968	崔斯坦火山岛	International
99	Viti Levu 1916	梅地来雾群岛（斐济群岛）	Clarke 1880
148	Voirol 1874	突尼斯/阿尔及利亚	Clarke 1880
149	Voirol 1960	阿尔及利亚	Clarke 1880
100	Wake-Eniwetok 1960	马歇尔群岛	Hough
101	World Geodetic System 1960 (WGS 60)	全球	WGS 60
102	World Geodetic System 1966 (WGS 66)	全球	WGS 66
103	World Geodetic System 1972 (WGS 72)	全球	WGS 72
104	World Geodetic System 1984 (WGS 84)	全球	WGS 84
105	Yacare	乌拉圭	International
106	Zanderij	苏里南	International

单位

下表列出了 MapInfoCoordinateSystemSet.xml 文件中可用的坐标单位和用于标识单位的编号：

编号	单位
6	厘米
31	链
3	英尺（也称为国际英尺）*
2	英寸
1	千米
30	令
7	米

编号	单位
0	英里
5	毫米
9	海里 [†]
32	杆
8	美国测量英尺（用于 1927 State Plane）‡
4	码

* 一国际英尺精确等于 30.48 厘米。

† 一海里精确等于 1852 米。

‡ 一美国测量英尺精确等于 $12/39.37$ 米，或近似等于 30.48006 厘米。

坐标系原点

原点是以经度和纬度指定的点，所有坐标都从该原点引用。选择原点的目的是优化特定坐标系的精确性。从原点向北移动时，Y 值增加。从原点向东移动时，X 值增加。这些坐标值通常称为北偏移和东偏移。

对于 Transverse Mercator 投影，原点的经度定义了中心子午线。在构造 Transverse Mercator 投影的过程中，定位一个柱面与地球相切。中心子午线是切线。沿着中心子午线，投影地图的比例为真。

在创建 Hotine Oblique Mercator 投影的过程中，有必要指定一个既不是赤道也不是子午线的大圆周。MapInfo 通过指定一个椭面上的点和从该点发出的方位角以实现该功能。那个点正是坐标系的原点。

标准纬线（圆锥投影）

在圆锥投影中，一个锥形沿着纬线的两条平行线穿过与它相交的地球区域。这两条平行线就是标准纬线。一条在投影区域的北面，另一条在投影区域的南面。要使用单一标准纬线两次指定那条纬线。它们都是用纬线的度数表示。

倾斜方位角（Hotine Oblique Mercator）

使用点和方位角指定上述的大圆周 (Hotine Oblique Mercator) 时，方位角被称为 Oblique Azimuth 并用度数表示。

比例因子 (Transverse Mercator)

比例因子应用于柱面坐标，以均分沿着东部和西部边界减少错误时地图中心区域的比例错误。比例因子有使柱面凹进地球区域的效果，所以它有两条相交线。沿着这两条相交线比例为 true。

比例因子可以表示为比值的形式，例如 1:25000。这种情况下它通常被称比例减少。比例因子和比例减少之间的关系是：

比例因子 = 1-比例减少

这种情况下比例因子就是 $1-(1/25000)$ 或 0.99996。

北伪偏移和东伪偏移

如果不涉及负数，则坐标的计算会更简单。要在计算 State Plane 和 Universal Transverse Mercator 坐标的过程中排除这个问题，常用的方法是将测量偏移量增加到北偏移和东偏移。这些偏移量被称为北伪偏移和东伪偏移。它们以坐标单位表示，而不是以度表示。（坐标单位由 **Units** 参数指定。）

范围（Azimuthal 投影）

以度为单位指定的范围，表示观测的地球区域的大小。范围可能介于 1 和 180 之间。指定的是 90 度时，就可以看到半球区域。指定的是 180 度时就可以看到整个地球，尽管它的大部分区域是扭曲的。

Polyconic 投影

以下描述节选自 *Map Projections – A Working Manual*, USGS Professional Paper 1395, 作者 John P. Snyder。

Polyconic 投影，通常被称为 American Polyconic in Europe，因为地图上每个平行圆弧的曲率是相同的而得名。它是以纬线的特定平行线为切线的地球外切锥形，沿着其表面的平行轨迹分解而生成的。因此就涉及到很多 ("poly-") 锥形，而不是每个圆锥投影的单一锥形。

Polyconic 投影既不是等区域投影也不是等角度投影。然而沿着中心子午线，就没有失真且比例为 true。每条平行线的比例为 true，但是子午线被各种数量延长以在平行线的正确位置穿过每条平行线。所以没有平行线符合标准等角度投影（或正确角度）的意义，除了中心子午线以外。靠近中心子午线的位置，失真量极其微小。

该投影方法不适合绘制大区域地图。绘制经度范围大的地图时，转换算法将失效。例如，如果使用 Polyconic 重投影，则样本表 WORLD.TAB 可能显示异常。

基准面转换

将坐标从一个基准面转换到另一个基准面时，MapInfo 已使用的有 Molodensky (3 参数) 和 Bursa-Wolf (7 参数) 方法。它们是可以将坐标从任何基准面转换到任何其它的基准面的多用途方法。

在引入 NAD 83 基准面之后，NOAA 开发了名为 NADCON 的程序，是 North American Datum CONversion 的缩写。它是非常特定的程序，只能将 NAD 27 转换为 NAD 83，或将 NAD 83 转换为 NAD 27。对于这项特定的任务，它比 Molodensky 常用方法具有更高的精确性；NADCON 的精确度可以达到大约 0.1 米，而 Molodensky 的精确度只能达到 10-30 米。大多数美国政府机构，包括人口调查局，已经将 NAD 27 和 NAD 83 之间转换的 NADCON 方法标准化。

从 MapInfo 4.1.2 开始，如果地图坐标位于 NADCON 覆盖的区域之内（美国、波多黎各和维京群岛），则使用 NADCON 算法在 NAD 27 和 NAD 83 之间转换坐标。如果坐标位于以上区域之外，或如果它们使用 NAD 27 或 NAD 83 以外的其它基准面，则 MapInfo 使用 Molodensky 或 Bursa-Wolfe 转换方法。

由于需要访问文件，NADCON 转换方法的速度将比 Molodensky 方法略微缓慢。如果要关闭 NADCON 转换，则要在 MapXtreme 程序或 MapXtreme Common 目录下删除 *.las 和 *.los 文件。

定制基准面

基准面是地球的形状和方位的数学描述。因为地球的形状是不规则的，所以有很多不同的局部基准面用于世界的不同地区。这些局部基准面提供在特定区域与地球表面接近的近似值。

每个地球坐标系使用特定的基准面以近似表示地球的表面。将坐标从一个坐标系转换到另一个坐标系时，如果两个坐标系使用了不同的基准面，则 Pitney Bowes Software 绘制的地图必须执行基准面变换。Pitney Bowes Software 使用 Bursa-Wolfe 基准面变换方法，其精确度通常在 10 米之内。（当转换是在两个使用相同基准面的坐标系之间进行时，无需执行基准面变换，其结果的精确度通常在 0.1 米之内。）

定义定制基准面

大多数坐标系使用本附录列出的 Pitney Bowes Software 预定义基准面之一。如果需要使用不在列表中的基准面，并知道该基准面的数学参数是什么，则可以使用定制基准面定义坐标系。MapInfo 使用以下信息定义基准面：

- 椭面，也称为球面。这是绕它的短轴旋转以构成三维表面的椭圆。该椭面由两个数学参数描述：它的短半轴长度，单位为米（由字母 **a** 表示）和它的修整度（由字母 **f** 表示）。MapInfo 支持 40 种以上的预定义椭面，在下一个表中列出。
- 三个平移参数，指定沿着其每个轴平移椭面的距离（单位为米）。这些参数通常由 **dX**、**dY** 和 **dZ** 表示。也可由 **DX**、**DY** 和 **DZ** 或 **u**、**v** 和 **w** 表示。
- 三个旋转参数，指定绕着其每个轴旋转椭面的角度（单位为弧度-秒）。这些参数通常由 **EX**、**EY** 和 **EZ** 表示。也可以由 **eX**、**eY** 和 **eZ** 或 **e**、**y** 和 **w** 表示。
- 比例校正因子，指定每百万部分中的数量，以调节椭面的大小。该参数由字母 **m** 表示，有时也由 **k** 表示。
- 本初子午线的经度，以东格林威治为基准面，单位为度。本初子午线指定地球上哪个位置分配经度 0°。大多数基准面使用格林威治作为本初子午线，所以该参数通常为 0。不过，某些基准面采用不同位置做为本初子午线。例如，NTF 基准面将巴黎做为其本初子午线，它是格林威治以东 2.33722917 度。如果在坐标系中使用 NTF 基准面，则该坐标系中所有经度都相对于巴黎，而不是格林威治。

可以在任何坐标系定义中定制自定义的基准面。使用基准面编号 9999，后跟基准面参数，按以下顺序：

```
9999, EllipsoidNumber, dX, dY, dZ, EX, EY, EZ, m, PrimeMeridian
```

某些基准面只指定椭面和平移参数 (dX、dY、dZ)，没有旋转参数、比例校正因子或本初子午线。这种情况下，可以使用基准面编号 999 替代 9999，以简化定义：

999, EllipsoidNumber, dX, dY, dZ

椭面编号必须从以下列表选择。目前尚没有方法定制自定义的椭面。如果需要使用本列表中没有出现的椭面，请通知 MapInfo 技术支持，我们可以将您的椭面增加到未来的 MapInfo 版本中。

编号	椭面	a	1/f
9	Airy 1930	6377563.396	299.3249646
13	Airy 1930 (为 Ireland 修改 1965)	6377340.189	299.3249646
51	ATS7 77	6378135.0	298.257
2	Australian	6378160.0	298.25
10	Bessel 1841	6377397.155	299.1528128
35	Bessel 1841 (为 NGO 修改 1948)	6377492.0176	299.15281
14	Bessel 1841 (为 Schwarzeck 修改)	6377483.865	299.1528128
36	Clarke 1858	6378293.639	294.26068
7	Clarke 1866	6378206.4	294.9786982
8	Clarke 1866 (为 Michigan 修改)	6378450.047484481	294.9786982
6	Clarke 1880	6378249.145	293.465
15	Clarke 1880 (为 Arc 修改 1950)	6378249.145326	293.4663076
30	Clarke 1880 (为 IGN 修改)	6378249.2	293.4660213
37	Clarke 1880 (为 Jamaica 修改)	6378249.136	293.46631
16	Clarke 1880 (为 Merchich 修改)	6378249.2	293.46598
38	Clarke 1880 (为巴勒斯坦修改)	6378300.79	293.46623
39	Everest (Brunei and East Malaysia (Sabah and Sarawak))	6377298.556	300.8017
11	Everest (India 1830)	6377276.345	300.8017
40	Everest (India 1956)	6377301.243	300.80174
50	Everest (Pakistan)	6377309.613	300.8017

编号	椭面	a	1/f
17	Everest (W.Malaysia 和 Singapore 1948)	6377304.063	300.8017
48	Everest (West Malaysia 1969)	6377295.664	300.8017
18	Fischer 1960	6378166.0	298.3
19	Fischer 1960 (为南亚修改)	6378155.0	298.3
20	Fischer 1968	6378150.0	298.3
21	GRS 67	6378160.0	298.247167427
0	GRS 80	6378137.0	298.257222101
5	Hayford	6378388.0	297.0
22	Helmer 1906	6378200.0	298.3
23	Hough	6378270.0	297.0
31	IAG 75	6378140.0	298.257222
41	Indonesian	6378160.0	298.247
4	International 1924	6378388.0	297.0
49	Irish (WOFO)	6377542.178	299.325
3	Krassovsky	6378245.0	298.3
32	MERIT 83	6378137.0	298.257
33	New International 1967	6378157.5	298.25
43	NWL 10D	6378135.0	298.26
42	NWL 9D	6378145.0	298.25
44	OSU86F	6378136.2	298.25722
45	OSU91A	6378136.3	298.25722
46	Plessis 1817	6376523.0	308.64
54	Popular Visualization	6378137.0	0.0
52	PZ90	6378136.0	298.257839303
24	South American	6378160.0	298.25

编号	椭面	a	1/f
12	Sphere	6370997.0	0.0
47	Struve 1860	6378297.0	294.73
34	Walbeck	6376896.0	302.78
25	War Office	6378300.583	296.0
26	WGS 60	6378165.0	298.3
27	WGS 66	6378145.0	298.25
1	WGS 72	6378135.0	298.26
28	WGS 84	6378137.0	298.257223563

平移和旋转参数描述椭面在空间中的方位（与 WGS 84 基准面相比较）。确保这些参数含有正确的符号（正号或负号）非常重要。通常，描述局部基准面的文档会列出将坐标从该局部基准面转换为 WGS 84 所需的参数。也就是说，这些参数是通过从 WGS 84 减去局部基准面的方法派生。）这种情况下，可以准确使用文档中出现的参数。不过，如果文档中列出的转换坐标的参数是反方向的（从 WGS 84 转换为局部基准面），则必须将平移、旋转和比例校正因子的符号取反。

以正确的顺序列出参数也非常重要。某些文档以 EZ 为首列出旋转参数，例如：EZ、EY 和 EX。在这些情况下，定义定制基准面时必须将旋转参数的顺序反向。文档使用希腊字母表示参数时，这种问题尤其容易被忽略。如果文档以 w、y、e 的顺序列出参数，则必须在定制基准面的定义中将它们的顺序反向。

以下是局部基准面描述的示例（名为 LD-2），可出现在技术文章中：

LD-2 椭面：Krassovsky

a	6378245.0 m
f	1 / 298.3

从 LD-2 转换为 WGS 84

u	+24 m
v	- 123 m
w	- 94 m
w	+0.13
y	+0.25

e	0.02
m	+1.1 10-6

该基准面使用 Krassovsky 椭面，在上文的椭面表中的编号为 3。因为参数描述的是从局部基准面到 WGS 84 的转换，所以不需要将参数的符号取反。不过，旋转参数是以 w 作为首参数，所以必须在定制基准面的定义中将旋转参数的顺序反向：

9999, 3, 24, -123, -94, -0.02, 0.25, 0.13, 1.1, 0

最后的示例 LD-3，只提供了椭面和平移参数：

LD-3 椭面：Clarke 1880

a	6378249.145 m
f	1 / 293.465

WGS 84 减去局部基准面 LD-3

dX	-7 m
dY	36 m
dZ	225 m

该基准面使用 Clarke 1880 椭面，在上文的椭面表中的编号为 6。不需要将参数的符号取反或担心旋转参数的顺序（因为它们没有出现）。在这种情况下，可以在定制基准面的定义中使用基准面编号 999 替代 9999。以下两种定义是等价的，可以使用任意一种：

999, 6, -7, 36, 225
9999, 6, -7, 36, 225, 0, 0, 0, 0, 0

对于其它定制基准面的定义，会在 MapInfoCoordinateSystemSet.xml 行的基准面编号位置插入这些定义中的一个，如下所示：

```
"Longitude / Latitude (LD-3)", 1, 999, 6, -7, 36, 225
```

```
<srsName>UTM 区域 30 (LD-3)</srsName>
- <srsID>
  <code>coordsys 8, 999, 6, -7, 36, 225, 7, -3, 0, 0.9996, 500000, 0
  </code>
  <codeSpace>mapinfo</codeSpace>
  <remarks>通用横墨卡托 (LD-3)</remarks>
</srsID>
```

National Transformation v.2 (NTv2)

National Transformation v.2 (NTv2) 算法和网格交换文件格式由 Geodetic Survey Division of Geomatics Canada 开发，让您能够将 NAD 27 参考系统的数据转换为 NAD 84 参考系统数据。使用此算法的网格转换文件包含一个或多个矩形网格，此网格表示 NAD 27 和 NAD 83 之间的坐标差。

National Transformation 的设计初衷是将来自 NAD 27 的加拿大数据转换为 NAD 84。后来，此格式也用于转换澳大利亚、新西兰和德国的数据。

NT v.2 算法已实现，用于支持这些转换，如今已为以下国家添加网格转换文件：

- 加拿大
- 澳大利亚
- 新西兰
- 德国

这些文件位于 MapXtreme Common Files 文件夹中。此文件的默认位置是：

C:\Program Files\Common Files\MapInfo\MapXtreme\7.0.0

除了上述算法和网格转换文件以外，还安装了一个 XML 配置文件 NTv2.xml，通过该文件，您可以扩展 NTv2 算法以便支持更多的数据转换。您可以向该配置文件添加新的网格转换文件，也可以针对特殊的网格转换文件打开或关闭 NTv2 算法。NTv2.xml 文件位于 MapXtreme Common Files 文件夹中。

NTv2 算法和适用于加拿大的网格转换文件受以下版权保护：

© 1995 Her Majesty the Queen in Right of Canada, represented by the Minister of Natural Resources.

下一节介绍各个国家/地区支持的转换格式以及如何使用配置文件。

加拿大

以下网格转换文件适用于加拿大：

NTV2_0.GSB - 用于将 NAD 1927 转换为 NAD 1983

MAY76V20.GSB - 用于将 NAD 1927 (1976 版定义) 转换为 NAD 1983 for Ontario

这些文件在 NAD 1927 或 NAD 1927 (1976 版定义) 数据和 NAD 1983 数据之间进行转换。NAD 1927 数据 (1976 版定义) 坐标系是重新调整的 NAD 1927 for Ontario。

有关算法、软件和网格转换文件的详细信息，可从 Geodetic Survey Division of Geomatics Canada 网站上下载：

www.geod.nrcan.gc.ca/index_e.php

澳大利亚

以下网格转换文件适用于澳大利亚：

A66_National.gsb - 用于将 AGD 1966 转换为 GDA94。

National_84.gsb - 用于将 AGD 1984 转换为 GDA94。

Australian 网格转换文件在 AGD 1966 或 AGD 1984 数据与 GDA 1994 数据之间进行转换。

有关转换和网格转换文件的详细信息，可从 Intergovernmental Committee on Surveying and Mapping (ICSM) 网站上下载：

www.icsm.gov.au/icsm/gda/gdatm/

新西兰

以下网格转换文件适用于新西兰：

nzgd2kgrid0005.gsb - 用于将 NZGD49 数据转换为 NZGD2000 数据。

New Zealand 网格转换文件将 NZGD49 数据转换为 NZGD2000 数据。

有关详细信息及在线转换程序可在 Land Information New Zealand (LINZ) 网站上找到：

www.linz.govt.nz/geodetic/conversion-coordinates/online-conversion-service

德国

以下网格转换文件适用于德国：

BETA2007.gsb - 用于将 DHDN 数据转换为 ETRS89

欲了解详细信息并下载官方网格转换文件，请访问 Information and Service System for European Coordinate Reference Systems (CRS EU) 网站：

www.crs-geo.eu/

配置文件

NTv2 配置文件 (NTv2.xml) 是一个 XML 文件，通过编辑该文件，您可以为其他数据转换添加网格转换文件并针对特定的网格转换文件打开或关闭 NTv2。

要添加新的网格转换文件，必须指定三个参数：

- 网格转换文件名称
- 源数据
- 目标数据

数据的定义有多种方式。例如，当源数据和目标数据均在 MapXtreme 中定义时，在该配置文件中将按照在 MapXtremeCoordinateSystemSet.xml 文件中为数据分配的编号定义数据。在新西兰示例中，此配置文件中的 XML 如下（粗体部分仅用于表示强调）：

```
- <NTv2Conversion>
  <Description>NTv2 Conversion for New Zealand</Description>
  <GridFile>NZGD2KGRID0005.GSB</GridFile>
  <Enabled>true</Enabled>
- <SourceDatum>
  <DatumID>1010</DatumID>
</SourceDatum>
- <DestinationDatum>
```

```
<DatumID>117</DatumID>
</DestinationDatum>
</NTv2Conversion>
- <NTv2Conversion>
```

然而，有时数据的定义并没有这么简单。为了定义数据，您可能需要提供数据转换值（有时甚至需要提供数据范围）。例如，以下 XML 示例显示了适用于加拿大的 NAD 27 数据定义。此处，使用椭面、(x, y, z) 转换值、值范围来定义数据（粗体部分仅用于表示强调）：

```
- <NTv2Conversion>
  <Description>NAD 1927 to NAD 1983 conversion for Canada</Description>
  <GridFile>NTV2_0.GSB</GridFile>
  <Enabled>true</Enabled>
- <SourceDatum>
  <EllipsoidID>7</EllipsoidID>
  <ShiftX>-8</ShiftX>
  <ShiftY>150.5</ShiftY>
  <ShiftZ>186</ShiftZ>
  <dShiftX>17</dShiftX>
  <dShiftY>19.5</dShiftY>
  <dShiftZ>8</dShiftZ>
  </SourceDatum>
- <DestinationDatum>
  <DatumID>74</DatumID>
  </DestinationDatum>
</NTv2Conversion>
```

关于坐标系和投影的信息

以下列出的前三种出版物是相对简短的小册子。最后两种是内容充实的书籍。我们还提供了 American Congress of Surveying and Mapping（小册子）和 U.S. Geological Survey（书籍）的地址和电话号码。

- American Cartographic Association.Choosing a World Map—Attributes, Distortions, Classes, Aspects.Falls Church, VA: American Congress on Surveying and Mapping.Special Publication No. 2.1988.
- American Cartographic Association.Matching the Map Projection the Need.Falls Church, VA: American Congress on Surveying and Mapping.Special Publication No. 3.1991.
- American Cartographic Association.Which Map is Best?Projections for World Maps.Falls Church, VA: American Congress on Surveying and Mapping.Special Publication No. 1.1986.
- John P.Snyder。Map Projections—A Working Manual.Washington: U.S.Geological Survey Professional Paper 1395.1987
- John P.Snyder and Philip M.Voxland.An Album of Map Projections.Washington: U.S.Geological Survey Professional Paper 1453.1989.
- 联系信息

- American Congress on Surveying and Mapping, 5410 Grosvenor Lane, Suite 100, Bethesda, MD 20814 2212; (301) 493-0200
- Earth Science Information Center, U.S.Geological Survey, 507 National Center, Reston, VA 22092; (703) 860-6045 or (800) USA-MAPS
- Peter H.大学地理系的 Peter H. Dana 还建立了一个关于地图投影、Geodetic 基准面和坐标系的网站。因为该网站的很多说明也是使用 MapInfo Professional 展示的，所以很有价值。网站的材料可以用于学习、研究和教育目的，但请保留作者的版权：
- Peter H.Dana, The Geographer’s Craft Project, Department of Geography, The University of Texas at Austin.

关于 Geodetic 基准面的信息和说明，请访问：

www.colorado.edu/geography/gcraft/notes/datum/datum.html

关于坐标系和其它原理的信息，请访问：

www.colorado.edu/geography/gcraft/notes/coordsys/coordsys.html

关于地图投影的信息请访问：

www.colorado.edu/geography/gcraft/notes/mapproj/mapproj.html

用户定义的元数据

本附录包含有关支持 **TableInfoServer** 查询的用户定义的元数据的信息。

在本附录中：

- ◆ 元数据和 **MapCatalog** 568
- ◆ 支持 **TableInfoServer** 查询的用户定义的元数据 568

元数据和 **MapCatalog**

执行空间数据库查询时，一直将 **MapCatalog** 用作唯一的元数据源来推导查询结果集中 **FeatureGeometry** 和 **Style** 列的准确定义。但是，在某些情况下，**MapCatalog** 可能不是一种方便、合适或可靠的来源。例如：

- 视图 - 定义远程数据库服务器中的视图表时，麻烦在于要依赖数据库管理员在 **MapCatalog** 中添加相应的条目。这可能不方便，也存在内容不准确之风险。
- 存储过程 - 作为空间结果集的潜在动态生成器，可能无法将单一静态的条目录入可准确表示所有结果集的 **MapCatalog** 中，所生成的这些结果集可作为存储过程调用的输出。

 存储过程查询目前仅支持 SQL Server。

- 包含多个空间列的表 - 尽管空间数据库可能允许表包含多个空间列，但是 **MapCatalog** 构架目前仅允许为表的单一空间列定义元数据。
- 复杂查询 - 由于以下多种原因，即使 **MapCatalog** 包含适用于复杂查询所引用的基表的准确信息，推理逻辑也可能不可靠。
 - 元数据推理逻辑可能难以查找正确的元数据（例如，复杂的连接查询）。
 - **MapCatalog** 元数据可能因几何体中应用的函数而不准确（例如，坐标系转换、缓冲点和凸多边形集合等）。
 - 推理逻辑可能不知道如何正确调和为两个或多个基表标识的元数据（例如，**UNION** 查询）。

支持 **TableInfoServer** 查询的用户定义的元数据

MapInfo.Data.TableInfoServer 类包含的属性可用于为表中的 **FeatureGeometry** 和 **Style** 列定义元数据。此用户定义的元数据或列提示，在没有 **MapCatalog** 时标识这些列。

ColumnHints 属性

添加到 **TableInfoServer** 中的 **ColumnHints** 属性可使用户以明确的方式提供用户定义的元数据，以影响对查询结果集中 **FeatureGeometry** 和 **Style** 列的正确标识。这些“提示”主要为了在无法找到 **MapCatalog** 元数据时提供完整的元数据，但是，这些提示也可用作潜在稀疏定义以选择性地覆盖 **MapCatalog** 元数据的组件。在任何一种情况下，用户定义的元数据均视为决定性的，因此，建议用户仅在必要时及仅在明确理解查询结果详情时才使用列提示。

ColumnHints 属性定义为 **Columns** 集合。创建集合实例并为其中的每条提示插入 **Column** 实例后，则将此集合分配至 **ColumnHints** 属性，例如，名称为 **tblInfoSrv…** 的 **TableInfoServer** 变量。

```
[C#]  
Columns hints = new Columns();  
// insert Column instances into the collection to serve as hints (see  
below)  
tblInfoSrv.ColumnHints = hints;
```

可将 **FeatureGeometry** 和 **Style** 以外类型的 **Column** 实例添加到这些提示的 **Columns** 集合中，但目前将忽略这些其他列定义。

应用 **FeatureGeometry Column** 提示

可使用 **FeatureGeometry** 列提示表明，查询结果中命名列包含的空间值可解释为图元几何体。例如，如果查询返回的称为 **geo** 的列包含空间对象（例如，**SpatialWare ST_Spatial** 二进制或 **Oracle Spatial MDSYS.SDO_GEOMETRY** 结构），我们则可标识该列，只需构建提示并将其添加到 **Columns** 集合（从以上），如下所示：

```
[C#]  
GeometryColumn geoCol = new GeometryColumn("geo");  
...  
hints.Add(geoCol);
```

上述代码片断中的省略号表明，完成构建提示并将其添加到 **Columns** 集合之前，也需要设置 **GeometryColumn** 实例的其他属性。**FeatureGeometry** 提示主要用于明确地提供 **MapCatalog** 所提供的相同信息，具体如下：

- 坐标系统
- **FeatureGeometry** 类型
- 默认视图
- 默认样式

如果考虑所有这些属性，**FeatureGeometry** 提示的更为完整的示例可能类似于：

```
[C#]  
CoordSysFactory cfs = new CoordSysFactory();  
CoordSys cs = cfs.CreateFromMapBasicString("Earth Projection 1, 0");  
GeometryColumn geo = new GeometryColumn("geo", cs);  
geo.PredominantGeometryType = GeometryType.Polygon;  
geo.DefaultView = new DRect(-74, 40, -70, 44);  
geo.DefaultStyle = new MapInfo.Styles.AreaStyle(  
new MapInfo.Styles.SimpleLineStyle(new LineWidth(2.0,  
MapInfo.Styles.LineWidthUnit.Pixel)),  
new MapInfo.Styles.SimpleInterior(10));  
hints.Add(geo);
```

如果已知存在的 **MapCatalog** 条目可为空间列提供默认元数据，则可使用“稀疏”提示来覆盖特定的属性。通常，尽管可根据复杂查询的独特要求覆盖任何属性集，但是，这些将可能是参数选择多样性属性，例如默认视图和默认样式。

ColumnHints 不可用于 (x,y) 数据表。鉴于这些原因，应继续使用 **SpatialSchemaXY**。在表中应用 **SpatialSchemaXY** 时，将忽略任何关联的 **ColumnHints**。

因为 **MapXtreme** 表目前仅支持单一 **FeatureGeometry** 列，所以，**Columns** 集合期望包含多个 **FeatureGeometry** 列定义，不过，**Columns** 集合不阻止用户随意添加尽量多的列定义。在此情况下，在编译最终元数据定义时，将允许使用所遇到的每种属性的最后非稀疏值。

应用 **Style Column** 提示

可应用 **Style** 列提示来标明，查询结果中的列包含的值（特别是 **MapBasic** 样式字符串）可用于为每个图元创建 **Style** 实例，实例与该图元中的关联 **FeatureGeometry** 相对应。例如，如果查询返回的称为 **mb_style** 的列包含 **MapBasic** 样式字符串，我们则可构建提示并将其添加到提示集合中（从以上），如下所示：

```
[C#]  
Column styleCol = new Column("mb_style", MIDbType.Style);  
hints.Add(styleCol);
```

因为表目前仅支持单一 **Style** 列，所以，提示集合期望包含单个 **Style** 列定义，但是，**Columns** 集合不阻止用户随意添加尽量多的列定义。在此情况下，将允许使用最后插入的列定义。

提示集合中不需要 **Style** 列提示。如果查询包含的列不包含样式信息，则可排除此提示。

KeyType 和 **KeyColumns** 属性

每个表必须拥有可为表中每个图元（行）提供唯一值的“键”。尽管不是由 **MapCatalog** 提供，另一个重要的元数据却可通过查看诸如主键定义之类的内容对远程数据库表进行推导。

以上所列的许多内容源及仅包含组合主键的表，也为推导要使用的唯一键提出了挑战。在这些情况下，通过基本 **TableInfo** 类提供的 **KeyType** 和 **KeyColumns** 属性可用于提供明确的指示。例如，可考虑使用包含称为 **CUSTID** 的列的客户信息数据库视图，用户已知该列包含唯一的非空值。我们可将该列用作所生成的 **MapXtreme** 表的键，如下所示：

```
[C#]  
tblInfoSrv.KeyType = KeyType.Explicit;  
StringCollection keyColumns = new StringCollection();  
keyColumns.Add("CUSTID");  
tblInfoSrv.KeyColumns = keyColumns;
```

在此示例中，**StringCollection** 类驻留在 **System.Collections.Specialized** 命名空间中。

可使用复合键，在字符串集合中添加两个或多个列名即可定义复合键。

当数据库表包含两个或多个候选项且当开发人员有特定的参数选择可供使用时，也可将明确的键定义用作首要定义。

工作空间永久性

用户定义的列提示和键定义的元数据结构永久用作工作空间文件 (.mws) 中数据源定义标志的一部分。

J

迁移到 MapXtreme

本附录适用于 MapX 的现有用户，他们熟悉 MapXtreme 中基于 .NET 的对象模型以及与 MapX 体系结构的不同之处。

在本附录中：

- ◆ 比较 MapXtreme 和 MapX 的对象模型 572

比较 MapXtreme 和 MapX 的对象模型

若已使用过 MapX 和 MapXtreme for Windows，将会发现它们与 MapXtreme 工作方式的相似和不同之处。本节描述了 MapXtreme 和先前 API 版本之间的一些主要不同之处。

特定对象模型实现的不同之处

源于 MapX 的 Map 对象现在用 3 或 4 个类表示。Map 包含了图层、重点区域（视图）、Adornment 和 Legend（无论以何种方式查看 Map）。MapControl 保存 Map 对象并作为 Map 的控件和与 Map 交互的工具。此外还具有既可用于 Windows Form（桌面）应用程序又可用于 ASP.NET (web) 应用程序的 MapControl。MapExport 类可用于将 Map 导出到文件或数据流。Session 保存 Maps、Selections 和 Tables 的集合。现在完全支持制图比例。标注图层可用于更好地控制标注的位置。组图层允许将大量图层在图层层次中视为一个图层。

MapX Dataset 概念在 `MapInfo.Data` 命名空间中已由一组更为灵活的选项替换。MapXtreme 使用 ADO.NET 进行数据访问。ADO.NET 可以使用非常类似的代码访问各种数据格式，简化了数据源之间的转换。增加了更多光标用于通过数据库表向前和向后访问。

MapXtreme 使用以表为中心的数据模型。在以前版本的 API 中，地图由图层组成，但没有特定的表类。在 MapXtreme 对象模型中，表是用于访问数据的中心对象。`Session.Catalog` 用于打开和枚举表。图层可以引用表，但所有表特定的方法从 `Layer` 移动到到表。另外，搜索在 MapXtreme 中的表上而不是在图层上执行。

修改样式的主题不再使用其自身的图层，例如范围主题和单值主题。在以前的模型中这种主题占用独立的图层，而且使用时独立于地图的其它图层。对象主题仍然保留其自身的图层且现在可以用选择的任何方式来显示，独立于主题所基于的原始地图对象。

几何体已更改为层次模型，可以更深入地访问对象的属性。样式也以层次表示。

此外还增加了大量地图工具供您使用。增强了这些工具的可扩展性和功能性，使得开发定制的工具更加容易。还有用于 web 应用程序（也可扩展）中的工具。

对于 web 应用程序，不再需要 `MapXBroker` 和 `MapXServer`。相反，MapXtreme 使用标准的 COM+ 将 Session 对象置入对象池。

应用程序状态通过串行化和 ASP.NET 状态服务器来管理。有关状态管理和序列化的详细信息请参阅 [第 6 章：了解状态管理](#)。

下表列出了 MapX 5.0 对象模型中的对象和 MapXtreme 当前对象模型中的等价对象。请注意，由于大范围重新设计了 MapXtreme 产品，因此等价的对象可能是近似的。

MapX 5.0	MapXtreme
AffineTransform	AffineTransform 它是不可变的（无设置）。使用 <code>CoordSysFactory.CreateAffineTransform</code> 来创建。
AllFeaturesConstraint	
Annotation	已过时。新的 <code>Adornments</code> 类。 <code>MapControl</code> 包含从 <code>Control</code> 中派生的修饰
Annotations	已过时。
BindLayer	已过时。替换对象位于 <code>Table.AddColumn()</code> 参数中。
BitmapSymbol	BitmapPointStyle
BitmapSymbols	
BoundsConstraint	
CoordSys	CoordSys. 它是不可变的（无设置）。使用 <code>CoordSysFactory</code> 创建。
Dataset	已过时。 替换对象： <ul style="list-style-type: none">表达式列：<code>Table.AddColumn()</code> 可以创建表达式列。手动数据绑定（动态或静态）：<code>Table.AddColumn()</code> 使用以 <code>TableInfoClient</code> 打开的 <code>Table</code>，指向 ADO.NET <code>DataTable</code>。自动数据绑定：由 <code>Geodictionary</code> 功能处理。非 ADO.NET 外部数据XY 绑定：<code>SpatialSchemaXY</code> 应用到 <code>TableInfo.SpatialSchema</code> 属性。PointRef 绑定：<code>SpatialSchemaPointRef</code> 应用到 <code>TableInfo.SpatialSchema</code> 属性。 RowValues: <code>MIDataReader</code> , <code>MIScrollableReader</code> , 请参阅 <code>MICommand</code> . Refresh: <code>Table.RefreshColumns()</code>
Datasets	已过时。请参阅 <code>Dataset</code> 。
Datum	Datum 它是不可变的（无设置）。使用 <code>CoordSysFactory.CreateDatum()</code> 来创建。
Feature	Geometry, 样式和键信息是 Feature。

MapX 5.0	MapXtreme
Features	请参阅 Feature。
FeaturesConstraint	
FeatureFactory	<p>表级别对象处理</p> <ul style="list-style-type: none">• Buffer: 对于每个图元，缓冲区都位于 Geometry 级别 (FeatureGeometry.Buffer)。对于多个图元，请使用 FeatureProcessor.Buffer。• Combine: 对于每个图元，组合都位于 Geometry 级别 (FeatureGeometry.Combine)。对于多个图元，请使用 FeatureProcessor.Combine。 <p>Intersect: 对于每个图元，相交都位于 Geometry 级别 (FeatureGeometry.Intersect)。对于多个图元，请使用 FeatureProcessor.Intersect。</p>
Field	
Fields	
查找	查找
FindFeature	FindResult
FindMatch	FindCloseMatch, FindAddressRange
FindMatches	FindCloseMatchEnumerator, FindAddressRangeEnumerator
FindResult	FindResult
Geoset	已过时。类似功能：MapLoader
Geoset	已过时。MapXtreme 打开 geosets。
Graphic	已过时。
IndividualValueCategory	IndividualValueThemeBin
IndividualValueCategories	ModifierThemeBins
Label	LabelLayer, LabelSource
Labels	LabelLayer, LabelSource
LabelProperties	LabelProperties

MapX 5.0	MapXtreme
Layer	<p>UserDrawLayer、LabelLayer、FeatureLayer、ObjectThemeLayer、GroupLayer...</p> <ul style="list-style-type: none">• AddFeature: 在使用 MICommand 执行插入操作期间, 当前的等价对象要将 Geometry 绑定到几何体列。• AllFeatures• AutoLabel: 不再存在。等价对象将 LabelSource.DefaultLabelProperties.Enabled 属性设置为 true。• Begin/EndAccess: 未记录• Bounds: FeatureLayer.Bounds• ClearCustomLabels: 未记录• ClippedBounds: 已过时。使用 Map.SetView()• CoordSys• Datasets: 已过时。图层具有 Table (FeatureLayer.Table.)• DeleteFeature: 等价对象将通过 MICommand 删除行。• DrawLabelsAfter: 此概念已很少使用。类似的功能是在每个 Layer 后面都使用 LabelLayer 实例。唯一的缺点是每个 LabelLayer 都清理其自己的标注缓存, 因而标注将不会在图层之间交互。• Editable: FeatureLayer.Editable• FeatureIDFromFeatureName: 等价对象将是通过 MICommand 使用的 SQL 命令。

MapX 5.0	MapXtreme
Layer (续)	<ul style="list-style-type: none">• Find: 查找• GetDrillDownFeaturesByID: 未记录• GetFeatureByID: 等价功能是使用 MICommand 的键光标。• Invalidate: IMapLayer.Invalidate• KeyField: 已过时。等价功能是 MICommand。• LabelAtPoint• LabelProperties: LabelSource.DefaultLabelProperties。• Labels• Name: FeatureLayer.Name 或 Alias。• NoFeatures• OverrideStyle: 等价对象是 FeatureLayer.Modifiers 集合。• Pack: 等价对象是 FeatureLayer.Table.Pack• PredominantFeatureType: MISpatialColumnInfo.PredominantObjectType。可以从 Table.TableInfo.ColumnInfos 属性获取列。• Refresh: 刷新缓存。Equivalent: Table.Refresh()• Search: 等价对象是 MICommand。但是，结果可能不仅仅包含图元键。• SearchAtPoint: 无“简单的”等价对象。• SearchWithinDistance: 无“简单的”等价对象。• SearchWithinRectangle: 无“简单的”等价对象。• SearchWithinFeature: 无“简单的”等价对象。• Selectable: FeatureLayer.Selectable。• Selection: Selection class in Session.Sessions 集合。• ShowCentroids: FeatureLayer.ShowCentroids• ShowLineDirection: FeatureLayer.ShowLineDirection• ShowNodes: FeatureLayer.ShowNodes• Style: 等价对象目前是 FeatureLayer.Modifiers 集合。• SupportsPack: Table.SupportsPack• Type: IMapLayer.Type• UpdateFeature: 通过使用 MICommand 的等价对象• Visible: IMapLayer.Enabled。IMapLayer.Visible 考虑：• ZoomLayer: IMapLayer.ZoomRangeEnabled• ZoomMax: IMapLayer.ZoomRange.End• ZoomMin: IMapLayer.ZoomRange.Start

MapX 5.0	MapXtreme
Layers	Layers <ul style="list-style-type: none">• AnimationLayer 等价对象: GroupLayer 和 BackingStoreState.Off• InsertionLayer 等价对象: AddMapTool.InsertionLayer• Selections: 位于 Session 对象中。
LayerInfo	已过时。等价对象是 TableInfo
Legend	Legend
LegendText	LegendRow
LegendTexts	LegendRows

MapX 5.0	MapXtreme
Map	<p>目前，Map 已分成两个实体：Map 和 MapControl。</p> <p>MapControl 包含适用于 System.Windows.Forms.Control 的功能；它是可嵌入 WinForm 的对象。</p> <p>Map 无论以何种方式查看（不管它是 WinForm 还是导出），均包含实际的地图绘制功能。</p> <p>下面是先前 Map 类中的属性/方法列表以及如何在 MapXtreme 模型中进行处理：</p> <ul style="list-style-type: none">Annotations: 已过时。 <p>CenterX/Y: Map.Center</p> <p>ConvertCoord: DisplayTransform.ToDisplay(), DisplayTransform.FromDisplay()</p> <ul style="list-style-type: none">CreateCustomTool: 从 MapTool 或任何可用的 Custom* 工具继承，它们通过使用鼠标提供选取线绘制功能。CurrentTool: MapTools.LeftButtonTool、MapTools.MiddleButtonTool、MapTools.RightButtonTool。可以通过 MapControl.MapTools 属性访问 MapTools。Datasets: 已过时。可通过 MICatalog 访问表。Dataset: 已过时。DatasetGeoField: 已过时。请参阅 Map.Dataset.DatasetTheme: 已过时。请参阅 Map.Dataset. <p>DefaultStyle</p> <ul style="list-style-type: none">DisplayCoordSys: Map.GetDisplayCoordSys()。Distance: CoordSys.Distance()。DynamicSelectionSupport: MapTools.DynamicSelectionModeSelectMapTool.DynamicSelectionEnabledEditableLabelsExportMap: MapExport 类可以使用各种选项来导出地图。ExportSelection: MapExport.ExportSelectionFeatureEditMode 枚举：<ul style="list-style-type: none">miEditModeFeature -> MapTools.NodeMode = falsemiEditModeNode -> MapTools.NodeMode = truemiMoveDuplicateNodes -> MapTools.MoveDuplicateNodesmiDeleteDuplicateNodes -> MapTools.DeleteDuplicateNodesmiEditModeAddNode -> MapTools.AddNodeModeFeatureFactory: FeatureProcessorGeoDictionary: GeodictionaryGeoset: 已过时。

MapX 5.0	MapXtreme
Map (续)	<ul style="list-style-type: none">• Geosets: 已过时。• GeosetWidth• hWnd• InfotipPopupDelay• InfotipSupport: 未记录。• IsPointVisible• Layers: Layers• MapPaperHeight/Width: PaperSize (通过 MapExport.ExportSize)• MapScreenHeight/Width: MapControl.Size, Map.Size• MapUnit: 显示坐标系的 CoordSys.Units• MatchNumericFields: GeoDictionary 功能。• MatchThreshold: GeoDictionary 功能。• MaxSearchTime: GeoDictionary 功能。• MilitaryGridReferenceToPoint: CoordSys.MilitaryGridToPoint()• MilitaryGridReferenceFromPoint: CoordSys.PointToMilitaryGrid()• Mouseicon: MapControl.Cursor• MousePointer: MapControl.Cursor• MouseWheelSupport: MapControl.MouseWheelSupport.MouseWheelBehavior• NumericCoordSys: 已过时。几何体现在有其自己的 CoordSys。具有坐标 的任何内容都有坐标系。• 平移• PanAnimationLayer• PreferCompactLegends• PrintMap: Map.Draw()• PropertyPage• RedrawInterval: Map.IncrementalDraw.Interval。• Refresh• ReuseEquivalentOnRestore• Rotation: Map.Rotation。• SaveMapAsGeoSet• SearchPath.• SelectionStyle: Selection.Style• SetSize• SnapToNodeSupport• SnapTolerance: MapTools.SnapTolerance• Title: Adornments• TitleText• 版本• WaitCursorEnabled• 缩放• ZoomTo

MapX 5.0	MapXtreme
MultivarCategory	MultiVariableThemeCategory
MultivarCategories	MultiVariableThemeCategories
NotesQueryInfo	已过时。
NotesViewInfo	已过时。
OCIQueryInfo	已过时。
ODBCQueryInfo	已过时。
Parts	等价
Point	DPoint
Points	已过时
RangeCategory	RangedThemeBin
RangeCategories	ModifierThemeBins
Rectangle	DRect
ResolveObject	MatchResolver
ResolveObjects	MatchResolver 集合。
RowValue	MICommand
RowValues	MICommand
Selection	Selection
SourceRow	SourceRow
SourceRows	SourceRows 集合
State	
样式	
Theme	ObjectTheme, FeatureStyleModifier
Themes	已过时。Themes 目前由 ObjectThemeLayer (对于 ObjectThemes) 包含或包含在 Layer 的 Modifiers 集合中 (对于 FeatureStyleModifier 主题图)。
ThemeProperties	已过时。Properties 位于主题图类本身内。
Title	Adornments

MapX 5.0	MapXtreme
Variable	MIParameter.由于在表达式中绑定，MICommand.Parameters 属性可以定义在该命令中使用的变量。
Variable	MIParameterCollection.

K

本地化工具包

本地化工具包是一种 Visual Studio 解决方案，它可以将软件文本元素（错误消息、对话/控件文本）转换为除英语、日语和简体中文之外的语言。

在本附录中：

- ◆ 本地化工具包 583
- ◆ 如何使用本地化工具包 585
- ◆ 卫星程序集的私有钥匙标记 587

本地化工具包

MapXtreme 提供一种 Visual Studio 解决方案，开发人员可用它来转换错误消息和对话/控件文本元素，以便放在他们自己的 MapXtreme 应用程序中使用。

“本地化工具包”包含 MapXtreme 中所有运行时组件的资源项目。它经过组织，形成了一种 Visual Studio 解决方案，可方便开发人员编辑资源字符串和创建程序集，同时还能进行许多后台资源管理。此解决方案可用于 Visual Studio、Visual C# Express Edition，也可用于 MSBuild 命令行构建实用程序。

在每个项目中，均包含可转换的英语资源字符串和强制命名键 (.snk) 文件。将编译到程序集中，以便集成到 MapXtreme 应用程序。按照下表中的信息，即可轻松确定哪个项目需要转换以用于应用程序。您也可在每个项目内逐层查看，以仅查找需要转换的资源文件或单个字符串。

LocalizationKit.sln 包括以下项目：

工程	描述
GeoDictionaryManager.resources	提供用于 GeoDictionaryManager.exe 的错误字符串和对话框文本元素
MapInfo.CoreEngine.resources *	提供与 MapXtreme 核组件相关的字符串（地图绘制、数据访问、图元、样式、主题图、空间处理等）
MapInfo.LinearReferencing.resources	提供与 Z 和 M 值支持以及线性参照运算相关的字符串
MapInfo.Ogc.resources	提供在 FeatureGeometries 和 OGC 通用文本/二进制数据之间转换的相关字符串。
MapInfo.Services.resources	提供用于地理编码和路径规划的字符串。
MapInfo.Web.resources	提供与 Web 应用程序相关的字符串。
MapInfo.WebControls.resources	提供与 Web 控件和工具相关的字符串。
MapInfo.Wfs.Server.resources	提供与 WFS 服务器错误消息相关的字符串。
MapInfo.Windows.Dialogs.resources	提供与 Windows 对话框控件相关的字符串
MapInfo.Windows.resources	提供与 Windows 控件相关的字符串
MapInfo.Wms.Client.resources	提供与 WMS 错误消息相关的字符串。
WorkspaceManager.resources	提供用于 WorkspaceManager.exe 的错误字符串和对话框文本元素

* 请参阅下表，以了解与此命名空间相关的资源信息。

资源文件	产品范围
EllisAllTypeResources.en-US.resx	用于从数据源读取的错误字符串
EllisCommandProcessorResources.en-US.resx	用于从数据源读取的错误字符串
EllisCoordSysExceptions.en-US.resx	用于错误坐标系值的错误字符串
EllisCoordSysResources.en-US.resx	用于错误坐标系值的错误字符串
EllisDAEngineResources.en-US.resx	用于数据源查询错误的错误字符串
EllisDBInfoResources.en-US.resx	用于数据库错误的错误字符串
EllisDBLayerResources.en-US.resx	用于地图错误的错误字符串
EllisExceptions.en-US.resx	用于通用系统错误的错误字符串
EllisExprPacketCreatorResources.en-US.resx	用于 SQL 查询的错误
EllisExprPacketResources.en-US.resx	用于 SQL 查询的错误
EllisFcNInfoServerResources.en-US.resx	用于 SQL 查询的错误
EllisFindResources.en-US.resx	用于“查找”操作的字符串
EllisGeoObjectProcessResources.en-US.resx	用于 GeoObject 操作（线段、多边形等）的错误字符串
EllisGeoObjectResources.en-US.resx	用于 GeoObject 操作（线段、多边形等）的错误字符串
EllisGeoResources.en-US.resx	用于 GeoDictionary 和文件加载的错误字符串
EllisGeosetResources.en-US.resx	用于 Geoset 文件的错误字符串
EllisGmlXlatResources.en-US.resx	用于 GML 文件的错误字符串
EllisLegendResources.en-US.resx	用于“图例”的字符串
EllisMapBasicInternalFcnResources.en-US.resx	用于 MapBasic 函数的错误字符串
EllisMapBasicTranslatorResources.en-US.resx	用于 MapBasic 函数的错误字符串
EllisMapperResources.en-US.resx	用于 Map 操控/查找函数的错误字符串
EllisMILexerResources.en-US.resx	用于字符串分析（SQL、MapBasic 等）函数的错误字符串

资源文件	产品范围
EllisMILicensingResources.en-US.resx	用于复制保护/许可证组件的错误字符串
EllisMIRDBResources.en-US.resx	用于数据库连接的错误字符串
EllisMIRDBSpatialResources.en-US.resx	用于数据库连接的错误字符串
EllisMIWindowResources.en-US.resx	用于 GUI 的字符串
EllisProgramResources.en-US.resx	用于文件操作的错误字符串
EllisRasterResources.en-US.resx	用于 WMS 连接和光栅/网格图像的错误字符串
EllisTextFileReaderResources.en-US.resx	用于文件操作的错误字符串
EllisThematicsResources.en-US.resx	用于主题图操作的错误字符串
EllisUtilityResources.en-US.resx	用于文件操作的错误字符串和普通的错误字符串
EllisXMLUtilResources.en-US.resx	用于 XML/GML 操作的错误字符串
strings.en-US.resxGeneric	错误字符串、单位、坐标系、基准面和椭面名称、各种 GUI 字符串、WFS 字符串、序列化和光栅错误。

系统要求

Visual Studio 2012 with Update 4 或 Visual Studio 2013。

如何使用本地化工具包

本地化工具包是一种 Visual Studio 解决方案，您可以将其中的资源字符串转换为任何语言。Visual Studio 解决方案可用于 Visual Studio 2012 with Update 4 或 Visual Studio 2013，也可用于 MSBuild 命令行构建实用程序。要执行此操作，只需打开 Visual Studio 资源编辑器中相应的 .resx 文件并编辑成对的名称/值即可。可以只转换一个 .resx 文件中的一行内容，也可以针对所有程序集转换所有 .resx 文件中的所有行。它只取决于应用程序的要求。

转换的文件名必须带有符合 Microsoft 标准的语言/本地代码标识符。例如：English-U.S. LinearReferencing.resources 项目的异常字符串在名为 LinearReferencingExceptionStrings.en-US.resx 的文件中。有关语言标识符的更多信息，请参阅 Microsoft .NET Framework 库中的 **System.Globalization.CultureInfo** 类。

大多数 .resx 文件都在每个项目下面的 Resources 文件夹中。Windows 和 Windows.Dialog 资源位于 Controls 或 Dialogs 文件夹中。

创建卫星程序集

卫星程序集为可选的独立程序集，它仅包含编译的资源。只要一转换 .resx 字符串，就可以随时创建卫星程序集。右键单击 Visual Studio 中的项目，然后选择“创建”。每个项目将生成一个回收站文件夹和一个包含程序集的语言/本地代码文件夹。程序集的文件命名规则和英文程序集名称相同，都必须带有扩展名 .resources.dll。例如，WebControls 程序集的名称可能为：MapInfo.WebControls.resources.dll。

在运行时，如果已提供语言/本地代码资源 dll，MapXtreme 将使用相应的 dll；而且 .NET Resource Manager 也将进行设置，以查找该卫星程序集。如果找不到本地程序集，则使用英语程序集。

标记程序集

使用本地工具包创建的卫星程序集和以下任何一个强制命名的公共钥匙（包括在工具箱中）一起做标记：

- MapInfo.CoreEngine.Public.snk
- MapInfo.Ogc.Public.snk
- MapInfo.WebControls.Public.snk.

MapInfo.CoreEngine.Public.Snk 是所有程序集的公共密钥，不过 MapInfo.Ogc.resources 和 MapInfo.WebControls.resourcess 除外，它们有自己的公共密钥。如果使用这些公共钥匙标记卫星程序集，则可以使用程序集用于测试和调试。如要使用这些程序集运行 MapXtreme，必须将程序集放入全局程序集缓存 (GAC)。只有已标记的程序集才能放入 GAC。

注册卫星程序集

Visual Studio 解决方案中的输出卫星程序集不能安装到 GAC，也不能测试。在使用公共钥匙创建和标记卫星程序集后，这些卫星程序集必须注册，以便 .NET 运行时允许它们载入 GAC。强制命名实用程序 (sn.exe) 必须在每一个卫星程序集上运行，以注册程序集用于跳过验证。此命令为：sn.exe -Vr <assembly_name>。卫星程序集一旦注册可跳过验证后，即可进行测试。

从命令行进行构建

MapXtreme 本地工具箱提供两个 MSBuild 项目文件，它们可以简化准备卫星程序集用于测试的过程。MSBuild 项目文件将创建、标记、注册程序集以跳过验证，并会将它们安装到全局程序集缓存。有关 MSBuild 及其使用方法的更多信息，请访问 [MSDN MSBuild Reference 页面](#)。

(<http://msdn.microsoft.com/en-us/library/0k6kkbsd.aspx>。)

这两个 MSBuild 项目文件分别是 *LocalizationKit.proj* 和 *LocalizationKit.Utilities.proj*。

LocalizationKit.proj

LocalizationKit.proj 目标为：

build — 创建 LocalizationKit 解决方案并调用 LocalizationKit.Utilities 项目的 "RegisterSkipVerification" 目标（默认目标）

rebuild — 重新创建 LocalizationKit 解决方案并调用 LocalizationKit.Utilities 项目的 "RegisterSkipVerification" 目标

clean - 删除所有输出卫星程序集和创建过程中产生的任何临时文件

LocalizationKit.Utilities.proj

LocalizationKit.Utilities.proj 目标为：

RegisterSkipVerification — 通过运行 `sn.exe -Vr <assembly_name>` 命令，注册所有卫星程序集以跳过验证（默认目标）

UnregisterSkipVerification — 通过运行 `sn.exe -Vu <assembly_name>` 命令，取消所有卫星程序集的注册，使其不能跳过验证

InstallGac — 通过运行 `gacutil.exe /I <assembly_name> /f` 命令，将卫星程序集安装到全局程序集缓存

UninstallGac — 通过运行 `gacutil.exe /uf <assembly_name>` 命令，取消安装卫星程序集，使其不能安装到全局程序集缓存

项目配置属性可接受的值为调试（默认）和发行版。使用方法请参阅以下内容。

LocalizationKit 项目文件取决于 LocalizationKit.Utilities 项目文件中的目标，而且它也将使用这些目标。

-
- i** 开发人员如使用 Visual C# Express Edition，则必须将 MSBuild、包含“强制命令实用程序”和“全局程序集缓存实用程序”的目录添加到系统路径。
-

假定有以下路径位置：

MSBuild (msbuild.exe) — C:\WINDOWS\Microsoft.NET\Framework\v3.5

强命名实用程序 (sn.exe) 和全局程序集缓存实用程序 (gacutil.exe) — C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin

，命令提示符中的行将为：

```
LocalizationKit> set  
path=%path%;C:\WINDOWS\Microsoft.NET\Framework\v3.5;C:\Program  
Files\Microsoft SDKs\Windows\v6.0A\bin
```

计算机中的目录路径可能会有不同，所以请务必在您的环境中使用正确路径。

要创建本地工具箱，请运行工具箱根目录中的此命令：

```
LocalizationKit> msbuild LocalizationKit.proj /target:build  
/property:configuration=release
```

此命令创建卫星程序集的发行版（无调试信息），针对每个程序集标记相应的公共强制命名钥匙文件、注册卫星程序集以跳过验证，然后将卫星程序集安装到全局程序集缓存。

卫星程序集即可进行测试。

卫星程序集的私有钥匙标记

当您准备好将本地化程序集放入应用程序中时，它们必须使用 Pitney Bowes Software 提供的私有钥匙做标记。有关此过程的详情信息，请联系您的技术支持代表。

L

术语表

本术语表定义本指南和 MapInfo 产品中有必要理解的术语，以及用于 MapInfo 产品和技术的特定术语。

术语

修饰 (Adornment)

由图例、标题或比例栏组成的 MapXtreme 地图元素。

仿射变换

几何对象的线性变换，例如：旋转、缩放或剪切，并伴有该变换中的转换。用于在 GIS 中将地图从一个坐标系变换到另一个坐标系。

抗锯齿 (Anti-aliasing)

用于在以较低分辨率呈现高清晰度转换时对线和曲线的锯齿状边缘及填充区域的边缘进行平滑处理。

笛卡儿 (Cartesian)

一种坐标系，使用与任何真实世界系统都无关的 x、y 比例。大多数 CAD 绘图对于注册对象（例如，滚珠轴承组件或建筑平面绘图）使用该方法。如果绘图使用笛卡儿坐标，则绘图的一个角的坐标大致为 0, 0。

笛卡儿坐标 (Cartesian Coordinates)

平面上几何体对象通过 x 和 y 值的常规表示。

质心 (Centroid)

通常是地图对象的中心。对于大多数地图对象，质心位于对象的中间位置（对象南北方向长度中点和东西方向长度中点的位置）。在某些情况下，因为有质心必须位于对象自身上的限制，所以质心可能不在中点位置。因此，对于新月形区域对象的情况，实际上对象的中点可能位于限制区域之外；不过，本质心始终应该在限制区域内。

字符编码 (Character Encoding)

一种将序列字节转换为序列字符的方法。另请参阅 [通用字符集 \(UCS\)](#) 和 [Unicode 转化格式-8 \(UTF-8\)](#)。

类

在面向对象语言中，类是包含执行某些函数类型方法的对象或对象集，在含义上类似于过程语言的派生类型。

Codespace

请参阅 [MapInfo Codespace](#)。

凸多边形缓冲区 (Convex Hull Buffer)

一种缓冲区类型，创建区域对象，表示基于输入对象节点的多边形。可以认为该凸多边形是在所有点周围放置橡皮圈的一个操作符。它由最小数目的点组成以便使所有点都位于多边形上或多边形内。使用凸多边形缓冲区，就没有大于 180 度的内角。

COM+ 对象池 (COM+ Pooling)

是一种 Microsoft 组件服务，用其可预装和共享对象，以节省资源。

坐标 (Coordinate)

笛卡儿坐标系的 x、y 位置，或者地球坐标系的经度、纬度位置。坐标表示地图上相对于其它位置的位置。地球坐标系可以使用赤道和格林威治本初子午线作为固定参考点。平面坐标系根据到固定参考点的距离描述二维 x、y 位置，通常使用第一象限，所以所有坐标值都是正数。

坐标系统

坐标系用于创建几何体对象的数字表示。几何体对象的每个点由一对数字表示。这些数字是点的坐标。在绘图学中，坐标系与投影密切相关。可以通过提供投影参数的特定值来创建坐标系。

数据提供程序

应用程序和数据源之间的桥梁，提供用于在应用程序中访问数据的机制。

数据绑定 (Data Binding)

数据源和服务器控件的关联。MapXtreme DataBinding 类包含关于 ASP.NET 服务器控件中单一数据绑定表达式的信息，该控件使 Visual Studio 的快速应用程序开发 (RAD) 设计器可以在设计阶段创建数据绑定表达式。

小数度 (Decimal Degree)

度的分数部分的小数表示。很多地图以度、分、秒为单位（例如，40_30i10l）表示坐标，其中分和秒就是度的分数部分。30 分等于半度，30 秒等于半分。MapXtreme 以小数度表示坐标（例如，72.558 度）。因此，经度：40 度 30 分，在 MapXtreme 中将表示为 40.5 度。

经度的度，纬度的度，小数度 (Degrees Longitude, Degrees Latitude, Decimal Degrees)

经度的度和纬度的度是用于表示地球表面位置的坐标。经度，或 X 坐标，表示位置的东西位置，任何在本初子午线以西位置的 X 值都为负。纬度，或 Y 坐标，表示位置的南北位置，任何在赤道以南位置的 Y 值都为负。

派生的类 (Derived Class)

包含其基类所有特性，且包含关于基类的附加功能或增强功能的类。

枚举 (Enumerate)

包含所有变量和它们可能的值的数据类型。

事件处理器 (Event Handler)

页面对象的属性，可以使用 JavaScript 或 VBScript 编写。例如，事件句柄描述用户单击按钮或选中列表框中的文本时要执行的操作。VBScript 和 JavaScript 都支持显式定义的事件句柄，例如在单击和选择中。此外，可以定义函数替换显式事件句柄。这样的函数被称为隐式事件句柄。

Feature

表中具有几何体、样式和属性的行。Feature 通常具有 Table 和 Key 来标识所表示的行。

FeatureLayer

显示表中图元的 MapXtreme 图层。例如表示世界国家的区域对象的图层是 FeatureLayer。FeatureLayer 必须通过 Map 的图层集合添加到 Map。FeatureLayers 可以是本地的 .TAB 数据、远程 RDB、无缝或光栅数据。

地理编码 (Geocode)

将 X 和 Y 坐标分配到表或数据库记录的过程，以便记录可以显示为地图上的对象。

GeoDictionary

包含与表有关的信息的 MapXtreme 文件（仅 TAB 文件）。GeoDictionary 用于自动确定应该将应用程序绑定到的表。

GeoDictionaryManager

维护 Geodictionary 的 MapXtreme 工具。

地理信息系统 (GIS)

计算机硬件和软件的有条理的集合，专为有效创建、操控、分析和显示所有地理类型或空间参考数据而设计。

几何质心 (Geometric Centroid)

不需要包含在对象内部的中心点（请参阅[质心 \(Centroid\)](#)），通常是 FeatureGeometry。

地理标记语言 (GML)

特定于绘制地图的标记语言。GML 正在由 Open GIS Consortium (OGC) 开发，OGC 是开发和推广地理标准的国际组织。

网格图 (Graticule)

显示在地球地图上的水平（纬度）直线和垂直（经度）直线构成的网格，间距为固定的距离（例如，每五度一格，每十五度一格）。用于建立参考框架。

Grid

区域之间数据值的内插值。网格从数据文件中创建，该数据文件中的数据以平均间距的点度量。整个地图区域转换为网格，其中每个网格单元表示一个值。请参阅[第 17 章：网格图像](#)。

Hillshading

根据虚拟光源而形成的网格地图的辅助底纹。每个网格单元的亮度与照射在表面上的光线一致，且亮度可以根据对光源的方向来调整。

超文本标记语言 (HTML)

纯文本 (Ascii) 语言，使开发人员可以创建 Web 页面，这些 Web 页面可以在不同的计算机平台上，通过不同的 Web 浏览器浏览。Html 使用标记以指定文档各部分的结构。Html 支持链接（使用 URL）指向其它 web 文档或文件。

超文本传输协议 (HTTP)

在 Web 客户端和 Web 服务器之间基于消息的网络接口。HTTP 运行于 TCP/IP 之上。

反射

网格地图的点，网格地图上的颜色会根据网格值或其值的百分比而变化。请参阅[第 17 章：网格图像和变化](#)。

Internet 信息服务 (IIS)

Microsoft 提供的软件服务，支持 web 站点的创建，配置和管理。具体而言，某些辅助的 Internet 信息服务包括：FTP（文件传输协议）和 SMTP（简单邮件传输协议）。在 MapXtreme 中，本公司提供的 WMS Server 必须配置为与 IIS 配合工作以运行服务器。

I Session 接口 (I Session Interface)

MapXtreme MapInfo.Engine 命名空间接口，提供所有 MapXtreme 应用程序的起点。I Session 管理 MapXtreme 应用程序所需的资源的初始化，并定义所有适用于程序实例的数据和功能。

纬度 (Latitude)

地图上的水平直线，从赤道的 0 度到北极点 (+ 90.0 度) 和南极点 (-90.0 度)。用于描述点的南-北位置，通常以在赤道上方或下方的度或小数度为单位度量。

Layer

MapInfo 产品中显示的地图基本组件，通常由多个有层次的图层组成（例如，街道数据图层叠放在郡的图层或邮政编码边界之上）。表在 Map 窗口出现时，它占用了该 Map 窗口的一个图层。通常，每个地图图层对应一个打开的表。

线性参考

相对于传统坐标参考系的另一个参考系，该参考系可将线性图元的位置连接到地球上的点。作为线性网络的部分，任何可用地图表示的实际资产都可保存为数据，该数据说明资产或与资产相关的情况或事件。在 MapXtreme 中，数据将在带有该位置的 X 和 Y 坐标的 MultiCurve 对象上存储为 M 或测量值。为更好的资源管理，可进一步地图化和分析 M 值。请参阅 [第 21 章：线性参考](#)。

经度 (Longitude)

地图上的垂直直线，从北极点延伸到南极点，用于描述点的东-西位置。该位置以本初子午线 (0 度) 以东 (至 -180.0 度) 或以西 (至 +180.0 度) 的度数表示。经度的直线在赤道分开最远而在极点相交，所以它们是不平行的。

经度 / 纬度 (Longitude/Latitude)

MapInfo 产品的地图中表示地理对象的默认坐标系。

MapControl

MapXtreme 对象，使用户可以在窗体上查看地图。MapControl 拥有地图绘制的目的窗口。它还控制地图的大小并与地图工具交互。MapXtreme 提供 MapControl 的桌面和 web 版本。

MapInfo Codespace

定义和标准的列表，常用于创建 MapInfo 地图和工作空间。MapInfo Codespace 包括坐标系设置；画笔、画刷、距离设置和缩写；图像大小设置；经常使用的类型和它们的缩写；可用的操作符的列表；时间、日期、温度单位设置和缩写。有关详细信息，请参阅 [附录 G：定义 MapInfo Codespace](#)。

MapInfo MapCatalog

包含有关空间表的列信息的服务器表。请参阅 [MapInfo_MapCatalog](#)。

MapInfo SQL 语言

用于 MapInfo 绘图产品的 SQL 语法的参考。该语言基于 SQL3，具有为空间分析而定义的特殊 MapInfo 运算符。

子午线 (Meridian)

从北极点延伸到南极点的直线或直线的一部分。子午线是经线。

命名空间 (Namespace)

具有分层结构的命名系统，用于将独立继承的类进行分组。例如，两个不相关的同名类可以在不同的命名空间中存在：System.Utilities.FileFinder 和 MyCompany.Utilities.FileFinder 可以有相同的名称，但是功能不同。此外，命名空间有助于防止编译器引用错误的类（即“冲突”）。

Non-Earth 地图 (Non-Earth Map)

一种地图，其中的对象没有显式引用地球表面的位置。建筑平面图是典型的示例。

永久性

永久性是指 MapXtreme（和其它 MapInfo 产品）管理数据的方法并确保使用该 API 创建的地图可以被其它 MapXtreme 用户使用。永久化和基于 XML 的工作空间的加载和保存有关，用于将 MapInfo Geometry 分析和发布到 GML，反之亦然。

PointRef 唯一 (PointRef Schema)

可以应用于不可制图的表，使其可以绘制地图的空间唯一。该构架通过将不可制图表的列 (MatchColumn) 中的值和可制图表的列 (RefColumn) 中的值进行匹配，引用了另一个表的 Geometry 对象。表打开时，它包含只读 Geometry 列。然后表可以作为图层添加到 Map。

对象池 (Pooling)

用于共享资源以提高性能和可伸缩性。在 MapXtreme Web 应用程序中，可将 MapXtreme Session 实例用在 COM+ 池中，可用其处理客户端的请求。

投影 (Projection)

将地球表面图元位置转换为二维表面位置（例如图纸地图）的数学模型。由于地图尝试在平面上表示球面对象（地球），所以所有投影都在一定程度上失真。地图投影可以保持面积、距离、形状或方向，但只有球形可以保持以上所有属性。某些投影（例如，Mercator）生成非常适合于导航的地图。其它投影（例如 Lambert 一类的等面积投影）生成非常适合于可视化分析的地图。

区域

区域是带有一个外部 Ring 和零个或多个内部 Ring（洞）的 MultiPolygon。

序列化 (Serialization)

序列化是将对象转换为数据流的过程，从而将对象保留在服务器上。该过程是在 MapXtreme web 应用程序中维护对象的基本部分。如果对象没有维护，则服务器需要为每个 web 请求重建对象（例如地图）。

空间唯一 (Spatial Schema)

一种可以应用到表来增强表的空间能力的服务。MapXtreme 中有如下两种类型的空间构架：PointRef 和 XY。包含可以引用可制图表列的列，或包含表示 XY 值的列的不可制图的表，可以使用上述唯一以创建 Geometry 列。然后这些表可以作为图层添加到 Map。关于每种方案类型的详细信息，请参阅 PointRef 方案和 XY 方案术语表定义。

状态管理 (State Management)

是 Web 应用程序开发中的常见术语，与通过浏览器会话保存和恢复信息有关。

表

以行和列格式组织的数据集合。在 MapXtreme 中，表包含要显示在地图上的数据。表保存描述图元的信息，包括它们的几何体、样式和属性。MapXtreme 支持各种数据源中的表，包括本机表 (MapInfo .TAB)、关系数据管理系统 (RDBMS)、dBase、MS Access、ASCII 和 ESRI ShapeFile。特性表包括光栅、网格、无缝、视图、WMS 和 ADO.NET。可通过 TableInfo 类查看表的类型。可以通过 Data 命名空间中的 Catalog 打开或关闭表。请参阅 第 8 章：使用数据。

地图瓦片处理

HTTP 处理器，可处理 Web 应用程序中地图平铺的请求。请参阅 MapXtreme 瓦块句柄。

统一资源定位符 (URL)

超文本链接或图像地图的基础实现，包含万维网上 Web 页面或文件的地址。URL 包含要使用的网络协议信息（通常是 HTTP）和页面或文件的路径。URL 的示例如：

“<http://www.mycompany.com/index.html>”，指向“my company” web 站点的索引页面。

通用字符集 (UCS)

国际标准 ISO 10646 定义了通用字符集 (UCS)。UCS 是所有其它字符集标准的超集。UCS 也定义了多种将字符串编码为字节序列的方法，例如 UTF-8 和 UTF-16。

Unicode 转化格式 -8 (UTF-8)

Unicode 字符的 8 位无损编码。MapXtreme 只按工作空间永久性构架指示的方式支持 UTF-8。

Web 控件 (Web Controls)

是用于与用户进行交互的网页中的元素，用于向 Web 服务器发送请求。

Web 地图服务 (WMS)

MS 是符合 OGC 标准的 Web 服务，提供在绘制地图的应用程序中用作图层的地图图像。如果需要管理 WMS 服务器和客户端以访问任意的符合 OGC 标准 (1.0.0、1.1.0 或 1.1.1) 的 WMS，则 MapXtreme 将提供 WMS 的服务器实现。

Web 图元服务 (WFS)

符合 OGC 的 Web 服务提供的地理参考地图图元可用于地图绘制应用程序。MapXtreme 提供 WMS Basic 的实现，包括只读服务和访问 WMS 服务器的客户端。

Web 服务器 (Web server)

运行超文本传输协议和 Web 服务器软件的计算机系统。Web 服务器接受来自 Web 用户浏览器的基于 URL 的 HTTP 请求，并将 HTML 页面发送回浏览器。一台 Web 服务器可以管理一个或多个 Web 站点。例如商用服务器，通常同时管理多个 Web 站点。

工作空间 (Workspace)

基于 XML 的永久性文件格式，使 MapXtreme 的用户可以在各种环境下共享已经创建的地图。这是未来 MapInfo 所有产品都将符合的文件格式。关于创建工作空间的详细信息，请参阅 [第 24 章：Workspace 管理器](#)。关于工作空间结构的详细信息，请参阅 [附录 C：理解 MapInfo 工作空间](#)。

XY 构架 (XY Schema)

可以应用于不可制图的表，以使其可以绘制地图的空间唯一。该表必须包含 X 和 Y 坐标值，该构架访问坐标值以创建表的 Geometry 列。

索引

- 数字**
- 32 位**
 - 支持的体系结构 30
- 64 位**
 - 支持的体系结构 30
- A**
- AddressCandidates 类 324**
- ADO.NET**
 - 支持的表类型 154
- ADO.NET 接口**
 - 分析数据 184
- ADO.NET 序列化 166**
- ADRG 文件**
 - 支持的光栅格式 301
- AJAXDemo**
 - 示例应用程序 90, 91
- AppStateManager**
 - 实现 113, 114
- ASCII**
 - 支持的表类型 153
- ASP.NET AJAX 90**
 - Web 控件 90, 91
- ASP.NET 应用程序**
 - 创建但不使用 MapXtreme 模板 50
 - 会话管理 189
 - 使用 ASP.NET 模板 48, 49
 - Web 控件 50
- ASRP**
 - 支持的光栅格式 301
- AvoidList 类 342**
- Azimuthal 投影 557**
- 安装**
 - 程序 40, 41, 42, 43
 - 类型 31
 - 先决条件 38
- 安装程序**
 - 部署应用程序 52, 56
- 安装的位图图像**
 - 访问 280
- B**
- BaseGeocodeMatchCode 类 324**
- .bil 文件**
 - 支持的光栅格式 301
- bitmap symbols 530**
- BitmapPointStyleRepository 类**
 - 定制 457
- .bmp 文件**
- 支持的光栅格式 301**
- 半径工具**
 - 桌面 136
- 半透明**
 - 打印时 499
 - MapStyleControl 499
 - 在 Workspace 管理器中启用 408, 416, 417
- 背景地图**
 - 定义 100
- 北偏移 557**
- 比例栏**
 - 修饰 240
- 比例因子**
 - Transverse Mercator 投影 556
- 变化**
 - 计算 310
 - 网格 279
 - 用于网格 310
- 变化值**
 - 网格 310
- 表**
 - 别名 148
 - 不可制图的 148, 152, 164, 171, 172, 593, 594
 - 创建临时 MemTable 161
 - 创建临时本地表 160
 - 创建永久本地表 159
 - 打包 157
 - 打开 152
 - 地图绘制术语说明 60, 61
 - 定义 593
 - 关闭 156
 - 关联 169, 170, 171
 - 结果集 170
 - 可扩展数据提供方
 - 可扩展数据提供方 ± 476
 - 可制图 171, 172
 - 列 148, 149
 - MI_Geometry 列 151
 - MI_Key 列 151
 - MI_Style 列 151
 - 目录 152
 - 事件 157
 - 视图 169
 - 添加表达式列 162
 - 支持的类型 153
 - 表达式
 - AddColumns 203

- 标注 204
 标注优先级 413, 414
 布尔值 201
 创建 200, 201, 202, 203, 204
 函数 201
 InfoTip 204
 示例 202, 203, 204
 使用 200
 图元搜索 203
 在缓冲区外选择对象 203
 在缓冲区中选择对象 202
 主题 204
- 标注**
 范围主题 267
 概述 236
 曲线 240, 420, 422, 423
 UML 图 236
 Workspace 管理器 411, 412, 413
 样式 412
 优先级表达式 413, 414
- 标注工具**
 调整曲线标注的位置 423
 手动移除添加的标注 411
 桌面 136
- 标注主题**
 创建 267, 270
- 标准纬线**
 圆锥投影 556
- 布尔表达式** 201
- 不规则三角形网络**
 网格内插器 309
- 部署 web 应用程序** 449
部署安装 31
部署许可证 33
部署应用程序
 安装程序 52, 56
- C**
- CADRG 文件**
 支持的光栅格式 301
- CalculateMissingMeasures**
 确定度量 (M) 值 355
- CandidateAddress 类**
 地理编码地址 324
- CIB**
 支持的光栅格式 301
- codespace**
 定义 536, 537, 538, 539, 540, 589
- codespace 定义** 536, 538
- COM 对象模型**
 .NET 可互操作性 120
- COM+ 池**
- 定义 589
CoordSys 对象
 创建 293
- CreateThemeWizard**
 Windows 对话框 129, 130, 132, 133
- 参数**
 投影 542
- 操作系统**
 支持的 30
- 查询**
 以 QueryDefinition 类定制 453
- 查找地图图元**
 错误的地址元素 259
 街道地址 246, 247
 街道十字路口 246
 位置名称 246
 优化边界表 248
- 产品功能**
 亮点 23
- 产品体系结** 66
- 常用对话框**
 MapXtreme 附带 128
- 常用工具**
 Web 78, 82
 桌面 139
- 常用样式**
 默认样式 279
- 超文本标记语言 (HTML)**
 定义 591
- 超文本传输协议 (HTTP)**
 定义 591
- 程序集**
 创建自源代码 88, 89
 新建 88, 89
 重定向到新的 45
- 池大小**
 配置 108, 117, 118
- 池, 定义** 593
- 处理异常**
 可扩展数据提供方 487
- 创建**
 LineStrings 288
 MapInfo_MapCatalog 221
 MapXtreme 应用程序 431
 Web 应用程序 75, 76, 77, 78, 102, 103, 105,
 106, 107, 109, 435, 436, 437, 439,
 440, 441, 442, 443, 444, 445
 桌面应用程序 120, 121, 134, 432, 433, 434,
 436, 439
- 创建标注主题**
 单值 270
 范围 267
- 创建地图**

- MapFactory 231
- 创建工作空间**
 - 编程方式 471
 - 通过 Workspace 管理器 394
- 创建几何体**
 - 点 286
 - 多边形 290
 - 环 290
 - 可扩展数据提供方 485
 - LineStrings 288
 - MultiPoint 对象 286
 - 曲线 289
- 创建图层**
 - UserDrawLayer 类 454
- 创建网格 308**
- 创建应用程序**
 - 不使用模板 50
 - 在 Visual Studio 中 47, 49
- 创建主题**
 - 饼图 263
 - 单值 268
 - 点密度 271
 - 范围 266
 - 分级符号 262, 263
 - 条形图 264
- 从 MapXtreme 应用程序进行打印 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509
- 错误处理**
 - Web 控件 85
- D**
- Data.Find 命名空间**
 - 概述 68
 - UML 图 249
- Date data type 149**
- DateTime data type 149**
- dBBase**
 - 支持的表类型 153
- DBMS 数据库**
 - 访问远程表 207
 - 将空间表导入至 220
- 打包**
 - Web 应用程序 446, 447, 448
 - 桌面应用程序 441
- 打包表 157**
- .dem 文件**
 - 支持的网格格式 306
- 打开表**
 - 目录 152
- DropMeasures**
 - 确定度量 (M) 值 355
- DTED 文件**
- 支持的网格格式 306
- 单个近似地址匹配项**
 - 地理编码结果代码 333
- 单值标注主题**
 - 创建 270
- 单值主题**
 - 创建 268
- 导出**
 - 地图到图像 230
- 导入**
 - XML 文件的图元 458
- 地理编码**
 - 参考地址 325
 - 地址候选项 323
 - 地址字典 325
 - 定义 322, 590
 - 发送请求 323
 - 概述 64, 324, 326
 - 街道地址 326
 - 街道十字路口 328
 - 结果代码 332, 333, 334
 - 近似匹配 332
 - 示例 326
 - 输入地址 325
 - 显示地图上的数据 64
 - 向应用程序添加地理编码客户端 322
 - 邮政编码 328
 - 约束 329
 - 约束参数选择 329
- 地理编码地址**
 - CandidateAddress 类 324
- 地理编码类 322**
- 地理标记语言 (GML) 359**
 - 定义 591
- 地理信息系统 (GIS)**
 - 定义 591
- 笛卡儿坐标**
 - 定义 589
- 地图**
 - 地图绘制术语和概念 60, 61, 62, 63
 - 制图图例 273, 274
- 地图标题**
 - 修饰 240
- 地图对象**
 - 确定坐标系 295
- 地图工具**
 - 流程图 83
 - 使用 84
 - 体系结构 82
 - 支持 JavaScript 82
- 地图绘制术语和概念 60, 61, 62, 63**
- 地图视图**
 - 视图工具 136

- 地图图例** 273
 对齐框架 274
- 地图瓦片处理** 92
- 地图应用程序模板**
 快速原型开发 120
- 地址**
 用 Find 匹配 246
- 地址范围**
 Find 类结果 250
- 地址号码**
 用 Find 匹配 247
- 地址候选项**
 地理编码响应 323
- 地址字典**
 定制 325
- 点**
 创建 286
- 点到点路径规划** 337
- 点密度主题**
 创建 271
- 点样式**
 BasePointStyle 类 279
 将位图图像用作 280
 在远程空间表中 226
- 定义可制图的表** 216, 217
- 定制**
 对话框 133
 EngineCustomProperties 类 452
 FeatureOverrideStyleModifier 453
 FeatureStyleModifier 453
 GMLFeatureCollection 458
 工具 456
 搜索函 453
 Web 工具 86, 89
 Windows 工具 139
 Windows 控件 455, 456
 Workspace 管理器 459, 460, 461, 462, 463
 WorkSpaceLoader 458
 UserDrawLayer 454
 样式 457
- 定制工具**
 行为 137
- 定制位图**
 点样式 277
- 动态选择** 136
- 东偏移量** 557
- 对话框**
 CreateThemeWizard 129, 130, 132, 133
 定制 133
 添加至应用程序 128
 网格样式 312
- 对象池**
 定义 106
 开发模型 107
 优势 107
 与 InProc 104, 105, 106, 107
- 对象池开发模型**
 保存状态 108
 定义 100
- 对象池应用程序**
 开发模型 106
- 对象模型**
 MapXtreme 和 MapX 572
 命名空间 67, 68, 69
 实现 572
 数据访问 163, 164, 170, 171, 184, 185, 187
 体系结构 67
- 对象主题**
 饼图 263
 分级符号 262, 263
 说明 261
 条形图 264
- 多边形**
 创建 290
- 多点路径规划** 338
- O**
- ECW 文件**
 支持的光栅格式 300
- .emf 文件**
 支持的光栅格式 301
- EnableTranslucency 属性** 408, 499
- Engine 命名空间**
 概述 68
 异常类 198
- Envinsa Location Utility 服务**
 结果代码 324
- EPSG 代码**
 注册到 MapXtreme 295, 296, 298
- ExecuteFeatureCollection** 176
- F**
- FeatureCollection** 179
- FeatureGeometry** 178
- FeatureGeomety.Distance**
 线性参考 356
- FeatureLayer**
 定义 590
- FeatureOverrideStyleModifier**
 定制 453
 覆盖样式 243, 282
 用于网格 312
- FeatureStyleModifier**
 定制 453
 更改图层外观 453

- UML 图 243
修改样式 242
fill patterns 511
fill styles 511
Find 类
 返回的对象 250, 253
 要求 249
反距离加权
 网格内插器 309
范围
 投影中 557
范围标注主题 267
 创建 267
范围主题
 创建 266
 分布类型 266, 267
反序列化
 过程描述 192, 489
 恢复状态 114, 115
放大工具
 示例 83, 84
仿射变换
 定义 589
访问不可制图的表 148, 152, 164, 171, 172
访问不可制图的数据 452
访问可制图的表 172
访问远程表
 不使用 .tab 文件 207
 DBMS 数据库 207
 定义可制图的表 216
 缓存管理 218, 219, 220
 可制图的表 216, 217
 MapInfo_MapCatalog 220, 221
 MemTable 165, 166
 ODBC 连接字符串格式 214
 属性数据 217
 性能 187, 218
 映射到到 X/Y 列 208
 有 .tab 文件 207
 指定样式 226
非地球地图
 定义 592
分布式应用程序
 许可 33
分段信息
 路径规划 348
分级符号主题
 创建 262, 263
分析数据 184
 ADO.NET 接口 184
 MICommand 184
 目录 184, 185
 OGC 查询接口 184
覆盖样式 282
符号样式
 在远程空间表中 226
服务器表
 缓存数据 218
服务器表查询
 定义可制图的表 216, 217
辅助底纹
 请参阅 hillshading
- ## G
- GDI+** 半透明度和抗锯齿 416, 417, 499
GeocodeClientFactory 类 323
GeocodeMatchCode 类 324
GeocodeRequest 类 326
 发送地理编码请求 323
GeocodeResponse 类
 候选地址 323
Geocoding 命名空间
 概述 69
Geocoding World 组件 326
GeocodingConstraints 类 323
GeoDictionary
 定义 590
 实用程序 426, 427, 429
GeoDictionaryManager 429
 定义 590
Geometry 类 284, 285
Geometry 命名空间
 概述 68
 UML 图 285
GeoTIFF 文件
 支持的光栅格式 300
GIF 文件
 支持的光栅格式 301
GML
 请参阅地理标记语言 (GML)
GmlFeatureCollection 类 458
 定制 458
.grc 文件
 支持的光栅格式 301
.grd 文件
 支持的光栅格式 301
 支持的网格格式 306
GridInflectionCalculator 类
 计算变化值和颜色 311
GTOPO30 文件
 支持的网格格式 306
更新现有的应用程序 44
工具
 定制 139, 456
 工具事件 140, 141

- 设计定制 137
- Web 体系结构 82
- Workspace 管理器 400, 402, 403
 - 用于 Web 应用程序 78
 - 用于桌面应用程序 139
 - 桌面命名空间 69
- 工作空间**
 - 打开包含命名资源的 MWS 193, 194
 - 定义 594
 - 加载 108, 117, 118
 - 结构 467, 468, 469
 - 通过编程方式创建 471
 - XML 格式 395
 - 预载 76
 - 重投影图像 302
- 构建**
 - 可扩展数据提供方 476
- 构建桌面应用程序 120, 121, 134**
- 故障诊断**
 - Oracle 应用程序 227
 - SpatialWare 应用程序 227
 - 许可 36, 37
- 关闭表 156**
- 关键字**
 - MapInfo.abb 文件替换 256
- 关联表 169, 170, 171**
- 光标**
 - 可扩展数据提供方 476
- 光栅**
 - 支持的表类型 154
- 光栅处理程序 303, 304**
- 光栅图像**
 - 打印透明 408
 - 添加到地图 302
 - 限制 302
 - 样式 278
 - 支持的文件格式 300
 - 主类 301
 - 坐标系 302
- 光栅重投影 302, 409**
- 规划**
 - 数据访问 171, 184, 187
 - Web 应用 75, 76, 77, 78, 102, 103, 105, 106, 107, 109
 - 桌面应用程序 120, 121, 134
- 规则划应用程序**
 - Web 435, 436, 437, 439, 440
- H**
- hillshading**
 - 定义 591
 - 网格参数 279
- Hotine Mercator 投影**
 - 倾斜方位角 556
- HTTP Session**
 - 定义 100
- 函数**
 - 日期和时间 201
- 合并模块**
 - 创建安装程序 53, 56
- 画笔样式**
 - 在远程空间表中 226
- 画刷样式**
 - 在远程空间表中 226
- 环**
 - 创建 290
- 缓冲区**
 - 在里面选择对象 202
 - 在外面选择对象 203
- 缓存**
 - MapXtreme 瓦块句柄 94
- 缓存管理**
 - 访问远程表 219, 220
 - 远程表 218
- 回调**
 - 可扩展数据提供方 492
- 会话管理 189**
 - ASP.NET 189
 - ISession 接口 189
- 会话实例**
 - 有对象池 107
- 绘制点**
 - 定制位图样式 277
 - MapInfo 3.0 Compatible 字体 280
 - 字体点样式 278
- 绘制矩形**
 - 矩形工具 139
- 绘制区域**
 - 内部样式 278
 - 样式 277
- 绘制椭圆**
 - 椭圆工具 138
- 绘制文本**
 - 样式 280
 - 字体样式 278
- 绘制圆圈**
 - 圆圈工具 138
- 绘制折线**
 - 样式 279
 - 折线 138
- 绘制直线**
 - 直线工具 138

J

- JavaScript**
 Web 工具 82, 86, 89
- ICursor 界面**
 可扩展数据提供方 478
- IDataProvider**
 可扩展数据提供方 477
- IDataSource**
 可扩展数据提供方 482
- IDW**
 参阅反距离加权内插器
- IFeatureAccessor 界面**
 可扩展数据提供方 478
- 几何体**
 坐标系 293, 294, 295, 296, 297
- 几何体对象**
 转换为 FeatureGeometry 对象 289
- 几何体列**
 可制图的表 216
- 几何质心**
 定义 591
- IInterpolator**
 编程界面 310
- IIS**
 定义 591
- IIS 7/8**
 安装要求 38
 开发 Web 应用程序, 使用 38
- IIS 支持**
 7 和 8 30
- InfoTips 139**
- InProc**
 与对象池 104, 105
- InProc 开发模型 104, 105**
 定义 100
 Web 应用程序 104, 106
 与对象池 104, 106
- InteriorStyleRepository 类**
 定制 457
- Internet 信息服务 (IIS)**
 定义 591
- JPEG 文件**
 支持的光栅格式 301
- JPEG2000 文件**
 支持的光栅格式 301
- ISearchResultProcessor 接口 453**
- ISession 接口**
 定义 591
 会话管理 189
 UML 图 189
- ISessionEventHandlers**
 Workspace 管理器扩展 461
- isoChrono**
 驾驶时间 344
- isoChronoPreferences 类 344**
- isoDistance**
 驾驶距离 346
- isoDistancePreferences 类 347**
- isogramPreferences 类 344**
- ITable 界面**
 可扩展数据提供方 478
- ITableDefinition**
 可扩展数据提供方 478
- ITableMetaData 界面**
 可扩展数据提供方 478
- IWorkspaceManagerNotifications**
 事件处理 461
- 技术支持, 联系 28**
- 基准面**
 定义定制 558, 559, 561, 562
 列表 546, 547, 548, 549, 551, 552, 553, 554
 投影 546
 转换坐标 557
- 驾驶导航**
 路径规划 341
- 加载数据**
 MapLoader 232
- 键**
 MI_Key 列 151
- 检索图元 178**
- 教程**
 创建 Web 应用程序 441, 442, 443, 444, 445
 创建桌面应用程序 432, 433, 434, 436, 439
 打包 Web 应用程序 446, 447, 448
 打包桌面应用程序 441
 应用程序 46
- 街道地址**
 Find 结果代码 253
 用 Find 匹配 246
- 街道地址地理编码 326**
- 街道名称**
 用 Find 匹配 246
- 街道十字路口**
 用 Find 匹配 246
- 街道十字路口地理编码 328**
- 街道缩写**
 用 Find 匹配 246, 247
- 结果代码 253, 254**
 地理编码 332, 333, 334
 类 324
- 结果集 170**
 选择语句 170
- 结果属性**
 查找匹配 252, 253
- 近似匹配项**

- Find 类结果 251
- 经度**
 - 定义 592
- 经度 / 纬度**
 - 定义 592
- 经度的度**
 - 定义 590
- 矩形工具**
 - 绘制矩形 139
 - 桌面 136
- 矩阵路径规划** 339
- 居中工具**
 - 桌面 136
- K**
- 开发安装** 31
- 开发环境**
 - 支持的 30
- 开发框架** 30
- 开发模型**
 - InProc 104, 105, 106
 - 有对象池 106
- 开始状态**
 - 定义 101
 - 应用程序 108
- 抗锯齿** 499
 - 打印时 499
 - 定义 589
 - 使用曲线标注 412
 - 在 Workspace 管理器中启用 408, 416, 417
- 可地图化表**
 - 将行添加到 MapCatalog 222, 224, 225
- 可扩展**
 - Workspace 管理器 459
- 可扩展数据提供方**
 - 抽象基类 479
 - 处理异常 487
 - 创建几何体 485
 - 概述 474
 - 构建 476
 - 构建和测试 482
 - 光标 476
 - ICursor 界面 478
 - IDataProvider 界面 477
 - IDataSource 482
 - IFeatureAccessor 界面 478
 - ITable 界面 478
 - ITableDefinition 界面 478
 - ITableMetaData 界面 478
 - 可选界面 481
 - 软件开发工具包 473, 495
 - SimpleFeature 类 479
- 身份验证 492
- 数据提供方 475
- 数据源 475
- 所需组件 477
- 线程安全 495
- 序列化 489
- 样式 486
- 永久性提供方 487
- 坐标系 486
- 可选界面**
 - 可扩展数据提供方 481
- 可制图的表**
 - 使用服务器表查询进行定义 216, 217
 - 制表 171
- 控件**
 - Web 78, 80, 81, 84, 85, 86, 96, 97, 98
- 空间构架**
 - PointRef 171, 172
 - XY 空间 171
- 空间架构**
 - 定义 593
- 库中的类** 457
- 扩展**
 - Workspace 管理器 403
 - 用于 Workspace 管理器 459, 460, 461, 462, 463
- L**
- LabelLayer** 237
- LabelModifier** 237
- LabelProperties** 238
- LabelSource** 237
- LayerControl** 80
 - 创建定制选项卡 455
 - 定制 455
 - 图层节点 455, 456
 - Windows 控件 127
 - Workspace 管理器 404, 405, 406, 411
 - 样式 281
- LayerNodeHelper 类**
 - 定制 455
- LegendControl** 80
- line styles** 525
- LineStrings**
 - 创建 288
- LineStyleRepository 类**
 - 定制 457
- LocateAlong**
 - 线性参考运算 355
- LocateBetween**
 - 线性参考运算 356
- LocateMeasure**

- 线性参考运算 355
- 来自多个候选匹配项的最佳匹配**
- 地理编码结果代码 333
- 类**
 - 定义 589
- 联系技术支持 28**
- 列**
 - 添加 203
 - 添加表达式 162
 - 支持的数据类型 148, 149
- 列提示 568, 569, 570**
- 浏览器**
 - 支持的 30
- 浏览器会话**
 - 定义 100
- 路径规划**
 - 避免点、图元和分段 342
 - 参数选择 340
 - 点到点 337
 - 定义 336
 - 多点 338
 - 返回分段信息 348
 - 概述 64
 - 基于时间 342
 - 驾驶导航 341
 - 将路径规划客户端添加到应用程序 336
 - 矩阵 339
 - 开始和结束时间 343
 - 瞬态更新 348
 - 停止时间 343
- 路径几何体 342**
- M 类别匹配项**
 - 地理编码结果代码 333
- Map 类**
 - 地图内容 231
- MapAlias 属性**
 - Web 控件 85
- MapCatalog**
 - 定义 592
- MapControl 80**
 - 定义 592
 - Web 控件 76
 - Windows 控件 123, 232
 - 在表单元格中 98
- MapControlModel**
 - Web 应用程序 83
- MapExport**
 - 导出到图像 230
- MapFactory**
 - 创建地图 231
- MapForm 应用程序**
 - 使用 MapForm 模板 46, 47, 48
 - Windows 控件 50
- MapInfo 3.0 Compatible 字体**
 - 绘制点 280
- MapInfo 3.0 兼容字体**
 - VectorSymbolRepository 类 280
- MapInfo ADO.NET 数据提供方**
 - MICommand 175, 176
 - MIConnection 174
 - 数据访问 174
 - 数据提供方 163
- MapInfo Codespace**
 - 定义 592
- MapInfo codespace 定义 536, 537, 538, 539, 540**
- MapInfo MapCatalog**
 - 定义 592
- MapInfo SQL 函数**
 - 表达式中 201
 - 日期和时间 201
- MapInfo SQL 语言 177**
 - 参考 185
 - 定义 592
 - MIDataReader 176
- MapInfo Workspace 格式 (.MWS) 394**
- MapInfo 网格文件**
 - 支持的网格格式 306
- MapInfo.Engine.Session.State**
 - 定义 101
- MapInfo_MapCatalog**
 - 创建 221
 - 添加行 222, 224, 225, 226
- MapInfoCoordinateSystemSet.xml 文件**
 - 投影 544
 - 投影 ID 编号 543
 - 坐标单位 555
- Mapinfo.abb 文件**
 - 编辑 254, 256, 259
 - 替换方法 256, 257, 258
 - 替换关键字 256
 - 与街道缩写匹配 246, 247
- MapLoader**
 - 加载数据 232
- MapMarker 服务器**
 - 结果代码 324
- Mapping 命名空间**
 - 概述 68, 230
 - UML 图 230
- Mapping.Legends 命名空间**
 - 概述 68
- Mapping.Thematics 命名空间**
 - 概述 68
 - UML 图 261
- MapPrinting**
 - 类 501
- MapStyleControl 499**

- MapToolBar**
Windows 控件 125, 126
- MapTools**
桌面应用程序 134
- MapView**
视图列表 232
- MapX 5.0**
迁移到 MapXtreme 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581
- MapXtreme**
安装 40
产品概述 23
创建应用程序 431
对象模型 67, 68, 69, 572
功能概述 23
迁移到 25
升级 43, 45
体系结构概述 66
Web 控件体系结构 81
支持资源 27
- MapXtreme ASP.NET Web 应用程序**
构建但不使用模板 50
使用模板 48
- MapXtreme JavaScript Web 控件**
从 MapXtreme 回传控件迁移 96
- MapXtreme Session**
定义 100
- MapXtreme Session 实例**
干净 115
说明 116
- MapXtreme Windows 地图应用程序**
使用模板 46
- MapXtreme 瓦块句柄 92**
缓存 94
请求可用地图 93
使用 92
- MatrixRouteRequest 类 339**
- MatrixRoutingPreferences 类 340**
- MDAC 2.8**
支持的工具 30
- MemTable**
支持的表类型 154
- MI_Geometry 列 151**
- MI_Key 列 151**
- MI_Style 列 151**
- MICommand 176**
分析数据 184
SQL 语句 175
数据访问 174
- MICollection**
MapInfo ADO.NET 数据提供方 174
数据访问 174
- Microsoft Data Access Components (MDAC)**
- 部署应用程序 56
- MIDataReader 176**
- MrSID 文件**
支持的光栅格式 300
- MS Access**
支持的表类型 154
- MSI 安装程序**
创建安装程序 53, 56
- MultiPoint 对象**
创建 286
- MWS**
具有命名资源的工作空间 193, 194
- .mws**
MapInfo Workspace 格式 394
- MXTRunNCP.exe**
请参阅运行时安装程序
- ## M
- 每记录样式 225, 226**
- 枚举**
定义 590
- 命名空间 67, 68, 69**
定义 592
- 命名连接**
类型 397
- 命名资源**
打开和保存工作空间 193, 194
- 模板**
ASP.NET 48, 49
MapForm 46, 47, 48
用于 MapXtreme 项目 121
- 默认样式**
组合样式 278
- 目录**
表的注册表 152
分析数据 184, 185
数据访问 163, 164
搜索方法 180
- ## N
- NAD 83 基准面 557**
- NADCON**
在 NAD 27 和 NAD 83 之间转换 557
- National Transformation v.2 (NTv2) 563, 564, 565**
- .NET 对象模型**
COM 可互操作性 120
- NITF**
支持的光栅格式 301
- North American Datum Conversion (NADCON)**
请参阅 NADCON
- 内部样式**

- BaseInterior 类 279
绘制区域 278
SimpleInterior 278
- ODBC 连接字符串格式 214**
- OGC 查询接口**
分析数据 184
- OpenGIS 实现规范**
Web 地图服务 377
Web 图元服务 359
- Oracle 空间**
几何体转换 208, 209, 210
连接字符串格式 215
- Oracle 应用程序**
故障诊断 227, 228
- P**
 - PCX 文件**
支持的光栅格式 301
 - pen styles 525**
 - PerpendicularOffset**
线性参考运算 356
 - Persistence 命名空间**
概述 68
 - Photoshop 文件**
支持的光栅格式 301
 - PNG 文件**
支持的光栅格式 301
 - point styles 525**
 - PointExclude 类 342**
 - PointPercentageSpeedUpdate 类 351**
 - PointRef 构架 171, 172**
定义 593
 - PointRelativeSpeedUpdate 类 350**
 - PointSpeedUpdate 类 350**
 - Polyconic 投影 557**
 - 派生的类**
定义 590
 - 匹配**
地理编码 332
 - 匹配数据**
GeoDictionary 171
自动匹配 171
 - 平移工具**
桌面 136
- Q**
 - QueryDefinition**
定制 453
 - QueryFilters**
定制 453
 - 迁移到 MapXtreme 25, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581**
- 请求 / 响应周期**
Web 工具 83
- 倾斜方位角**
Hotine Oblique Mercator 投影 556
- 曲线**
创建 289
- 曲线标注 240, 420, 422, 423**
- 曲线顺序**
在线性参考运算中 356
- 确定度量 (M) 值**
CalculateMissingMeasures 355
DropMeasures 355
ScaleMeasures 355
SetMeasures 355
SetMeasuresAsDistance 355
TranslateMeasures 355
- R**
 - Raster 命名空间**
概述 69
UML 图 300
 - RDBMS 服务器**
支持的表类型 154
 - ResultSet**
支持的表类型 155
 - RoadTypePercentageSpeedUpdate 类 352**
 - RoadTypeRelativeSpeedUpdate 类 351**
 - RoadTypeSpeedUpdate 类 351**
 - RouteInstructionsRequest 类 341**
 - RouteRequest 类 338**
 - RouteSegmentDataRequest 类 348**
 - Routing 命名空间**
概述 70, 336
UML 图 337
 - RoutingPreferences 类 340**
 - 日期和时间表达式 204**
 - 日期和时间函数 201**
- S**
 - S 类别匹配项**
地理编码结果代码 333
 - ScaleMeasures**
确定度量 (M) 值 355
 - SDK**
可扩展数据提供方 473
 - SDK 安装 31**
 - SDK 许可证 32**
 - Seamless**
支持的表类型 155
 - Search 函数 179**
 - SearchResultProcessor 类**
定制 453

- SegmentExclude** 类 342
SegmentPercentageSpeedUpdate 类 349
SegmentRelativeSpeedUpdate 类 349
SegmentRoadTypeUpdate 类 349
SegmentSpeedUpdate 类 349
Selection 类 194, 195
sessionState
 定义 101
SetMeasures
 确定度量 (M) 值 355
SetMeasuresAsDistance
 确定度量 (M) 值 355
ShapeFile
 支持的表类型 154
.sid 文件
 支持的光栅格式 300, 301
SimpleFeature
 可扩展数据提供方实用程序类 479
SmoothingMode 属性 499
SpatialWare 应用程序
 故障诊断 227, 228
Spot 文件
 支持的光栅格式 301
SQL 语句
 MICommand 175, 176
SQL 语言
 MapInfo 参考 185
SRID 代码
 注册到 MapXtreme 295, 296, 298
StateManager 类
 定义 101
 实现 113, 114
StyleRepository
 样式存储 280
StyleRepository 类
 访问预定义的样式 280
styles
 lookup tables 510
 supported custom bitmap symbols 530
 supported fill patterns 511
 supported line styles 525
 supported vector symbols 525
Styles 命名空间
 概述 69
 UML 图 276
symbol font sets 525
身份验证
 可扩展数据提供方 492
生成 Web 应用程序 75, 76, 77, 78, 102, 103, 105, 106, 107, 109, 435, 436, 437, 439, 440
升级到 MapXtreme 43, 45
事件
 工具 140, 141
- 事件处理**
 IWorkspaceManagerNotifications 461
- 事件句柄**
 定义 590
 Web 控件 85
- 时间数据类型**
 SQL 表达式 202
- 示例代码**
 Workspace 管理器扩展 462
- 示例应用程序**
 AJAXDemo 90, 91
 MapXtreme 附带 121
 搜索 181
 线性参考 357
- 视图**
 MapView 232
 支持的表类型 155
- 视图表** 169
- 视图工具**
 更改地图视图 136
- 实用程序**
 GeoDictionary 管理器 171, 426, 427, 428, 429, 430
 Workspace 管理器 394, 424
- 试用许可证** 33
- 数据**
 分析 184
 在 MapXtreme 中使用 168
- 数据绑定** 168
 定义 590
 GeoDictionary 中注册表 426
 描述 168
- 数据访问**
 表达式 200, 201, 202, 203, 204
 部署应用程序 56
 DataTable 165
 ExecuteFeatureCollection 176
 GeoDictionary 管理器 171
 规划 163, 164, 170, 171, 184, 185, 187
 IDBCommand 166, 167
 MapInfo ADO.NET 数据提供方 174
 MapInfo SQL 语言 177
 MICommand 175, 176
 MIConnection 174
 MIDataReader 176
 SQL 命令 175, 176
 图元 178
 优化性能 187
 源行 170
- 数据访问工具**
 支持的 30
- 数据库, 支持** 30
- 数据命名空间**

- 概述 67
- UML 图 147
- 数据提供方**
 - 可扩展 473, 495
 - 描述 475
 - 选择 163
- 数据源**
 - 描述 475
 - 支持的 162
- 属性数据**
 - 访问 217
- 双变量主题 272**
- 水印**
 - 使用 SDK 许可证 32
 - 使用试用许可证 33
 - 使用运行时许可证 33
- 瞬态更新**
 - 路径规划 348
- 搜索函数**
 - 定制 453
 - 目录方法 180
 - SearchInfo 181
 - SearchInfoFactory 181
- 搜索示例应用程序 181**
- 缩放工具**
 - 桌面 136
- 缩放图案 504**
- 缩写文件**
 - 编辑 254, 256, 259
 - 替换方法 256, 257, 258
 - 替换关键字 256
- T**
- .TAB**
 - 支持的表类型 153
- TableInfo**
 - 元数据 158
- Targa 文件**
 - 支持的光栅格式 301
- TIFF 文件**
 - 支持的光栅格式 300
- Time data type 149**
- TIN**
 - 参阅不规则三角形网络
- Tools 命名空间**
 - 概述 69
- TranslateMeasures**
 - 确定度量 (M) 值 355
- Transverse Mercator 投影**
 - 比例因子 556
- TravelTime 类 343**
- TrueType 字体**
- StyleRepository 类 280**
- 特殊字符**
 - Mapinfo.abb 文件替换 258
- 替换**
 - Mapinfo.abb 文件关键字 256
- 体系结构**
 - 对象模型实现 572
 - Web 应用程序 71, 72
 - 应用程序 70, 71, 72, 73
 - 支持的 30
 - 桌面应用程序 73
- 体系结构概述 66**
- 填充样式**
 - 绘制区域 278
 - 在远程空间表中 227
- 添加**
 - 将对话框添加至应用程序 128
 - 将光栅图像添加到地图 302
 - 将几何体对象添加到地图 137
 - 将网格图像添加到地图 307
- 添加工具**
 - 绘制几何体图元 138
 - 将几何体对象添加到地图 137
- 条形图主题**
 - 创建 264
- 统一资源定位符 (URL)**
 - 定义 593
- 通用字符集 (UCS)**
 - 定义 594
- 透明**
 - 应用于地图图元 418, 419, 420
- 投影**
 - 参数 542
 - 出版物和资源 565, 566
 - 地图绘制术语说明 63
 - 定义 593
 - 基准面 546, 547, 548, 549, 551, 552, 553, 554
 - MapInfoCoordinateSystemSet.xml 544
 - MapInfoCoordinateSystemSet.xml ID 编号 543
- 图案缩放 504**
- 图层**
 - 地图绘制术语说明 61
 - 定义 592
 - 定制外观 453
 - 类型 232
 - UML 图 232
 - UserDrawLayer 类 454
 - 应用半透明效果 418, 419
- 图层覆盖样式**
 - 组合样式 278
- 图层控件**
 - Workspace 管理器 404, 407, 408, 409, 410,

- 411, 412, 413, 414
- 图层类型**
- FeatureLayer 233
 - GroupLayer 234
 - MapLayer 233
 - ObjectThemeLayer 234
 - UserDrawLayer 234
- 凸多边形**
- 定义 589
- 图例 273**
- 地图绘制术语说明 61
 - 对齐框架 274
 - 修饰 240
 - 制图 273, 274
- 图元**
- 从表检索 178
 - 地图绘制术语说明 61
 - 定义 590
 - ExecuteFeatureCollection 176
 - FeatureCollection 179
 - FeatureGeometry 178
 - Selection 类 194, 195
 - 搜索 179, 181
 - 修改 179
 - 修饰符主题 261
 - 已描述 178
- 图元几何体样式**
- 可扩展数据提供方 486
- 椭圆**
- 绘制椭圆 138
- W**
- Web 地图服务 (WMS)**
- 定义 594
 - 服务器上图层配置 390, 391, 392
 - 配置 IIS6 的服务器 387
 - 配置 IIS7 的服务器 386
 - 配置服务器 382, 383, 384, 388, 389
 - 通过客户端发送地图请求 380
 - 向服务器发送的请求 377, 379
- Web 服务器**
- 定义 594
- Web 工具**
- 定制 86, 89
 - 使用 84
 - 使用 MapControl 83, 98
 - 说明 80, 81
 - 体系结构 82
 - 与 MapControl 交互 82
 - 与全页回贴 79
 - 支持 javascript 79
- Web 控件 78, 80, 81, 84, 85, 86, 96, 97, 98**
- ASP.NET AJAX 90
- ASP.NET Web 应用程序 50
- 本地化 98
- 从 MapXtreme 回传控件迁移 96
- 错误处理 85
- 定义 594
- 分发 88, 89
- 工具 78
- MapAlias 属性 85
- 目录结构 80
- 事件处理 85
- 使用 80, 84, 85, 86, 96, 97, 98
- 说明 80, 81
- 体系结构 81
- 修改源代码 88, 89
- 在框架中 97
- 状态管理 86
- Web 浏览器**
- 支持的
 - 数据库 $\div \beta \checkmark \mu f$ 30
- Web 图元服务 (WFS)**
- 创建图层 372
 - 定义 594
 - 客户端 369, 370, 371, 372
 - 配置服务器 363, 365, 366, 367
 - 向服务器发送的请求 359, 361, 362, 363
- Web 应用程序**
- 背景地图 75
 - 部署 449
 - 初始请求 115
 - 打包 446, 447, 448
 - 打包教程 446, 447, 448
 - 对象池 78
 - 服务器端命令体系结构 83
 - 更新到 MapXtreme 43, 44
 - 工具 78
 - 规划 75, 76, 77, 78, 100, 101, 102, 103, 104, 105, 106, 107, 109, 435, 436, 437, 439, 440
 - 教程 441, 442, 443, 444, 445
 - 客户端地图交互 82
 - 客户端命令执行 82
 - MapXtreme Session 实例 115, 116
 - 设计考虑因素 75
 - 设计时间视图 77
 - 体系结构 71, 72
 - 添加 ASP.NET AJAX 控件 90, 91
 - 运行时视图 77
 - 术语 100, 101
 - 状态管理 78, 445
- web.config 文件**
- 会话管理 112, 189

- 设置 84
- 预载工作空间 84
- WebControls 命名空间**
 - 概述 69, 78
- VectorPointStyleRepository 类**
 - 定制 458
- Vertical Mapper Classified Grid**
 - 支持的光栅格式 301
- Vertical Mapper Continuous Grid**
 - 支持的光栅格式 301, 306
 - 支持的网格格式 306
- WFS**
 - 请参阅 Web 图元服务 (WFS)
- where 子句 201**
- ViaPoint 类 343**
- Windows 7**
 - 安装要求 38
 - 支持的操作系统 30
- Windows 8 和 8.1**
 - 支持的操作系统 30
- Windows 对话框**
 - CreateThemeWizard 129, 130, 132, 133
 - 定制 133
 - 添加至应用程序 128
- Windows 控件**
 - LayerControl 127
 - MapControl 123
 - MapForm 应用程序 50
 - MapToolBar 125, 126
 - 桌面应用程序 121
- Windows Server 2008**
 - 支持的操作系统 30
- Windows Server 2008 R2**
 - 支持的操作系统 30
- Windows Server 2008/2012**
 - 安装要求 38
- Windows Server 2012 和 2012 R2**
 - 支持的操作系统 30
- Windows 位图文件**
 - 支持的光栅格式 301
- Windows 元文件**
 - 支持的光栅格式 301
- Windows 增强元文件**
 - 支持的光栅格式 301
- Windows.Controls 命名空间**
 - 概述 69
- Windows.Dialogs 命名空间**
 - 概述 69
- Visual Basic .NET**
 - ASP.NET 应用程序模板 48, 49
 - 地图应用程序模板 46
 - 开发框架 30
 - MapForm 应用程序模板 47
- Visual Studio**
 - 创建应用程序 47, 49
 - 开发框架 30
- 瓦块句柄**
 - 缓存 94
 - 请求可用地图 93
- .wmf 文件**
 - 支持的光栅格式 301
- UML 图**
 - apping.Labels 236
 - Data.Find 249
 - 地图绘制 230
 - 几何体 285
 - 路径规划 337
 - Mapping.FeatureStyleModifier 243
 - Mapping.Layers 232
 - Mapping.Thematics.IModifierTheme 261
 - Mapping.Thematics.ObjectTheme 261
 - Raster 300
 - 数据命名空间 147
 - 样式 276
- WMS**
 - 请参阅 Web 地图服务 (WMS)
 - 支持的表类型 155
- Unicode 转化格式 (UTF-8)**
 - 定义 594
 - 在工作空间中受支持 468
- Workspace 管理器**
 - 标注工具 411, 423
 - 菜单命令 395, 397, 398, 399, 402, 403
 - 定制 459, 460, 461, 462, 463
 - 覆盖样式 405
 - 工具菜单命令 400
 - LayerControl 405, 406, 411
 - 另存为 XML 394
 - 命名连接 397
 - 特性摘要 394
 - 图层控件 404, 410, 411, 412, 413, 414
 - 图层控件工具 404, 407, 408, 409
 - 主题 411
 - “扩展”菜单 403
 - “网格样式”对话框 312
- Workspace 管理器扩展**
 - 创建 459
 - 加载 461
 - 示例代码 462
- WorkSpaceLoader 类**
 - 定制 458
- URL 请求**
 - 服务器端 83
 - 客户端 82, 83
 - MapControlModel 83
- UserDrawLayer 类 454**

- USGS DEM 文件**
 - 支持的网格格式 306
- 网格 306**
 - 变化 279
 - 创建 308
 - 定义 591
 - GridInflectionCalculator 类 311
 - 检索数据 307
 - 山体阴影参数 279
 - 添加到地图 307
 - 修改变化值和颜色 310
 - 修改设置 312
 - 样式 278, 310
 - 样式对话框 312
 - 支持的表类型 155
 - 主类 306
- 网格内插器**
 - 编程界面 310
 - 不规则三角形网络 309
 - 反距离加权 309
- 网格图**
 - 定义 591
 - 图层设置 414
 - 在 Workspace 管理器中启用 423, 424
- 纬度**
 - 定义 591
- 纬度的度**
 - 定义 590
- 位图**
 - 创建定制 281
- 位置名称 246**
 - Find 结果代码 254
- 文本**
 - 绘制 278
 - 绘制样式 280
- 文档集 27**
- 文件格式**
 - 光栅图像 300
 - 网格 306
- 无缝光栅图层 405**
- 无匹配代码**
 - 地理编码结果代码 334
- X**
- XML 文件**
 - 导入图元 458
 - GeoDictionary 426, 427, 429
 - Workspace 格式 (.MWS) 394
- XY 构架 171**
 - 定义 594
- 线程安全**
 - 可扩展数据提供方 495
- 线性参考 353, 354, 355, 356, 357**
 - CalculateMissingMeasures 355
 - DropMeasures 355
 - FeatureGeometry.Distance 356
 - LocateAlong method 355
 - LocateBetween method 356
 - LocateMeasure method 355
 - PerpendicularOffset 356
 - 曲线顺序 356
 - 确定度量 (M) 值 355
 - ScaleMeasures 355
 - SetMeasures 355
 - SetMeasuresAsDistance 355
 - 示例应用程序 357
 - TranslateMeasures 355
 - 运算 355
- 线样式**
 - 在远程空间表中 226
- 项目模板**
 - MapXtreme 附带 121
- 小数度**
 - 定义 590
- 性能**
 - 优化数据访问 187
- 修改图元 179**
- 修饰**
 - 地图元素 240
 - 定义 589
- 修饰符样式**
 - 组合样式 278
- 修饰符主题**
 - 单值标注主题 270
 - 单值主题 268
 - 点密度主题 271
 - 范围 266
 - 范围标注主题 267
 - 说明 261
- 许可**
 - 故障诊断 36, 37
 - 已知问题 37
 - 应用程序 32, 33
- 许可证**
 - 获取 33
 - 类型 32, 33, 34
 - 文件位置 34
- 序列化**
 - ADO.NET 166
 - 保存状态 114, 115
 - 定义 593
 - 过程描述 192, 489
 - 恢复状态 115
 - 可扩展数据提供方 489
 - ManualSerializer 114, 115

- 选择对象**
在缓冲区外 203
在缓冲区中 202
- 选择工具**
选择点和区域 136
- 选择语句**
结果集 170
MICommand 175, 176
- Y**
- 样本连接字符串** 215
- 样式**
标注 412
常用 279
创建定制位图 281
点 277, 279
覆盖 243, 282, 405
光栅 278
LayerControl 281
区域 277
矢量点 280
填充 278, 279
Workspace 管理器 412
弯曲 279
网格 278, 310
网格 hillshading
文本 280
修改 242
以库中的类定制 457
在远程空间表中 226
直线 279
字体 278
字体点 278
组合 278
- 样式覆盖**
图层设置 414
- 异常类**
Engine 命名空间 198
- 影响**
定义 591
- 应用程序**
部署 441
打包 Web 446, 447, 448
打包桌面 441
规划 Web 75, 76, 77, 78, 102, 103, 105, 106, 107, 109, 435, 436, 437, 439, 440
规划桌面应用程序 120, 121, 134
开始状态 108
启动工作空间 108, 117, 118
体系结构 70, 71, 72, 73
许可 32, 33
桌面体系结构 73
- 应用程序模板**
ASP.NET 48, 49
MapForm 46, 47, 48
- 应用程序状态**
定义 100
主题示例 115, 116
- 用户创建的安装程序**
部署应用程序 53, 56
- 用户定义的元数据** 568, 569, 570
- 用户状态**
定义 101
管理 110
- 永久性**
保存信息 458
定义 592
过程描述 193
- 永久性提供方**
可扩展数据提供方 487
- 优化边界表**
查找街道地址 248
Find 结果代码 253
- 邮政编码的地理编码** 328
- 邮政编码中心位置匹配**
地理编码结果代码 333
- 预定义的样式** 280
- 远程表**
不使用 .tab 文件 207
DBMS 服务器 207
定义可制图的表 216
访问属性数据 217
缓存管理 218, 219, 220
可制图的表 216, 217
ODBC 连接字符串格式 214
性能 218
映射到到 X/Y 列 208
有 .tab 文件 207
- 远程空间表**
导入至 DBMS 数据库 220
将行添加到 MapCatalog 222, 224, 225, 226
MapInfo_MapCatalog 220, 221
每记录样式 225, 226
样式 226
指定样式 226
- 源代码**
MapXtreme Web 控件 88, 89
- 圆圈工具**
绘制圆圈 138
- 元数据**
TAB 文件 159
TableInfo 158
- 源行** 170
- 原型开发**
快速桌面应用程序开发 120

- 圆锥投影** 初始请求 115, 116
 标准纬线 556
- 约束** 对象池 100, 101
 地理编码 329
- 运算** Web 应用程序规划 100, 101
 线性参考 355
- 运行时安装** 31
运行时安装程序 状态管理 100, 101
 部署应用程序 52
- 运行时许可证** 32, 33
- Z**
- Z 类别匹配项** 定义 100, 101, 593
 地理编码结果代码 333
- 折线** 对象池应用程序 108
 绘制样式 279
- 折线工具** 138
支持 REST 的瓦块句柄 92
- 支持的操作系统** 30
支持的环境 30
 IIS 7 和 8 30
- 支持的体系结构** 30
支持的文件格式 会话对象 189
 光栅图像 300
 网格 306
- 制图图例** 273, 274
直线工具 138
直线样式 手动 108, 113, 114
 BaseLineStyle 类 279
 绘制折线 279
- 质心** 手动配置 110
 定义 589
- 重定向程序集** 45
主题 手动示例 109, 110
 饼图 263
 CreateThemeWizard 129, 130, 132, 133
 单值标注主题 270
 单值主题 268, 270
 地图绘制术语说明 62
 点密度主题 271
 对象 261
 范围 266
 范围标注主题 267
 分级符号 262, 263
 双变量 272
 条形图 264
 图例 273
 Workspace 管理器 399, 411
 修饰符 261
 应用半透明效果 420
 重新计算容器 131
- 主题示例** 图层设置 414
- 术语** 坐标
 定义 589
 在 NAD 27 和 NAD 83 之间转换 557
 转换为另一个基准面 557
- 术语表** 589, 590, 591, 592, 593, 594
状态管理 坐标单位 555
 定义 100, 101
 对象池 100, 101
 Web 应用程序规划 100, 101
- 桌面工具命名空间** 69
桌面控件 121
桌面应用程序 初始请求 115, 116
 创建 432, 433, 434, 436, 439
 打包 441
 更新 44
 规划 120, 121, 134
 会话管理 189
 教程 432, 433, 434, 436, 439
 MapControl 123
 MapTools 134
 体系结构 73
 原型开发 120
- 自动标注** 坐标
 定义 589
 在 NAD 27 和 NAD 83 之间转换 557
 转换为另一个基准面 557
- 字符编码** 坐标单位 555
 定义 589
 工作空间中的 UTF-8 468
- 字体点样式** 地图绘制术语说明 62
 绘制点 278
- 字体样式** 图层设置 414
 绘制文本 278
- 子午线** 坐标
 定义 592
- 组合样式** 278
组图层

坐标系

- 出版物和资源 565
- 地图绘制术语说明 63
- 定义 590
- 光栅图像 302
- 几何体 293, 294, 295, 296, 297
- 可扩展数据提供方 486
- 如何确定 295
- 添加到 MapXtreme 295, 296, 298
- 原点 556, 557
- 注册 EPSG 和 SRID 代码 295, 296, 298
- 注册到 CoordSysFactory 对象 293
- 转换 Geometry 对象 294