**DemoMap：**

DemoMap是WinForm应用程序，使用了MEF的方法来加载组件。

在主窗体MainForm的构造函数中，appManager是AppManager的实例，其中主要的是：

//加载组件

Thread updateThread = new Thread(AppLoadExtensions);

updateThread.Start();

//激活组件

ActivateAllExtensions();

加载组件中AppLoadExtensions()方法包含了MEF的常规配置，配置目录，配置组合容器，以及配置初始化等。

**类库：**

Control：

1. IMapView：规定坐标转换。屏幕点与地理点的相互转换、相对于地图控件的像素坐标中的矩形与地理包络线的相互转换、ViewExtents属性获取或设置要在视图中显示的地理区域。
2. IBasicMap：
3. ISelectable：

bool SelectionEnabled获取或设置是否支持selection

bool ClearSelection(out IEnvelope affectedArea)清除所有选中的要素

bool InvertSelection(IEnvelope tolerant, IEnvelope strict, SelectionMode mode, out IEnvelope affectedArea)反选

bool Select(IEnvelope tolerant, IEnvelope strict, SelectionMode mode, out IEnvelope affectedArea)选中要素

bool UnSelect(IEnvelope tolerant, IEnvelope strict, SelectionMode mode, out IEnvelope affectedArea)从选中的要素中取消选择。

1. Map类：map控件。

方法：

* + - 1. IMapImageLayer[] GetImageLayers()获取特定为Image层的子集，允许控制它们的符号。
      2. IMapRasterLayer[] GetRasterLayers()获取特定为Raster层的子集，允许控制它们的符号
      3. IMapLineLayer[] GetLineLayers()获取只包含line层(而不是普通层)的列表。
      4. IMapPolygonLayer[] GetPolygonLayers()获取只包含Polygon层(而不是普通层)的列表。
      5. IMapPointLayer[] GetPointLayers()获取只包含Point层(而不是普通层)的列表
      6. IMapFeatureLayer[] GetFeatureLayers()获取仅包含Feature层的列表，而不管它们是线条、点还是多边形。
      7. IMapFunction GetMapFunction(string name)获取基于字符串名称的MapFunction。
      8. bool ClearSelection(out IEnvelope affectedArea) 从所选状态中的现有成员中清楚所有选择。并有bool ClearSelection()扩展方法。
      9. 实现ISelectable接口定义的方法。
      10. 添加图层。包括矢量图层和栅格图层
      11. virtual void ResetBuffer()这可以在任何时候调用，并且目前用于捕获在实际数据应该更新时重新调整大小事件。
      12. virtual void SaveLayer()保存图层，矢量栅格均可被保存。调用SaveFileDialog进行保存。
      13. void ZoomToMaxExtent()指示Map更改透视图以包含整个绘图内容
      14. Extent GetMaxExtent(bool expand = false) 获取当前Map的最大范围窗口。
      15. void Print(Graphics device, Rectangle targetRectangle)
      16. void Print(Graphics device, Rectangle targetRectangle, Extent targetEnvelope) 不使用通常的缓冲区，而是绕过任何缓冲，并指示层直接绘制到图形对象上指定的目标矩形
      17. void ClearLayers()指示Map清除图层。
      18. List<ILayer> GetLayers()获取Map中图层。是浅拷贝，获得对象的引用，因此可以访问到各个layer
      19. List<ILayer> GetAllLayers()获取映射的所有层，包括嵌套在组内的层。组对象本身不包括在此列表中，但包括所有功能层、栅格层、图像层和其他层。
      20. List<IMapGroup> GetAllGroups()获取map中的所有地图组，包括嵌套组
      21. override void Refresh()将重新绘制所有的数据图层到缓冲区，而不仅仅是只绘制缓冲区本身
      22. void RefreshMap(Rectangle clipRectangle)重新绘制指定矩形框中的数据到缓冲区
      23. Coordinate PixelToProj(Point position) 将单个点位置转换为等效的地理坐标。
      24. Extent PixelToProj(Rectangle rect) 将相对于地图控件以像素坐标表示的矩形转换为地理包络。
      25. Point ProjToPixel(Coordinate location) 将单个地理位置转换为屏幕上相对于地图左上角的等效点。
      26. Rectangle ProjToPixel(Extent env) 将单个地理包络转换为等效的矩形，因为它将在屏幕上绘制。
      27. void ZoomIn()地图放大
      28. void ZoomOut()地图缩小
      29. void ZoomToNext()下一视图
      30. void ZoomToPrevious()前一视图
      31. Bitmap SnapShot()捕获图像，这个图像包含屏幕大小的后台缓冲区的任何内容。
      32. Bitmap SnapShot(int width) 创建缩放到适合指定宽度的位图的快照。

属性：

1. bool ExtendBuffer指示绘图层是否应该将屏幕外数据缓存到缓冲区，平移漫游功能的流畅性，但缩放、选择和调整大小将导致/性能损失。
2. string ProjectionEsriString获取或设置Map的投影Esri字符串。此属性用于将投影字符串序列化到项目文件。WKT对投影的描述。
3. ActionMode ProjectionModeDefine和ActionMode ProjectionModeReproject获取或设置一个提示模式枚举，该枚举控制用户在添加具有与Map不同的坐标系统的Layer之前是如何被提示的。
4. bool RedrawLayersWhileResizing获取或设置一个值，该值指示层是否应在实际调整大小的过程中绘制。正常行为是在新的大小和位置中绘制现有的图像缓冲区，这对于大型数据集来说要快得多，对于小数据集效果肉眼不好分辨。
5. Cursor Cursor更改光标的指针类型。
6. bool SelectionEnabled获取或设置一个布尔值，该布尔值指示是否能够选择。
7. bool CollectAfterDraw获取或设置一个布尔值，该布尔值指示垃圾收集器是否应在绘图后进行收集。可以禁用快速操作窗格，但通常应该启用。
8. List<IMapFunction> MapFunctions获取或设置此项目中内置的工具字典。有8个：MapFunctionKeyNavigation、MapFunctionPan、MapFunctionSelect、MapFunctionClickZoom、MapFunctionZoomOut、MapFunctionZoom、MapFunctionLabelSelect、MapFunctionIdentify。
9. FunctionMode FunctionMode启用或禁用指定工具。
10. IMapLayerCollection Layers获取Map中图层集合。
11. ProjectionInfo Projection获取或设置投影。这应该反映加载的第一个数据层的投影。随后的加载，但不匹配的投影应该抛出警报，并允许重新投影。
12. bool ZoomOutFartherThanMaxExtent这样可以放大比地图范围更远的范围。
13. ActionMode枚举：描述map function在使用时具有的不同行为，这些行为遵循其他的mapfunctions。Prompt由用户决定图层是否应被重投影，Always总是重新投影以匹配MapFrame投影，Never不投影，PromptOnce只提示一次表示之后的均同意进行重投影。
14. FunctionMode枚举：是默认工具。有：

Info鼠标轮仍然放大地图，但左键将项目带入视图，鼠标变成 “问号”可以进行要素识别。

ZoomIn用鼠标左键放大地图，用鼠标右键缩小。同时可按下滚轮保持不放，进行平移，也可拉框放大。

ZoomOut用鼠标左键缩小地图，用鼠标右键放大。时可按下滚轮保持不放，进行平移，也可拉框缩小。

ZoomPan鼠标滚轮缩放，按住鼠标滚轮平移。

Pan用鼠标左键平移地图，右键打开上下文菜单，用鼠标滚轮缩放地图。

Select鼠标左键选中Shape，右键打开上下文菜单，用鼠标滚轮缩放地图。

Label左键选择、移动或编辑，右键生成上下文菜单。鼠标指针为IBeam。

None禁用所有工具。

Data：

1. AffineTransform类：仿射变换
2. Argb结构体：颜色组件
3. AttributeCache类：属性缓存。记录缓存、恢复数据
4. BgdRaster：ESRI的bgd栅格数据格式。栅格数据的打开，保存，读，写、复制，获取栅格数据类型。
5. BinaryRaster类：构造二进制栅格。复制栅格文件，获取和复制栅格Window、获取统计信息、文件打开、Window打开、粘贴栅格、从指定的行列的起始值读取“Window”的内容、读写文件头、保存
6. BinaryRasterProvider类：根据指定的数据类型打开栅格数据、创建栅格数据。
7. BibaryReaderWriterExtensions类：提供对指定类型读、写的功能。
8. BitDepth枚举：
9. BitmapGrid类：创建空白image网格、创建断开连接副本、清空网格值、释放占用资源、颜色填充、转换为Bitmap、计算栅格图之间的差异、绘制指定像元的颜色值、匹配网格字节值是否相同、随机值替代网格字节值、设置像元颜色
10. BoxStatistics类：将数据分成4个四分位数
11. BufferedBinaryReader类：使shapefile读取更有效，也可以对二进制读取。扩展缓冲区大小、填充缓冲区、快速到达指定位置并开始读取、从缓冲区读取各种数据类型并自动加载、将计数字节拷贝到指定缓冲区索引，并作为起始点。
12. BufferedBinaryWriter类：功能类似BufferedBinaryReader类。
13. ByteBlock类：字节快类。使用指定的reader来读取字节数，包括使用Stream.Read()和BufferedBinaryReader.Read()、获取整形（或double）偏移量、将ByteBlock复制给数组、转换为Int或Double型数组、
14. ColorType枚举：颜色模型
15. CoordinateType枚举：仅X和Y的坐标、带有M和Z值的坐标。
16. CopyList类：对内部项目进行拷贝，如果项目没实现IClonable，则只进行浅拷贝。
17. CRC32类：加密算法。
18. DataFormats枚举：包括Vector（lines、points、polygons（符号化器绘制而成）），Raster（是integers、doubles、floats或其他数值类型数字。可以被符号化或代表images，不能被直接绘制），Image（特指存储色彩值的像元坐标，可被直接绘制），还有自定义数据格式。
19. IDataSet：读写数据集范围、读写数据集名称、获取或设置用于此数据集所采取的内部操作的进度处理程序、获取或设置为此数据集提供的空间和时间的支持、描述在此数据集中元素的类型、关闭数据集（断开与数据源的链接）
20. DataSet类：（protected）判断投影是否受支持、实现IDataSet接口和继承DisposeBase类
21. IExtent：定义范围。MinX（x维最左或最小经度）、MinY、MaxX、Max Y、HasM、HasZ。
22. Extent类：根据不同的参数类别（默认无参数、xy范围、数组+偏移量、数组、包络线）构造Extent、获取Extent的中心坐标、运算符重载实现相等和不等比较运算、判断范围的包含关系、判断范围是否包含某一坐标、判断范围是否包含包络线、从已有范围中拷贝范围、判断范围是否相等、调整（扩展）范围（通过size、padding）、扩展到已有较大范围或根据xy值进行扩展、获取HashCode、计算两个范围的交集、判断坐标（或xy值、节点）是否在范围内、判断范围间是否相交、判断范围和包络线是否相交、判断范围是否为空、解析范围的字符串版本生成范围、设置范围中心点、范围转为包络线（或string）、判断是否在另一个范围（或包络线）内。
23. Vertex结构体：通过double型x和y构造节点。节点转换为坐标、判断与另一节点是否相等、运算符重载运算节点相等或不等（容限内）、获取HashCode。
24. IFeature：定义Feature。DataRow、BasicGeometry、ContentLength、包络线、Fid、parentFeatureSet包含该feature的父要素图层、记录数、shape类型、shapeIndex、深复制、转换为shape、将源feature的属性表复制给实例。
25. ShapType枚举：形状类型。包括：无类型或不可用、点、折线、面、多点、带Z值点、带Z值折线、带Z值面、带Z值多点、带M值点、带M值折线、带M值面、带M值多点（带M值的都带Z值）
26. VectorFileType枚举：CacheType：Cached、Dynamic。VectorFileType：shape file、CSV、GML、MapInfo等等。
27. Feature类：实现IFeature接口。

构造：

* + - * 1. 默认无参构造
        2. 通过Shape参数构造
        3. 通过Vertex参数构造
        4. 通过Coordinate参数构造
        5. 通过IBasicGeometry参数构造
        6. 通过IBasicGeometry、IFeatureSet参数构造（IFeatureSet可看作是feature的容器）
        7. 通过FeatureType、IEnumerable<Coordinate>参数构造

方法：

1. Int IComparable<IFeature>.CompareTo(IFeature other)比较两个feature之间的待比较字段（ComparisonField）的index的大小。
2. IBasicGeometry GetBasicGeometryN(int index)获取index位置的BasicGeometry
3. IFeature IFeature.Copy()返回copy()函数，深拷贝
4. void CopyAttributes(IFeature source) 这将使用字段名将属性值从源复制到此feature。将列明相同的列的值进行复制，同时应确保该feature的ParentFeatureSet不为bull。
5. void UpdateEnvelope()强制几何更新其Envelope，然后更新特性的缓存Envelope。
6. virtual string ExportToGml()创建一个新的gml字符串，用于描述此点的位置。
7. virtual byte[] ToBinary()生成代表Geometry的WKB
8. Shape ToShape()根据这个feature本身创建一个新的shape。

属性：

* + - * 1. CacheTypes NumPartsSource是否缓存
        2. int NumParts不懂表示什么意思？
        3. 实现IFeature定义的属性

1. IFeatureSet：AttributePopulated获取属性是否已全部加载到数据表中、获取或设置整个featureset中坐标类型、获取或设置所关联的指定的feature的Datatable、获取包含在此layer中的要素的集合、获取feature查找表、获取或设置FeatureType、获取该要素图层的文件名、获取或设置ShapeIndices、获取或设置索引模式indexMode、获取或设置节点数组、获取或设置Z或M坐标数组、选择一些索引、

保存由数据源提供的信息到其现有文件位置、另存为、选择Extent内的要素集合、选择Extent内的要素集合，并返回选定区域的地理范围、根据属性选择feature、根据属性选择index、更新范围。

1. FeatureSet类：实现IFeatureSet接口。

构造：

方法：

* + - * 1. void AddFid()添加Fid。如果有“FID”则退出函数，如果没有，先添加“FID”，然后为其添加值（从0开始，根据记录数量依次加1）。
        2. 添加feature（IBasicGeometry）
        3. void AddShape(Shape shape)
        4. void AddShapes(IEnumerable<Shape> shapes)

1. FeatureSetExt类：是FeatureSet的扩展方法。
2. FeatureExt类：是Feature的扩展方法。
3. Fields类：

构造：

1. Fields(IEnumerable<Field> fields) 构造字段并更新各个字段的数据地址

索引器：

1. Field this[int pos] 按顺序检索字段
2. Field this[string name] 按名称检索字段
3. int Count获取字段数

方法：

1. void Add(Field field) 添加字段
2. void UpdateDataAddresses()更新每个字段的数据地址

Topology：

* 1. IRectangle：定义矩形。Height（y值的差）、Width（x值的差）、X（包络线的最小x值）、Y（包络线的最大y值）
  2. IEnvelope：定义包络线。Minimum、Maximum、纵坐标数、判断是否为空、创建包络线副本、设置为空、判断是否有M值、判断是否有Z值。
  3. FeatureType枚举：要素类型。包括：未指定、点、线、面、多点。
  4. IBasicGeometry：定义BasicGeometry。Coordinates、Envelop、几何图形数、点数、几何类型、要素类型、根据索引获得basicGeometry、导出为GML、转换为WKB、转换为WKT、强制使用坐标重新计算包络线。

Symbology：

ILayer：

属性：

* + 1. IDataSet DataSet：获取或设置layer的核心数据集。
    2. Extent InvalidRegion：获取当前无效的范围。
    3. IFrame MapFrame：获取或设置layer的MapFrame。
    4. IProgressHandler ProgressHandler获取或设置进程处理程序。

方法：

* + - 1. bool VisibleAtExtent(Extent geographicExtent) 给定地理范围，这将测试"IsVisible","UseDynamicVisibility","DynamicVisibilityMode"and "DynamicVisibilityWidth"以确定该层是否可见。
      2. void Invalidate(Extent region) 通知该layer，下次指定与此区域相交的区域时，必须首先将内容重新绘制到图像缓冲区。
      3. bool IsWithinLegendSelection()查询此层和整个父树，直到Map Frame，以确定此层是否在所选层中。

**Layer类：**可以表示的不只是一个包含**形状**和**属性数据**的图层，也有可能是Group或者MapFrame。

构造：为protected类型，配置上下文菜单。外部的类中，无法访问。

方法：

1. bool VisibleAtExtent(Extent geographicExtent) 给定地理范围，这将测试"IsVisible","UseDynamicVisibility","DynamicVisibilityMode"and "DynamicVisibilityWidth"以确定该层是否可见。
2. override bool CanReceiveItem(ILegendItem item) 测试指定的图例项。如果该项是另一个层、一个组或一个Map Frame，则返回false。此外，如果该项的父对象也不是此对象，那么它也将返回false。其思想是，层可以让子节点移动，而不是从一个地方传输到另一个地方。
3. bool IsWithinLegendSelection()查询此层和整个父树，直到map frame，以确定该层是否在所选层中。
4. virtual void Invalidate(Extent region) 通知该layer，下次指定与此区域相交的区域时，必须首先将内容重新绘制到图像缓冲区。
5. override void Invalidate()通知父层此项无效，应重新绘制。
6. static ILayer OpenFile(string fileName) 使用默认layer provider打开文件名并返回新的layer。该层不会自动有一个容器或被添加到一个Map。
7. static ILayer OpenFile(string fileName, IProgressHandler progressHandler)作用同上
8. static ILayer OpenFile(string fileName, ICollection<ILayer> container)打开一个新图层，并自动添加进一个容器。
9. virtual ILayer OpenLayer(string fileName, bool inRam, ICollection<ILayer> container, IProgressHandler progressHandler) 对于与字符串形式匹配的Extension的ILayerProvider插件，尝试调用打开文件名方法。
10. void Dispose()释放图层中的内存对象。
11. void LockDispose()锁处理。这通常添加内部引用计数器的一个实例。
12. void UnlockDispose()解锁处理。这通常移除内部引用计数器的一个实例。
13. virtual void Reproject(ProjectionInfo targetProjection)为图层重新投影

属性：

1. override bool IsSelected判断图层是否被选中。这里的图层可以是一个单个图层也可以是该图层的父树，直到map frame。
2. IDataSet DataSet获取或设置内部数据集。这可以是null，如在Groups或MapFrames的情况下。在复制图层时，进行**浅复制。**复制一个层不应该创建数据集的副本，而是应该指向原始数据集。但是DataSet本身是**可以被克隆**的。
3. ILayerActions LayerActions为图层设置或获取自定义操作
4. double DynamicVisibilityWidth动态可见性表示仅在放大到足够近时才出现的层。此值表示发生这种情况的地理宽度。
5. DynamicVisibilityMode DynamicVisibilityMode这控制当放大到更接近DynamicVisibilityWidth时层是否可见，或者仅在进一步远离DynamicVisibilityWidth时才可见。
6. virtual Extent InvalidRegion获取当前已失效的区域。
7. virtual IFrame MapFrame获取父图层集合的MapFrame
8. virtual IProgressHandler ProgressHandler获取或设置此图层的进程处理程序。设置重写的默认行为
9. bool UseDynamicVisibility获取或设置布尔值，指示是否允许动态可见性信封控制可见性。
10. override bool IsVisible获取或设置在Map中的图层是否可见。
11. bool IsDisposed指示内存对象是否已被处理。
12. bool IsDisposeLocked指示现有引用是否请求该对象不被处理。
13. bool CanReproject指示该图层是否可以被重新投影
14. ProjectionInfo Projection获取或设置此层的数据集的投影信息。这仅定义投影信息，而不重新投影dataset或layer。
15. string ProjectionString获取或设置未修改的投影字符串，不管DotSpatial.Projection module是否可用，都可以使用它。此字符串可以是Proj4string格式，也可以是EsriString格式。设置投影字符串仅定义投影信息。调用Reproject()方法实际重新投影数据集和层。

IFeatureLayer: 这是不应该实例化的，因为它本身不能执行必要的功能。相反，大多数指定的功能必须在更具体的类中实现。这也是为什么这个类没有直接构造函数的原因。您可以使用静态“Fromfile”或“FromFeulurelayer”从一个文件创建FeatureLayer。

属性：

* + - 1. new IFeatureSet DataSet获取或设置基础FeatureSet
      2. bool EditMode获取或设置布尔值。如果为true，则忽略功能索引，并且假定Features完全加载到RAM中，并且停止高亮显示功能。如果为false，则使用索引代替，而不假定功能加载到RAM中。
      3. FastDrawnState[] DrawnStates
      4. bool DrawnStatesNeeded获取或设置布尔标志，该标志控制是否需要绘制状态。如果没有选中的要素，并且只有一个类别，并且该类别上没有筛选器表达式，则这应该是假的。
      5. ILabelLayer LabelLayer获取或设置标签layer。
      6. ISelection Selection获取一个选择类，该类允许用户为每个方法循环所有未选择的特性，即使这些特性属于许多类别。
      7. IFeatureSymbolizer SelectionSymbolizer获取或设置要与选定功能一起使用的共享特征。
      8. bool ShowLabels获取或设置是否应绘制标签。
      9. IFeatureSymbolizer Symbolizer获取或设置点、线和多边形特征之间的共享符号特征和接口。
      10. IFeatureScheme Symbology获取当前的feature scheme，但是要更改它，应该调用applyplan，以便更新feature类别

方法：

IFeatureScheme：