

MapX 4.0

Справочник

Содержание

Введение	i	Метод CoordSys.Clone	28
Объект AffineTransform	1	Свойство CoordSys.Datum	28
AffineTransform. Свойства A, B, C, D, E, F	1	Свойства CoordSys.FalseEasting, FalseNorthing	28
Метод AffineTransform.Set	2	Свойства CoordSys.OriginLatitude, OriginLongitude	29
Свойства AffineTransform.Units	3	Метод CoordSys.PickCoordSys	29
Объект и коллекция Annotation	3	Свойство CoordSys.Range	30
Свойство Annotation.Graphic	6	Свойство CoordSys.ScaleFactor	30
Свойство Annotation.Type	6	Метод CoordSys.Set	30
Метод Annotations.ActiveAnnotation	7	Свойства CoordSys.StandardParallelOne, StandardParallelTwo	32
Метод Annotations.AddSymbol	7	Свойство CoordSys.Type	32
Метод Annotations.AddText	8	Свойство CoordSys.Units	32
Свойство Annotations.Count	9	Объект и коллекция DataSet	32
Свойство Annotations.Editable	9	Метод DataSet.AddField	36
Свойство Annotations.Item	9	Свойство Datasets.BuildSourceRows	37
Метод Annotations.Remove	10	Свойство DataSet.Fields	38
Метод Annotations.RemoveAll	10	Свойство DataSet.GeoField	38
Объект BindLayer	11	Свойство DataSet.Layer	38
Свойство BindLayer.CoordSys	13	Свойство DataSet.Name	40
Свойство BindLayer.Filespec	14	Метод DataSet.Refresh	40
Свойство BindLayer.KeyLength	14	Метод DataSet.RowValues	41
Свойство BindLayer.LayerName	15	Свойство DataSet.RowCount	41
Свойство BindLayer.LayerType	15	Свойство DataSet.ReadOnly	41
Свойство BindLayer.RefColumn1	16	Метод DataSets.Restore	42
Свойство BindLayer.RefColumn2	16	Свойство DataSet.SecondaryGeoField	43
Свойство BindLayer.ReferenceLayer	17	Свойство DataSet.SourceRows	43
Свойство BindLayer.ReferenceLayerField20		Свойство DataSet.Themes	43
Объект BitmapSymbol и коллекция BitmapSymbols	20	Свойство DataSet.Type	43
Свойство BitmapSymbol.Name	24	Свойство DataSet.Value	44
Свойство BitmapSymbols.Count	24	Метод DataSets.Add	48
Свойство BitmapSymbols.Item	25	Свойство DataSets.Count	58
Метод BitmapSymbols.Refresh	25	Свойство DataSets.Item	59
Метод BitmapSymbols.Unload	26	Метод DataSets.Remove	60
Объект CoordSys	26	Метод DataSets.RemoveAll	61
Свойство CoordSys.AffineTransform	27	Объект Datum	63
Свойство CoordSys.Azimuth	28	Свойство Datum.Eccentricity	63
Свойство CoordSys.Bounds	28	Свойство Datum.Ellipsoid	64
		Свойство Datum.Flattening	64

Свойство Datum.PrimeMeridian	64	Метод FeatureFactory.BufferFeatures	103
Свойства Datum.RotateX, RotateY, RotateZ	65	Метод FeatureFactory.CombineFeatures	105
Свойство Datum.ScaleAdjust	65	Метод FeatureFactory.CreateArc	106
Свойства Datum.SemiMajorAxis, SemiMinorAxis	65	Метод FeatureFactory.CreateCircularRegion	109
Метод Datum.Set	65	Метод FeatureFactory.CreateEllipticalRegion	110
Метод Datum.SetFromList	66	Метод FeatureFactory.CreateLine	113
Свойства Datum.ShiftX, ShiftY, ShiftZ	67	Метод FeatureFactory.CreateRegion	113
Объект и коллекция Feature	67	Метод FeatureFactory.CreateSymbol	114
Свойство Feature.Area	70	Метод FeatureFactory.CreateText	114
Метод Feature.Attach	71	Метод FeatureFactory.EraseFeature	115
Метод Feature.Clone	73	Метод FeatureFactory.IntersectFeatures	118
Свойство Feature.Bounds	74	Метод FeatureFactory.IntersectionPoints	119
Свойство Feature.Caption	75	Метод FeatureFactory.IntersectionTest	120
Свойства Feature.CenterX, CenterY	77	Объект и коллекция Field	121
Свойство Feature.FeatureID	77	Свойство Field.AggregationFunction	122
Свойство Feature.FeatureKey	78	Свойство Field.Name	122
Свойство Feature.KeyValue	78	Свойство Field.Type	122
Свойство Feature.LabelPoint	82	Метод Fields.Add	122
Свойство Feature.Layer	82	Свойство Fields.Count	125
Свойство Feature.Length	82	Свойство Fields.Item	125
Свойство Feature.Name	85	Метод Fields.Remove	126
Свойство Feature.Nodes	85	Метод Fields.RemoveAll	128
Метод Feature.Offset	87	Объект Find	130
Свойство Feature.Parts	88	Свойство Find.Abbreviations	136
Свойство Feature.Perimeter	88	Свойство Find.CloseMatchMax	136
Свойство Feature.Point	88	Свойство Find.ClosestAddr	136
Метод Feature.Smooth	89	Свойство Find.FindDataset	136
Свойство Feature.Style	89	Свойство Find.FindField	137
Свойство Feature.Type	89	Метод Find.SearchEx	137
Метод Feature.Update	89	Свойство Find.OtherBoundary	140
Метод Features.Add	94	Свойство Find.RefineDataset	140
Метод Features.AddByID	95	Свойство Find.RefineField	141
Свойство Features.Bounds	96	Свойство Find.RefineLayer	141
Метод Features.Clone	97	Метод Find.Search	141
Метод Features.Common	97	Объект FindFeature	142
Свойство Features.Count	99	Свойство FindFeature.FindRC	142
Свойство Features.Item	99		
Метод Features.Remove	99		
Метод Features.RemoveByID	100		
Метод Features.Replace	101		
Объект FeatureFactory	101		

Объект FindMatch и коллекция FindMatches

144

Свойство FindMatch.Name	144
Свойство FindMatch.FeatureID	145
Свойство FindMatch.Score	145
Свойство FindMatches.Count	145
Свойство FindMatches.Item	145

Объект FindResult

146

Свойство FindResult.AddressOutOfRange	146
Свойство FindResult.ExactMatch	146
Свойство FindResult.FindRC	147
Свойство FindResult.IntersectionNotFound	147
Свойство FindResult.MatchedFeature	147
Свойство FindResult.Matches	147
Свойство FindResult.MultipleMatches	147
Свойство FindResult.RefineRegion	148
Свойство FindResult.Substitute	148

Объект Geoset и коллекция Geosets

148

Свойство Geoset.Centroid	149
Свойство Geoset.PathName	149
Свойство Geoset.UserName	149
Свойство Geosets.Count	149
Свойство Geosets.Item	150

Объект Graphic

150

Свойство Graphic.Caption	150
Свойство Graphic.Position	151
Свойство Graphic.Style	151
Свойство Graphic.X	151
Свойство Graphic.Y	152

Объект IndividualValueCategory, коллекция IndividualValueCategories

152

Свойство IndividualValueCategories.AllOthersCategory	154
Свойство IndividualValueCategories.Count	155
Свойство IndividualValueCategories.Item	155

Свойство IndividualValueCategory.NumItems	156
---	-----

Свойство IndividualValueCategory.Style	156
--	-----

Свойство IndividualValueCategory.Value	156
--	-----

Объект LabelProperties

157

Свойство LabelProperties.DataField	158
Свойство LabelProperties.DataSet	161
Свойство LabelProperties.Duplicate	161
Свойство LabelProperties.LabelMax	161
Свойство LabelProperties.LabelZoom	162
Свойство LabelProperties.LabelZoomMax	162

Свойство LabelProperties.LabelZoomMin	162
---------------------------------------	-----

Свойство LabelProperties.LineType	163
Свойство LabelProperties.Offset	163
Свойство LabelProperties.Overlap	163
Свойство LabelProperties.Parallel	164
Свойство LabelProperties.PartialSegments	164

Свойство LabelProperties.Position	164
-----------------------------------	-----

Свойство LabelProperties.Style	164
--------------------------------	-----

Свойство LabelProperties.Visible	165
----------------------------------	-----

Объект Layer и коллекция Layers

165

Метод Layer.AddFeature	168
Метод Layer.AllFeatures	172
Метод Layer.BeginAccess	173

Метод Layer.ClearCustomLabels	174
-------------------------------	-----

Свойство Layer.Editable	174
-------------------------	-----

Метод Layer.EndAccess	174
-----------------------	-----

Метод Layer.Search	175
--------------------	-----

Свойство Layer.AutoLabel	176
--------------------------	-----

Свойство Layer.Bounds	176
-----------------------	-----

Свойство Layer.CoordSys	177
-------------------------	-----

Свойство Layer.DataSets	178
-------------------------	-----

Метод Layer.DeleteFeature	178
---------------------------	-----

Свойство Layer.DrawLabelsAfter	179
--------------------------------	-----

Метод Layer.DrilldownAddFeatures	179
----------------------------------	-----

Метод Layer.DrilldownRemoveFeatures	180
-------------------------------------	-----

Метод Layer.DrilldownReset	181
----------------------------	-----

Метод Layer.FeatureIDFromFeatureName	181
--------------------------------------	-----

Метод Layer.FeatureKeyFromFeatureName	182	Метод Layers.Move	214
Свойство Layer.FileSpec	182	Метод Layers.Remove	215
Свойство Layer.Find	182	Метод Layers.RemoveAll	216
Метод Layer.GetDrilldownFeaturesById	183	Объект LayerInfo	217
Метод Layer.GetFeatureById	183	Свойство LayerInfo.Type	219
Метод Layer.GetFeatureByKey	183	Метод LayerInfo.AddParameter	220
Метод Layer.Invalidate	184	Объект Legend	220
Свойство Layer.KeyField	185	Свойство Legend.BodyTextStyle	221
Метод Layer.LabelAtPoint	185	Свойство Legend.Compact	222
Свойство Layer.LabelProperties	187	Свойство Legend.CompactTitle	222
Свойство Layer.Name	187	Свойство Legend.CompactTitleStyle	222
Метод Layer.NoFeatures	187	Свойство Legend.CurrencyFormat	223
Свойство Layer.OverrideStyle	188	Метод Legend.ExportLegend	223
Свойство Layer.PredominantFeatureType	188	Свойство Legend.Height	224
Метод Layer.Refresh	188	Свойство Legend.Left	224
Метод Layer.SearchAtPoint	189	Метод Legend.LegendDlg	224
Метод Layer.SearchWithinDistance	190	Свойство Legend.LegendTexts	226
Метод Layer.SearchWithinFeature	195	Свойство Legend.PaperHeight	226
Метод Layer.SearchWithinRectangle	196	Свойство Legend.PaperWidth	226
Свойство Layer.Selectable	197	Метод Legend.PrintLegend	227
Свойство Layer.Selection	197	Свойство Legend.ShowCount	229
Свойство Layer.ShowNodes	197	Свойство Legend.ShowEmptyRanges	230
Свойство Layer.ShowCentroids	197	Свойство Legend.SubTitle	231
Свойство Layer.ShowLineDirection	198	Свойство Legend.SubTitleStyle	231
Свойство Layer.Style	198	Свойство Legend.Title	231
Свойство Layer.Type	198	Свойство Legend.TitleStyle	231
Метод Layer.UpdateFeature	201	Свойство Legend.Top	232
Свойство Layer.Visible	202	Свойство Legend.Visible	232
Свойство Layer.ZoomLayer	202	Свойство Legend.Width	232
Свойство Layer.ZoomMax	202	Объект и коллекция LegendText	234
Свойство Layer.ZoomMin	203	Свойство LegendTexts.AllOthersText	234
Метод Layers.Add	203	Свойство LegendText.Text	235
Метод Layers.AddGeosetLayers	204	Свойство LegendText.Visible	237
Метод Layers.AddServerLayer	205	Свойство LegendTexts.AutoGenerate	239
Метод Layers.AddUserDrawLayer	207	Свойство LegendTexts.Count	239
Свойство Layers.AnimationLayer	207	Свойство LegendTexts.Item	239
Свойство Layers.Bounds	208	Объект Map	240
Метод Layers.ClearSelection	209	Метод Map.AboutBox	242
Свойство Layers.Count	209	Свойство Map.Annotations	242
Метод Layers.CreateLayer	210	Свойство Map.AreaUnit	243
Свойство Layers.InsertionLayer	212	Свойство Map.AutoRedraw	243
Свойство Layers.Item	213		
Метод Layers.LayersDlg	213		

Свойство Map.BackColor	245	Свойство Map.PaperUnit	285
Свойство Map.Bounds	245	Свойство Map.PreferCompactLegends	285
Свойство Map.CenterX	246		
Свойство Map.CenterY	247	Метод Map.PrintMap	287
Метод Map.ClipLine	247	Метод Map.PropertyPage	289
Метод Map.CliplineV	249	Свойство Map.RedrawInterval	289
Метод Map.ConvertCoord	250	Метод Map.Refresh	290
Метод Map.ConvertCoordV	252	Свойство Map.Rotation	291
Метод Map.CreateCustomTool	255	Метод Map.SaveMapAsGeoset	291
Свойство Map.CurrentTool	259	Свойство Map.SearchPath	292
Свойство Map.DataSet	260	Свойство Map.SelectionStyle	293
Свойство Map.DataSetGeoField	261	Метод Map.SetSize	294
Свойство Map.DataSets	261	Свойство Map.Title	295
Свойство Map.DataSetTheme	262	Свойство Map.TitleText	295
Свойство Map.DefaultConversionResolution	262	Свойство Map.Version	295
		Свойство Map.WaitCursorEnabled	296
Свойство Map.DefaultStyle	263	Свойство Map.Zoom	296
Свойство Map.DisplayCoordSys	265	Метод Map.ZoomTo	296
Метод Map.Distance	265		
Метод Map.ExportMap	267	Объект MultivarCategory и коллекция	
Свойство Map.ExportSelection	269	MultivarCategories	298
Свойство Map.FeatureFactory	270	Свойство MultivarCategories.Count	301
Свойство Map.GeoDictionary	270	Свойство MultivarCategories.Item	301
Свойство Map.GeoSet	271	Свойство MultivarCategory.Style	301
Свойство Map.GeoSets	273		
Свойство Map.GeoSetWidth	273	Объект NotesQueryInfo	301
Свойство Map.hWnd	274	Свойство NotesQueryInfo.BeginDate	302
Свойство Map.InfotipSupport	274	Свойство NotesQueryInfo.Database	302
Свойство Map.InfotipPopupDelay	275	Свойство NotesQueryInfo.DefaultNumericValue	303
Метод Map.IsPointVisible	275		
Свойство Map.Layers	276	Свойство NotesQueryInfo.DefaultStringValue	303
Свойство Map.MapUnit	276		
Свойство Map.MapPaperHeight	278	Свойство NotesQueryInfo.EndDate	303
Свойство Map.MapPaperWidth	278	Свойство NotesQueryInfo.FullTextSearch	303
Свойство Map.MapScreenHeight	279		
Свойство Map.MapScreenWidth	280	Свойство NotesQueryInfo.MaxNumDocs	304
Свойство Map.MatchNumericFields	280		
Свойство Map.MatchThreshold	281	Свойство NotesQueryInfo.Query	304
Свойство Map.MaxSearchTime	281	Свойство NotesQueryInfo.Server	304
Свойство Map.Mouselcon	281		
Свойство Map.MousePointer	282	Объект NotesViewInfo	305
Свойство Map.MouseWheelSupport	283	Свойство NotesViewInfo.Database	305
Свойство Map.NumericCoordSys	283	NotesViewInfo.Server	305
Метод Map.Pan	284	NotesViewInfo.View	306
Свойство Map.PanAnimationLayer	284		

Объект ODBCQueryInfo	306	Свойство Rectangle.YMax	340
Свойство ODBCQueryInfo.ConnectionString	310	Свойство Rectangle.YMin	340
Свойство ODBCQueryInfo.DataSource	314	Объект ResolveObject и коллекция ResolveObjects	340
Свойство ODBCQueryInfo.SqlQuery	314	Свойство ResolveObject.TableName	341
Коллекция Parts	314	Свойство ResolveObject.SourceMatch	341
Метод Parts.Add	317	Свойство ResolveObject.TableMatch	341
Свойство Parts.Count	317	Свойство ResolveObjects.Count	341
Свойство Parts.Item	318	Свойство ResolveObjects.Item	342
Метод Parts.Remove	318	Объект RowValue и коллекция RowValues	342
Метод Parts.RemoveAll	318	Свойство RowValue.ReadOnly	344
Объект и коллекция Point	321	Свойство RowValue.Dataset	344
Метод Point.Offset	323	Свойство RowValue.Field	344
Метод Point.Set	324	Свойство RowValue.Value	344
Свойство Point.X	324	Свойство RowValues.County	344
Свойство Point.Y	324	Свойство RowValues.ReadOnly	345
Метод Points.Add	325	Свойство RowValues.Item	345
Метод Points.AddXY	328	Метод RowValues.Remove	345
Свойство Points.Count	329	Метод RowValues.Add	346
Свойство Points.Item	329	Метод RowValues.RemoveAll	346
Метод Points.Remove	329	Метод RowValues.Clone	347
Метод Points.RemoveAll	329	Коллекция Selection	347
Объект RangeCategory, Коллекция RangeCategories	332	Метод Selection.Add	349
Свойство RangeCategories.AllOthersCategory	334	Метод Selection.ClearSelection	351
Свойство RangeCategories.Count	335	Метод Selection.Clone	351
Свойство RangeCategories.Item property	335	Метод Selection.Common	352
Свойство RangeCategory.Max	336	Метод Selection.Remove	352
Свойство RangeCategory.Min	336	Метод Selection.Replace	353
Свойство RangeCategory.NumItems	337	Метод Selection.SelectAll	353
Свойство RangeCategory.Style	337	Метод Selection.SelectById	354
Объект Rectangle	337	Метод Selection.SelectByPoint	354
Свойство Rectangle.Height	338	Метод Selection.SelectByRadius	357
Метод Rectangle.Offset	338	Метод Selection.SelectByRectangle	357
Метод Rectangle.Set	339	Метод Selection.SelectByRegion	359
Свойство Rectangle.Width	339	Объект и коллекция SourceRow	361
Свойство Rectangle.XMax	339	Свойство SourceRow.Row	361
Свойство Rectangle.XMin	340	Свойство SourceRows.Count	362
		Свойство SourceRows.Item	362
		Объект Style	365

Метод Style.Clone	367	Свойство Style.SymbolFontShadow	397
Метод Style.DrawLineSample	369	Свойство Style.MinVectorSymbolCharacter	398
Метод Style.DrawRegionSample	371	Свойство Style.MaxVectorSymbolCharacter	398
Метод Style.DrawSymbolSample	373	Свойство Style.SymbolVectorColor	398
Метод Style.DrawTextSample	376	Свойство Style.SymbolVectorSize	398
Метод Style.ExportLineSample	378	Свойство Style.TextFont	399
Метод Style.ExportRegionSample	379	Свойство Style.TextFontAllCaps	399
Метод Style.ExportSymbolSample	380	Свойство Style.TextFontBackColor	399
Метод Style.ExportTextSample	382	Свойство Style.TextFontColor	399
Свойство Style.LineColor	383	Свойство Style.TextFontDbfSpace	400
Свойство Style.LineInterleaved	383	Свойство Style.TextFontHalo	400
Свойство Style.LineStyle	386	Свойство Style.TextFontOpaque	400
Свойство Style.LineStyleCount	387	Свойство Style.TextFontShadow	400
Свойство Style.LineSupportsInterleave	387	Свойство Style.TextFontRotation	401
Свойство Style.LineWidth	387	Свойство Style.SymbolType	401
Свойство Style.LineWidthUnit	388	Объект и коллекция Theme	401
Метод Style.PickLine	388	Свойство Theme.AutoReCompute	402
Метод Style.PickRegion	388	Свойство Theme.ComputeTheme	403
Метод Style.PickSymbol	389	Свойство Theme.DataMax	403
Метод Style.PickText	389	Свойство Theme.DataMin	404
Свойство Style.RegionBackColor	389	Свойство Theme.Fields	404
Свойство Style.RegionBorderColor	390	Свойство Theme.Layer	404
Свойство Style.RegionBorderStyle	390	Свойство Theme.Legend	404
Свойство Style.RegionBorderWidth	390	Свойство Theme.Name	405
Свойство Style.RegionBorderWidthUnit	390	Метод Theme.ThemeDlg	405
Свойство Style.RegionColor	391	Свойство Theme.ThemeProperties	407
Свойство Style.RegionPattern	391	Свойство Theme.Type	407
Свойство Style.RegionTransparent	391	Свойство Theme.Visible	408
Свойство Style.SupportsBitmapSymbols	392	Метод Themes.Add	408
Свойство Style.SymbolBitmapColor	392	Свойство Themes.Count	413
Свойство Style.SymbolBitmapName	393	Свойство Themes.Item	414
Свойство Style.SymbolBitmapOverrideColor	393	Метод Themes.Remove	414
Свойство Style.SymbolBitmapSize	394	Метод Themes.RemoveAll	415
Свойство Style.SymbolBitmapTransparent	394	Объект ThemeProperties	416
Свойство Style.SymbolCharacter	395	Свойство ThemeProperties.AllowEmptyRanges	418
Свойство Style.SymbolFont	395	Свойство ThemeProperties.DataValue	418
Свойство Style.SymbolFontBackColor	396	Свойство ThemeProperties.DistanceMethod	419
Свойство Style.SymbolFontColor	396	Свойство ThemeProperties.DotSize	419
Свойство Style.SymbolFontHalo	396	Свойство ThemeProperties.Graduated	420
Свойство Style.SymbolFontOpaque	397		
Свойство Style.SymbolFontRotation	397		

Свойство ThemeProperties.Independent	420	Свойство ThemeProperties.InflectRanges	430
Свойство ThemeProperties.IndividualValueCategories	421	Свойство ThemeProperties.InflectionRange	431
Свойство ThemeProperties.MultivarCategories	421	Свойство ThemeProperties.InflectionColor	431
Свойство ThemeProperties.NumRanges	421	Свойство ThemeProperties.ColorMethod	432
Свойство ThemeProperties.RangeCategories	421	Свойство ThemeProperties.ApplyAttribute	432
Свойство ThemeProperties.Size	421	Объект Title	433
Свойство ThemeProperties.SpreadBy	422	Свойство Title.Border	434
Свойство ThemeProperties.SymbolStyle	422	Свойство Title.Caption	435
Свойство ThemeProperties.ValuePerDot	423	Свойство Title.Editable	435
Свойство ThemeProperties.Width	423	Свойство Title.Position	435
Свойство ThemeProperties.PieClockwise	423	Свойство Title.TextStyle	435
Свойство ThemeProperties.PieStartAngle	424	Свойство Title.Visible	435
Свойство ThemeProperties.PieHalfPies	424	Свойство Title.X	436
Свойство ThemeProperties.PieGraduated	424	Свойство Title.Y	436
Свойство ThemeProperties.BarStacked	424	События MapX	436
Свойство ThemeProperties.BarGraduatedStack	425	AddFeatureToolUsed	438
Свойство ThemeProperties.BarIndependentScale	425	AnnotationAdded	439
Свойство ThemeProperties.BarWidth	426	AnnotationChanged	441
Свойство ThemeProperties.BarFramed	426	DataMismatch	444
Свойство ThemeProperties.BarFrameStyle	426	DrawUserLayer	449
Свойство ThemeProperties.BorderStyle	427	MapDraw	453
Свойство ThemeProperties.ShowNegativeValues	427	MapInitialized	454
Свойство ThemeProperties.PositiveSymbolStyle	427	MapViewChanged	454
Свойство ThemeProperties.NegativeSymbolStyle	428	MouseWheel	454
Свойство ThemeProperties.GraduateSizeBy	428	PolyToolUsed	457
Свойство ThemeProperties.DotColor	429	RequestData	463
Свойство ThemeProperties.RoundRanges	429	ResolveDataBind	467
Свойство ThemeProperties.RoundBy	429	ResolveDataBindEx	468
		SelectionChanged	469
		ThemeModifyRequested	470
		ToolUsed	471
		Приложение A: Коды ошибок MapX	477
		Приложение B: Константы	489
		Приложение C: Создание выражений	504
		Операторы	504

Функции	509
Приложение D: Ключи Geoset	525
Приложение E: Значения OLE_COLOR	
	531

Объект AffineTransform

Объект AffineTransform позволяет вращать и искажать системы координат.

Свойства объекта

- Свойство AffineTransform.A, B, C, D, E, F (Объект AffineTransform)
- Свойство AffineTransform.Units (Объект Affine Transform)

Методы объекта

- Метод AffineTransform.Set

Примечание

Все свойства открыты только для чтения. Для изменения объекта AffineTransform, используйте метод Set.

Для получения объекта AffineTransform, сошлитесь на свойство CoordSys.AffineTransform или объявите новый, независимый объект Affine Transform.

Смотрите также

Для получения общих представлений об аффинных преобразованиях, смотрите раздел Using Coordinate Systems в электронной справочной системе.

AffineTransform. Свойства A, B, C, D, E, F (Объект AffineTransform)

Назначение

Свойство, открытое только для чтения, представляет параметры аффинного преобразования; возвращает значения двойной точности (типа Double).

Примечание

Аффинные преобразования вычисляются по следующим формулам:

$$x' = Ax + By + C$$

$$y' = Dx + Ey + F$$

В этих уравнениях, начальные координаты (x,y) преобразуются, в результате трансформации возникают новые значения координат (x', y'). Свойства от A до F определяют единственным образом параметры трансформации, а именно:

A Осуществляет трансформацию вдоль оси X.

- В Осуществляет вращение оси X.
- С Осуществляет сдвиг вдоль оси X.
- D Осуществляет трансформацию вдоль оси Y.
- E Осуществляет вращение оси Y.
- F Осуществляет сдвиг вдоль оси Y.

Эти настройки открыты только для чтения; чтобы их изменить, используйте метод Set.

Метод AffineTransform.Set (Объект AffineTransform)

Назначение

Устанавливает настройки аффинного преобразования.

Синтаксис

ОБЪЕКТ.Set UNITS [A, B, C, D, E, F]

Раздел	Описание
ОБЪЕКТ	Объект, подвергающийся аффинному преобразованию.
UNITS	Константа единиц измерения карты (MapUnitConstants), такая как miUnitMeter (7).
A	Величина двойной точности. Осуществляет трансформацию вдоль оси X.
B	Величина двойной точности. Осуществляет вращение или искажение вдоль оси X.
C	Величина двойной точности. Осуществляет сдвиг вдоль оси X.
D	Величина двойной точности. Осуществляет трансформацию вдоль оси Y.
E	Величина двойной точности. Осуществляет вращение или искажение вдоль оси Y.
F	Величина двойной точности. Осуществляет сдвиг вдоль оси Y.

Примечание

Требуется задавать все аргументы.

Аффинные преобразования вычисляются по следующим формулам:

$$x' = Ax + By + C$$

$$y' = Dx + Ey + F$$

В этих уравнениях, начальные координаты (x,y) преобразуются, в результате чего рассчитываются новые значения координат (x', y')..

Свойства `AffineTransform.Units` (Объект `AffineTransform`)

Назначение

Свойство, открытое только для чтения (Read-only) определяющее, какие единицы измерения карты используются в аффинных преобразованиях; возвращает константу типа короткое целое (short) соответствующую объявленной в выражении `MapUnitConstants`.

Синтаксис

`object.Units`

`object` - это тот объект, к которому применяется аффинное преобразование, в результате чего возникает новый объект.

Объект и коллекция `Annotation`

Назначение

Каждая Карта имеет коллекцию аннотаций (подписей) `Annotations` (Свойство `Map.Annotations`). `Annotations` - это или символьный или текстовый объект, отображающийся сверху карты.

`Annotations` обычно используются для добавления сообщения (текст) к карте или для размещения символа на карте. Такие аннотации будут изменяться при изменении масштаба карты. Нет необходимости связывать их с каким-либо слоем карты, так как они всегда сверху карты.

Обратите внимание, что объект `Annotation` не имеет таких настроек как позиция, стиль символа или текст. Чтобы управлять такими настройками аннотации, Вам надо использовать свойство `Annotation.Graphic`, что бы получить объект `Graphic`, и затем изменить этот объект.

Свойства объекта

- Свойство `Annotation.Graphic` (Объект `Annotation`)
- Свойство `Annotation.Type` (Объект `Annotation`)

Свойства коллекции

- Свойство `Annotations.Count` (Коллекция `Annotations`)
- Свойство `Annotations.Editable` (Коллекция `Annotations`)
- Свойство `Annotations.Item` (Коллекция `Annotations`)

Методы коллекции

- Метод `Annotations.ActiveAnnotation` (Коллекция `Annotations`)
- Метод `Annotations.AddSymbol` (Коллекция `Annotations`)
- Метод `Annotations.AddText` (Коллекция `Annotations`)
- Метод `Annotations.Remove` (Коллекция `Annotations`)
- Метод `Annotations.RemoveAll` (Коллекция `Annotations`)

Пример на C++

```
// Annotations object
// Annotation.Type method
// Annotation.Graphic method
void CSampleProjectView::ChangeAllAnnotations() {
    // Change the color of all of the existing annotations
    // (both symbol and text) to Maroon
    try {
        CMapXAnnotations Annotations = Map.GetAnnotations();
        CMapXAnnotation CurrentAnnotation;
        long i, AnnotCount = Annotations.GetCount();
        for(i=1; i<=AnnotCount; i++) { // Loop through all of the annotations
            CurrentAnnotation = Annotations.Item(i);
            if(CurrentAnnotation.GetType() == miSymbolAnnotation) {
                // The annotation is a symbol
```

```

        CurrentAnnotation.GetGraphic().GetStyle().
        SetSymbolFontColor(miColorMaroon);
    } else {
        // The annotation is a text annotation
        CurrentAnnotation.GetGraphic().GetStyle().
        SetTextFontColor(miColorMaroon);
    }
}
} catch (ColeDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (ColeException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Пример на Visual Basic

```

Private Sub ChangeAnnotations_Click()
    ' Change the color of all of the existing annotations
    '(both symbol and text) to Maroon
    Dim CurrentAnnotation As Annotation
    For Each CurrentAnnotation In Map1.Annotations
        ' Loop through all of the annotations
        If CurrentAnnotation.Type = miSymbolAnnotation Then
            CurrentAnnotation.Graphic.Style.SymbolFontColor _
            = miColorMaroon
        Else

```



```

        CurrentAnnotation.Graphic.Style.TextFontColor _
        = miColorMaroon
    End If
Next
End Sub

```

Смотрите также

Объект Map

Свойство Annotation.Graphic (Объект Annotation)

Назначение

Это свойство содержит объект Graphic, который содержит в свою очередь свойства для аннотации (Annotation).

Смотрите также

Объект Graphic

Свойство Annotation.Type (Объект Annotation)

Назначение

Определяет тип аннотации. Это величина типа AnnotationTypeConstants, и она может быть или miSymbolAnnotation или miTextAnnotation. Это свойство только для чтения.

Смотрите также

AnnotationTypeConstants

Метод **Annotations.ActiveAnnotation** (Коллекция **Annotations**)

Назначение

Этот метод возвращает активную (выбранную в данный момент) аннотацию.

Синтаксис

ОБЪЕКТ.ActiveAnnotation

Метод **Annotations.AddSymbol** (Коллекция **Annotations**)

Назначение

Добавляет символ к коллекции аннотации. По умолчанию используется стиль, определенный в Map.DefaultStyle.

Синтаксис

Annotations.AddSymbol [X, Y]

Часть Описание

X	X координата добавляемого символа. Она имеет двойную точность (Double), и определяет долготу в координатах карты
Y	Y координата добавляемого символа. Она имеет двойную точность (Double), и определяет широту в координатах карты.

Смотрите также

Свойство Map.DefaultStyle

Свойство Graphic.X

Свойство Graphic.Y

Пример на **Visual Basic**

```
Map1.Annotations.AddSymbol -75.14, 42.9
Map1.Annotations.AddSymbol -98, 31.56
Map1.Annotations.AddSymbol -118.92, 36.75
Map1.Refresh
```

Метод **Annotations.AddText** (Коллекция аннотаций)

Назначение

Добавляет текст к коллекции аннотаций. По умолчанию используется стиль, определенный в Map.DefaultStyle.

Синтаксис

ОБЪЕКТ.AddText (Text, X, Y, [Position])

Часть Описание

ОБЪЕКТ	Объект для которого создается аннотация.
Text	Строковая величина. Она становится заголовком новой аннотации..
X	X координата добавляемого текста. Она имеет двойную точность (Double), и определяет долготу в координатах карты
Y	Y координата добавляемого текста. Она имеет двойную точность (Double), и определяет широту в координатах карты.
Position	Привязывает текст к точке с координатами x,y. Для этого берется значение PositionConstants.

Смотрите также

Свойство Map.DefaultStyle

Свойство Graphic.Caption

Свойство Graphic.Position

Свойство Graphic.X

Свойство Graphic.Y

PositionConstants

Примеры на Visual Basic

```
Private Sub Command1_Click()
    Map1.Annotations.AddText "An Annotation", -75.14, 42.9, _
    miPositionTL
    Map1.Annotations.AddText "Another One", -98, 31.56, _
    miPositionCC
```

```

Map1.Annotations.AddText "A Third", -118.92, 36.75, _
miPosition BR
End Sub

```

Свойство Annotations.Count (Коллекция аннотаций)

Назначение

Определяет количество объектов Annotation в коллекции. Это целая величина, и открыта только для чтения.

Свойство Annotations.Editable (Коллекция аннотаций)

Назначение

Определяет, являются ли аннотации редактируемыми для конечного пользователя. Редактируемая аннотация может быть перемещена, может изменяться в размерах и редактироваться. Эта величина логическая. Принимает значения True или False, по умолчанию True.

Свойство Annotations.Item (Коллекция аннотаций)

Назначение

Для получения объекта Annotation из коллекции. Индекс используется для того, чтобы указать, какой объект Annotation надо получить. Индекс это целая величина от 1 до Annotations.Count. Это свойство используется по умолчанию для коллекции Annotations.

Смотрите также

Свойство Annotations.Count

Метод **Annotations.Remove** (Коллекция аннотаций)

Назначение

Удаляет объект аннотации из коллекции.

Внимание: Если Вы осуществите удаление, индексы коллекции изменятся так, чтобы заполнить образовавшийся пробел.

Синтаксис

ОБЪЕКТ.Remove (Index)

Часть Описание

ОБЪЕКТ Это объект аннотации (Annotations object).

Index Индекс используемой аннотации. Это целая величина от 1 до Annotations.Count.

Пример на Visual Basic

```
`Remove the second annotation
Map1.Annotations.Remove 2
```

Смотрите также

Свойство Annotations.Count

Метод **Annotations.RemoveAll** (Коллекция аннотаций)

Назначение

Удаляет все объекты аннотаций из коллекции.

Синтаксис

ОБЪЕКТ.RemoveAll

Часть Описание

ОБЪЕКТ Это объект аннотации (Annotations object).

Объект BindLayer

Объект Bindlayer используется как параметр в методе Datasets.Add. Используется для указания того, как выполнять связь данных (data binding). Необходим при создании точечных объектов в новом слое по известным координатам X и Y или по почтовым кодам (zip-кодам).

Свойства объекта

- Свойство BindLayer.CoordSys (Объект BindLayer)
- Свойство BindLayer.Filespec (Объект BindLayer)
- Свойство BindLayer.KeyLength (Объект BindLayer)
- Свойство BindLayer.LayerName (Объект BindLayer)
- Свойство BindLayer.LayerType (Объект BindLayer)
- Свойство BindLayer.RefColumn1 (Объект BindLayer)
- Свойство BindLayer.RefColumn2 (Объект BindLayer)
- Свойство BindLayer.ReferenceLayer (Объект BindLayer)
- Свойство BindLayer.ReferenceLayerField (Объект BindLayer)

Пример на Visual Basic

```
` Set up the bind Object Layer
Dim BindLayerObject As New MapXLib.BindLayer
BindLayerObject.LayerName = "Customer's By Lat/Long coords"
BindLayerObject.RefColumn1 = 3           ` The third _
column in my source data contains x coords
BindLayerObject.RefColumn2 = 4           ` The fourth _
column in my source data contains y coords
BindLayerObject.LayerType = miBindLayerTypeXY
` The values in the source data column named "GEOABBR" will _
be used as the geofield for the new point table
Set MyDataset = Map.Datasets.Add(1, Data2.Recordset, _
"Lat/Long Dataset", "GEOABBR", , BindLayerObject)
```

MyDataset.Themes.Add

Пример на C++

```
void CSampleProjectView::BindLayerUse(CDaoRecordset& SourceDAORecordset)
{
    CMapXBindLayer BindLayerObject;
    CMapXDataset MyDataset;
    COleVariant SourceDataVt, BindLayerVt;
    // Create a new bindОбъект Layer
    if(!BindLayerObject.CreateDispatch(BindLayerObject.GetClsid())) {
        TRACE0("Failed to Create a BindОбъект Layer");
        return;
    }

    try {
        // Set up the bind Объект Layer
        BindLayerObject.SetLayerName("Customers By Lat/Long Coords");

        // The third column in my source data contains the
        // X coordinates
        BindLayerObject.SetRefColumn1(3);

        // The fourth column in my source data contains the
        // Y coordinates
        BindLayerObject.SetRefColumn2(4);

        BindLayerObject.SetLayerType(miBindLayerTypeXY);

        BindLayerVt.vt = VT_DISPATCH;
        BindLayerVt.pdispVal = BindLayerObject.m_lpDispatch;
```

```

SourceDataVt.vt= VT_DISPATCH;

SourceDataVt.pdispVal = SourceDAORecordset.m_pDAORecordset;

// The values in the source data column named "GEOABBR" _
will be used as the geofield for the new point table

MyDataset = m_Map.GetDatasets().Add(miDataSetDAO,
SourceDataVt, COleVariant("Lat/Long Dataset"),
COleVariant("GEOABBR"), COptionalVariant(), BindLayerVt,
COptionalVariant(), COptionalVariant());

MyDataset.GetThemes().Add();
} catch (COleDispatchException *e) {
    e->ReportError();

    e->Delete();
} catch (COleException *e) {
    e->ReportError();

    e->Delete();
}
}

```

Смотрите также

Метод Datasets.Add

Свойство BindLayer.CoordSys (Объект BindLayer)

Назначение

Свойство CoordSys, определяет систему координат, используется при создании слоя.

Примечание

Когда Вы преобразуете координаты x и y графических объектов или данные о точках, полученные из значений в колонках таблиц, для построения слоя содержащего только точки свойство `BindLayer.CoordSys` определяет систему координат, в которой будет создан слой с точками. Кроме того, в случае связывания данных с координатами x/y , MapX использует эту координатную систему для интерпретации источника данных с координатами x и y . Например, если Вы знаете, что в источнике данных x – и y –координаты в плановых координатах, установите `BindLayer.CoordSys` чтобы использовать плановую систему координат. Если Вы не установите это свойство, то система координат будет использовать свойство `Map.NumericCoordSys`.

Свойство `BindLayer.Filespec` (Объект `BindLayer`)

Назначение

Позволяет Вам указать имя и местоположение файла, так что метод `Datasets.Add` может создать постоянный слой вместо временного слоя.

Примечание

Это свойство применяет связь точечных данных с координатами x/y и данных о точках, полученных из значений в колонках таблиц (Свойство `BindLayer.LayerType` это `miBindLayerTypeXY` или `miBindLayerTypePointRef`).

Если Вы присваиваете путь и имя файла этому свойству, метод `Datasets.Add` создает постоянный слой, хранящийся по тому адресу, который Вы указали. Если Вы не присваиваете это свойство, слой будет временным.

Свойство `BindLayer.KeyLength` (Объект `BindLayer`)

Назначение

Определяет длину колонки символов в слое, создаваемом `Datasets.Add`.

Примечание

Если Вы не присваиваете значение этому свойству, то по умолчанию размер будет 254 символа. Другими словами, результирующий слой будет включать колонку шириной в 254.

Свойство `BindLayer.LayerName` (Объект `BindLayer`)

Назначение

Определяет имя слоя для связанных данных, если `BindLayer.LayerType` является `miBindLayerTypeNormal` или имя вновь созданного слоя, если `BindLayer.LayerType` является `miBindLayerTypeXY` или `miBindLayerTypePointRef`. Это строковая величина.

Смотрите также

`BindLayerTypeConstants`

Свойство `BindLayer.ReferenceLayer`

Свойство `BindLayer.LayerType`

Свойство `BindLayer.RefColumn1`

Свойство `BindLayer.RefColumn1`

Свойство `BindLayer.LayerType` (Объект `BindLayer`)

Назначение

Определяет какой тип данных на слое, который связывается. Связанные данные передаются процессу `DataSet`, который выполняется используя метод `DataSets.Add`. Это свойство дает значение `BindLayerTypeConstant`, как описано ниже.

Обсуждение

Если установлено `miBindLayerTypeNormal`, это означает, что Вы связываете (добавляете) данные к слою в `GeoDictionary`.

Если установлено `miBindLayerTypeXY`, это означает, что данные имеют координаты X и Y и Вы можете создать новый слой с точками. Когда это происходит, Вам надо указать имя вновь создаваемого слоя с точками (`BindLayer.LayerName`), поле данных, содержащее X координату (`BindLayer.RefColumn1`), и поле содержащее Y координату (`BindLayer.RefColumn2`).

Если установлено `miBindLayerTypePointRef`, это означает, что Ваши данные содержат значения, которые вы можете захотеть сравнить с информацией из файла точек. Новый слой создается содержащим точки для каждой строки в исходных данных, при этом происходит сравнение с данными в файле точек. Например, если исходные данные содержат информацию о `ZipCode`, и у Вас есть файл с данными о центроидах этих `ZipCode`, установленный в Вашем `GeoDictionary`, то можно создать новый слой, содержащий точки для каждого `ZipCode`,

содержащегося в ваших данных. Когда Вы это используете, надо указать имя вновь создаваемого слоя точек (`BindLayer.LayerName`), поле в данных, содержащее ссылочные данные. (`BindLayer.RefColumn1`) и имя файла с данными о точках (`BindLayer.ReferenceLayer`). Файл с данными о точках должен быть установлен в `GeoDictionary`.

Смотрите также

Метод `DataSets.Add`

`BindLayerTypeConstants`

Свойство `BindLayer.ReferenceLayer`

Свойство `BindLayer.LayerName`

Свойство `BindLayer.RefColumn1`

Свойство `BindLayer.RefColumn1`

Свойство `BindLayer.RefColumn1` (Объект `BindLayer`)

Назначение

Определяет имя поля или номер поля (начиная с 1) или и то и другое:

- поле, содержащее значение X (долгота) когда `BindLayer.LayerType` является `miBindLayerTypeXY`.
- поле содержащее ссылки, такие как `ZipCode`, когда `BindLayer.LayerType` является `miBindLayerTypePointRef`.

Смотрите также

Объект `BindLayer`

`BindLayerTypeConstants`

Свойство `BindLayer.RefColumn2` (Объект `BindLayer`)

Назначение

Определяет имя поля или номер поля (начиная с 1), содержащего значение координаты Y (долготы) когда `BindLayer.LayerType` является `miBindLayerTypeXY`.

Смотрите также

`BindLayer.LayerType`

`BindLayerTypeConstants`

Свойство `BindLayer.ReferenceLayer` (Объект `BindLayer`)

Назначение

Определяет имя файла содержащего ссылки и используется когда `BindLayer.LayerType` является `miBindLayerTypePointRef`. Слой с ссылками используется при связи с ссылками. Определяется имя файла со ссылками которые используются. Файл ссылок содержит x и y координаты для объектов типа Zip кодов. Файл `ZIPCODES.CPF` включен в поставку с MapX. Свойство `ReferenceLayer` может быть установлено на имя файла содержащего слой или установлено на близкое имя слоя (описание, которое присваивается слою через `geodictionary`).

Связь с ссылками на точки (связь данных типа `miBindLayerTypePointRef`), может быть сделана с использованием любой таблицы как слоя с ссылками, а не только с использованием `CPF` файлов. Определите слой с ссылками как свойство `ReferenceLayer` объекта `BindLayer` обеспечив метод `Datasets.Add`, когда осуществляется ссылка на точечные объекты.

Пример на Visual Basic

```
` To do a ZipCode bind:

Dim MyDataset as Dataset

` Set up the bind Объект Layer

Dim BindLayerObject As New MapXLib.BindLayer

BindLayerObject.LayerName = "Customer's By Zipcode centers"

BindLayerObject.RefColumn1 = 3 ` The third _
    column in my source data contains zip code info

BindLayerObject.LayerType = miBindLayerTypePointRef

BindLayerObject.ReferenceLayer = "US 5 Digit Zipcode Centers"

` The values in the source data column named "GEOABBR" will _
    be used as the geofield for the new point table
```

```

Set MyDataset = Map.Datasets.Add(miDataSetDAO, _
Data2.Recordset, "Zipcode Dataset", "GEOABBR", , _
BindLayerObject)

MyDataset.Themes.Add

```

Пример на C++

```

void CSampleProjectView::ZipcodeBind(CDaoRecordset& SourceDAORecordset)
{
    CMapXDataset MyDataset;

    CMapXBindLayer BindLayerObject;

    COleVariant SourceDataVt, BindLayerVt;

    // Create the BindОбъект Layer
    if(!BindLayerObject.CreateDispatch(BindLayerObject.
GetClsid())) {
        TRACE0("Failed to Create BindОбъект Layer");
        return;
    }

    // Set up all of the Variant paramaters
    SourceDataVt.vt= VT_DISPATCH;
    SourceDataVt.pdispVal = SourceDAORecordset.m_pDAORecordset;

    BindLayerVt.vt = VT_DISPATCH;
    BindLayerVt.pdispVal = BindLayerObject.m_lpDispatch;

    try {

```

```

// Set up the bind Object Layer
BindLayerObject.SetLayerName("Customers By Zipcode Centers");

// The third column in my source data contains zip code info
BindLayerObject.SetRefColumn1(3);

BindLayerObject.SetLayerType(miBindLayerTypePointRef);

BindLayerObject.SetReferenceLayer("US 5 Digit Zipcode Centers");


// The values in the source data column named "GEOABBR"
//will be used as the geofield for the new point table

MyDataset = m_Map.GetDatasets().Add(miDataSetDAO,
SourceDataVt,ColeVariant("Zipcode Dataset"),
ColeVariant("GEOABBR"), COptionalVariant(),
BindLayerVt, COptionalVariant(),COptionalVariant());


MyDataset.GetThemes().Add();
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Смотрите также

BindLayer.LayerType

BindLayerTypeConstants

Свойство `BindLayer.ReferenceLayerField` (Объект `BindLayer`)

Назначение

Определяет какое поле (по номеру, начиная с 1) в таблице `mapinfo` будет связано. Если оно не указано, MapX будет просто смотреть данные по колонкам и выберет ту, у которой наибольшее сходство. Если совпадения нет, то будет выбрана первая колонка.

Примечание

Это свойство дополнительное (необязательное). Оно действует только для `BindLayerTypeNormal` доступно только когда связанный слой указан в свойстве `LayerName`.

Объект `BitmapSymbol` и коллекция `BitmapSymbols`

Объект `BitmapSymbol` представляет один из растровых символов, допустимых в системе пользователя.

Коллекция `BitmapSymbols` представляет установку всех растровых символов в системе пользователя.

Внимание: `BitmapSymbols.Item` это свойство по умолчанию для коллекции `BitmapSymbols`.

Свойства объекта

- Свойство `BitmapSymbol.Name`

Свойства коллекции

- Свойство `BitmapSymbols.Count`
- Свойство `BitmapSymbols.Item`

Методы коллекции

- Метод `BitmapSymbols.Refresh`
- Метод `BitmapSymbols.Unload`

Примеры на C++

Пример 1:

```
// Bitmap Symbols

void CSampleProjectView::CreateBitmapList(CListBox* List) {

    // Fill the ListBox with all of the names of the bitmap
    // symbols. Then, the application could respond to a double
    // click in the listbox by changing the style of some object
    // on the map

    CMapXBitmapSymbols Bitmaps;

    if(!Bitmaps.CreateDispatch(BitmapSymbols.GetClsid())) {

        TRACE0("Failed to create BitmapSymbols Object");

        return;

    }

    try {

        long bmpCount = Bitmaps.GetCount();

        for(long i= 1;i<=bmpCount;i++) {

            List->AddString(BitmapSymbols.Item(i).GetName());

        }

    } catch (COleDispatchException *e) {

        e->ReportError();

        e->Delete();

    } catch (COleException *e) {

        e->ReportError();

        e->Delete();

    }

}
```



```
}
```

Пример 2:

```
void CSampleProjectView::GlobeSymbols() {
    // Override the style of the US Top 20 Cities layer of the
    // US geoset
    // so that each city has the icon of a Globe
    CMapXLayer Top20Cities;

    try {
        Top20Cities = m_Map.GetLayers().Item("US Top 20 Cities");
        Top20Cities.SetOverrideStyle(TRUE);

        // Bitmap Symbols only work on Feature, Layer, and Theme
        // objects. The SupportsBitmapSymbols property will be true
        // if Bitmap symbols work on the style object
        if(Top20Cities.GetStyle().GetSupportsBitmapSymbols()) {
            Top20Cities.GetStyle().SetSymbolType(miSymbolTypeBitmap);
            Top20Cities.GetStyle().SetSymbolBitmapSize(12);
            Top20Cities.GetStyle().SetSymbolBitmapTransparent(TRUE);
            Top20Cities.GetStyle().SetSymbolBitmapName("Glob1-32.bmp");
        }
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

```

    }
}

```

Примеры на Visual Basic

Пример 1:

```

Private Sub Form_Load()

    ' Fill the ListBox with all of the names of the bitmap symbols
    ' Then, the application could respond to a double click _
    ' in the listbox by
    ' changing the style of some object on the map

    Dim Bitmaps As New MapXLib.BitmapSymbols
    For Each symb In Bitmaps
        List1.AddItem symb.Name
    Next

End Sub

```

Пример 2:

```

Private Sub GlobeSymbols_Click()

    'Override the style of the US Top 20 Cities layer of the _
    ' US geoset
    ' so that each city has the icon of a Globe

    Dim Top20Cities As MapXLib.Layer

    Set Top20Cities = Map1.Layers("US Top 20 Cities")
    Top20Cities.OverrideStyle = True

    ' Bitmap Symbols only work on Feature, Layer, and _
    ' Theme objects

```

```

' The SupportsBitmapSymbols property will be true if _
  Bitmap symbols
' work on the style object
If Top20Cities.Style.SupportsBitmapSymbols = True Then
  Top20Cities.Style.SymbolType = miSymbolTypeBitmap
  Top20Cities.Style.SymbolBitmapSize = 12
  Top20Cities.Style.SymbolBitmapTransparent = True
  Top20Cities.Style.SymbolBitmapName = "Glob1-32.bmp"
End If
End Sub

```

Свойство **BitmapSymbol.Name** (Объект **BitmapSymbol**)

Назначение

Строка, открытая только для чтения, представляющая имя файла (типа "filename.bmp"), которая идентифицирует растровый символ.

Примечание

Свойство Name открыто только для чтения. Используйте это свойство, чтобы определить имя растрового символа (*.bmp); Вы можете присвоить это имя свойству SymbolBitmapName объекта Style.

Смотрите также

Bitmap symbols в электронной справке

Свойство **BitmapSymbols.Count** (Коллекция **BitmapSymbols**)

Назначение

Возвращает количество символов в коллекции (количество растров, которые можно использовать как растровые символы). Доступно только для чтения.

Смотрите также

Bitmap symbols

Свойство BitmapSymbols.Item (Коллекция BitmapSymbols)**Назначение**

Возвращает объект BitmapSymbol из коллекции. Доступно только для чтения; использует только числовой индекс. Это свойство по умолчанию для коллекции BitmapSymbols.

Синтаксис

```
[ BitmapSymbol= ]OBJECT.Item (index)
```

Часть Описание

OBJECT Представляет объект BitmapSymbols.

index Целое, показывающее какой объект BitmapSymbol будет возвращен.

Смотрите также

Bitmap symbols в электронной справке

Метод BitmapSymbols.Refresh (Коллекция BitmapSymbols)**Назначение**

Обновляет коллекцию BitmapSymbols, если необходимо, гарантирует, что эта коллекция является текущей.

Примечание

Коллекция BitmapSymbols представляет совокупность .bmp файлов в директории CUSTSYMB. Таким образом, если Вы добавляете .bmp файлы, пока MapX работает, MapX не обновит коллекцию автоматически. Для гарантии того, что коллекция BitmapSymbols является текущей, используйте метод Refresh.

Синтаксис

ОБЪЕКТ.Refresh ()

Смотрите также

Bitmap symbols в электронной справке

Метод BitmapSymbols.Unload (Коллекция BitmapSymbols)**Назначение**

Выгружает все предварительно загруженные файлы .bmr и освобождает все задействованные ресурсы.

Примечание

Когда коллекция BitmapSymbols выходит из поля действия, ресурсы не освобождаются автоматически. Когда Вы работаете используя коллекцию BitmapSymbols, используйте метод Unload для того чтобы наверняка освободить задействованные ресурсы.

Если Вы не используете этот метод для уверенного освобождения задействованных ресурсов, то они могут освободиться когда выгружаются библиотеки MapX (например mitmdl30.dll).

Синтаксис

ОБЪЕКТ.Unload ()

Смотрите также

Bitmap symbols в электронной справке

Объект CoordSys

Объект CoordSys представляет систему координат.

Свойства Объекта

- Свойство `CoordSys.AffineTransform`
- Свойство `CoordSys.Azimuth`
- Свойство `CoordSys.Bounds`
- Свойство `CoordSys.Datum`
- Свойства `CoordSys.FalseEasting, FalseNorthing ()`
- Свойства `CoordSys.OriginLatitude, OriginLongitude`
- Свойство `CoordSys.Range`
- Свойство `CoordSys.ScaleFactor`
- Свойства `CoordSys.StandardParallelOne, StandardParallelTwo`
- Свойство `CoordSys.Type`
- Свойство `CoordSys.Units`

Методы объекта

- Метод `CoordSys.Clone`
- Метод `CoordSys.PickCoordSys`
- Метод `CoordSys.Set`

Примечание

Все свойства предназначены только для чтения. Для изменения системы координат используйте метод `Set` или метод `PickCoordSys`, или назначьте его равным другому объекту `CoordSys`.

Что бы получить объект `CoordSys`, обратитесь к свойствам `Map.DisplayCoordSys`, `Map.NumericCoordSys` или `Layer.CoordSys`.

Свойство `CoordSys.AffineTransform` (Объект `CoordSys`)

Назначение

Свойство, доступное только для чтения; возвращает объект `AffineTransform`.

Свойство CoordSys.Azimuth (Объект CoordSys)

Назначение

Свойство, доступное только для чтения; возвращает величину двойной точности Double представляющую азимут системы координат.

Свойство CoordSys.Bounds (Объект CoordSys)

Назначение

Свойство, доступное только для чтения; возвращает объект прямоугольник, границы которого являются границами действия системы координат.

Метод CoordSys.Clone (Объект CoordSys)

Назначение

Создает копию данной системы координат в виде нового объекта CoordSys.

Синтаксис

```
[ CoordSys= ]OBJECT.Clone ()
```

Свойство CoordSys.Datum (Объект CoordSys)

Назначение

Свойство, доступное только для чтения; возвращает объект Datum (номер региона).

Свойства CoordSys.FalseEasting, FalseNorthing (Объект CoordSys)

Назначение

Свойства, доступные только для чтения; возвращают величины двойной точности, представляющие смещение системы координат на восток /на север.

Примечание

Эти величины определяются свойством CoordSys.Unit.

Свойства CoordSys.OriginLatitude, OriginLongitude (Объект CoordSys)

Назначение

Свойства, доступные только для чтения, возвращают величины (двойной точности) начала координат.

Метод CoordSys.PickCoordSys (Объект CoordSys)

Назначение

Показывает диалог, в котором пользователь может выбрать координатную систему.

Синтаксис

[Boolean=]OBJECT.PickCoordSys (HelpFile, HelpID)

Часть	Описание
--------------	-----------------

OBJECT	Объект CoordSys.
HelpFile	Переменная: Строковая величина, определяющая имя и путь к Windows Help файлу. Необязательная величина; если она пропущена, диалог появится без кнопки Help.
HelpID	Переменная: Целая величина ID, определяющая, какой раздел справки будет показан. Необязательная величина; если она пропущена, диалог появится без кнопки Help.

Примечание

Окно диалога инициализуется автоматически, так что выдвинутое на первое место описание соответствует (matches) объекту CoordSys.
Если пользователь нажимает ОК в окне диалога, метод возвращает True, и объект CoordSys обновляется для подбора координатной системы, которую выбрал пользователь. Если пользователь нажал Cancel, метод возвращает False, и объект CoordSys остается неизменным.

MapX инициализирует список возможных координатных систем, прочитывая файл `marinfow.prj`, который размещается в той же директории, где и `marx30.osx`.

Свойство `CoordSys.Range` (Объект `CoordSys`)

Назначение

Свойство, предназначенное только для чтения; возвращает величину двойной точности, представляющую ранг координатной системы (число от 1 до 180, определяющее, какая часть Земли будет видна - охват).

Свойство `CoordSys.ScaleFactor` (Объект `CoordSys`)

Назначение

Свойство, предназначенное только для чтения; возвращает величину двойной точности, представляющую масштабный фактор системы координат.

Метод `CoordSys.Set` (Объект `CoordSys`)

Назначение

Устанавливает свойства объекта `CoordSys`.

Синтаксис

`OBJECT.Set Type, [Datum], [Units], [OriginLongitude], [OriginLatitude], [StandardParallelOne], [StandardParallelTwo], [Azimuth], [ScaleFactor], [FalseEasting], [FalseNorthing], [Range], [Bounds], [AffineTransform]`

Часть	Описание
<code>object</code>	Объект <code>CoordSys</code> .
<code>Type</code>	Короткая величина <code>CoordSysTypeConstants</code> , такая как <code>miRobinson (12)</code>
<code>Datum</code>	Объект <code>Datum</code> или номер, указывающий какой регион используется (такой как 62 для "NAD 27" datum для

	континентальной части США).
Units	Величина MapUnitConstants, такая как miUnitMeter (7).
OriginLongitude	Величины двойной точности, определяющие широту и долготу
OriginLatitude	начала координат.
StandardParallelOne	Величины двойной точности, определяющие стандартные
StandardParallelTwo	параллели (в градусах широты).
Azimuth	Величина двойной точности, определяющая azimuth в градусах.
ScaleFactor	Величина двойной точности, определяющая масштабный фактор.
FalseEasting	Величины двойной точности, определяющие восточное и
FalseNorthing	северное смещение системы координат (в единицах, определенных в аргументе Units).
Range	Величина двойной точности: Градусы широты от 1 до 180, определяющая охват (какая часть Земли будет видна).
Bounds	Объект Rectangle представляющий внешние ганицы системы координат в единицах, определенных аргументом Units. Требуется, если Type это план-схема (miNonEarth); в других случаях необязательна.
AffineTransformation	Объект AffineTransform; этот аргумент всегда необязательный.

Примечание

Аргумент Type требуется всегда. Все другие аргументы необязательны; эти аргументы могут потребоваться, в зависимости от того, какая проекция/система координат Вами выбрана. Если аргумент не требуется, Вы можете предъявлять любое значение или пропустить его полностью.

Свойства CoordSys.StandardParallelOne, StandardParallelTwo (Объект CoordSys))

Назначение

Свойства, предназначенные только для чтения; возвращают установки стандартных параллелей, значения имеют двойную точность в градусах широты.

Свойство **CoordSys.Type** (Объект **CoordSys**)

Назначение

Свойство, предназначенное только для чтения; возвращает короткое значение, совпадающее с одним из **CoordSysTypeConstants**.

Синтаксис

`object.Type`

Вместо `object` поместите объект **CoordSys**.

Свойство **CoordSys.Units** (Объект **CoordSys**)

Назначение

Свойство, предназначенное только для чтения, возвращает короткое значение, совпадающее с одной из **MapUnitConstants**.

Объект и коллекция **DataSet**

Каждая карта имеет коллекцию **DataSet** - наборов данных. Коллекция **DataSets** имеет методы и свойства, используемые для добавления и удаления объектов **DataSet** из коллекции.

Связывание данных

Связывание данных это процесс привнесения данных из источника данных в **MapX**.

Источником данных может быть Visual Basic data control или источник данных ODBC. В **MapX** данные представляются как объект **DataSet**.

Связывание данных может быть сделано двумя путями. Если у Вас есть Visual Basic для связывания контроля данных, можете использовать свойство **DataSet** объекта карты (**Map**). Или свяжите данные на программном уровне, используя метод **DataSets.Add**. Этот метод требует от Вас сообщить **MapX**, какой источник данных используется, некоторую другую информацию об источнике, и с каким слоем карты осуществляется связь.

Результаты процесса связывания заключаются в создании объекта DataSet. Этот набор данных DataSet, который добавляется к коллекции DataSets, содержит вычисленные значения для объектов на слое карты, с которым осуществляется связь. Например, если данные связаны с картой США, каждый штат будет иметь новые значения данных, которые будут использоваться для контроля за прорисовкой штата. Если источник данных имеет множественные записи для штата (multiple records), значения могут быть просуммированы, осреднены или подсчитаны (counted). DataSet имеет метод Value, который Вы можете использовать для доступа к вычисленным значениям для каждой строчки таблицы, отображенной в виде карты.

DataSource (второй параметр для DataSets.Add) актуален для интерфейса OLE. MapX использует интерфейс для доступа к данным прямо из источника данных. Данные прямо не передаются к DataSets.Add. Существует пять типов источников данных (DataSource): miDataSetDAO – Data Access Object Recordset. Параметром SourceData для DataSets.Add должен быть объект daoRecordset. Вы можете получить его из Visual Basic data control, из формы Access или при создании его в Visual Basic, Access или C++.

miDataSetODBC – источник данных Open DataBase Connectivity. MapX может использовать ODBC для возвращения данных из любого источника ODBC. Вам надо указать строку, описывающую соединение, имя источника данных и строку запроса SQL для выполнения при использовании ODBCQueryInfo.

miDataSetGlobalHandle – это путь для передачи в блок, состоящий из данных, с разделителем табулятором. Параметр SourceData для DataSets.Add должен быть переменной типа VT_I4 и lVal равным global memory handle. Формат каждой строчки данных в блоке глобальной памяти следующий:

```
field TAB field TAB field CRLF
      "      "
```

где поле это строковая величина в кавычках или числовое значение без кавычек и TAB это 0x09, а CRLF это 0x0D 0x0A.

miDataSetOLEData – Это контейнеры, такие как PowerBuilder, которые передают данные в MapX в формате, описанном в miDataSetGlobalHandle, когда инициализируется контроль MapX. Затем вызывается DataSets.Add с типом miDataSetOLEData, который сообщает MapX что надо построить dataset используя ранее переданные данные.

miDataSetUnbound – Если Вы не можете поддержать один из упомянутых выше форматов, то MapX обеспечит запасной выход'. Этот тип может использоваться для установки цикла, в котором MapX опрашивает контейнер с данными, по одной ячейке за раз. MapX будет посылать событие RequestData, с номерами строк и колонок разыскиваемых значений данных.

Затем Вы можете использовать любой метод получения данных и передавать их, посредством этого события обратно в MapX.

Источник данных может иметь много колонок данных. MapX имеет заголовок (overhead) для каждой колонки, которая имеет связь, Вам надо только связать необходимые данные с картой. Используйте последний параметр DataSets.Add (Fields) для установки коллекции полей (колонок), которые Вы хотите связать с картой. Здесь Вы такжу можете указать каким способом обрабатывать множественные данные для одного объекта:

Штат	Продажи
CA	120
NY	100
CA	50
CA	110

В этом примере три продажи для Калифорнии и поскольку только одно значение может быть прикреплено к Калифорнии, Вам надо указать MapX проводить ли суммирование, или осреднение или подсчет (count) продаж.

DataSets.Add имеет следующие автоматизированные элементы поведения:

- определяет какая колонка в источнике данных содержит географическую информацию
- определяет с каким слоем карты осуществляется соединение

Оба из этих автоматизированных элементов поведения не обязательны. Существуют параметры, указывающие, какие колонки в источнике данных содержат географическую информацию, и такие, которые указывают с каким слоем осуществить соединение. Что бы MapX был в состоянии связать данные со слоем карты, должно быть выполнено несколько обстоятельств:

- колонка с ключевыми географическими значениями должна быть индексируется. Большинство карт поступает с индексированными ключевыми колонками. Например, США может иметь как минимум 3 ключевых колонки – имя штата (“New York”), аббревиатура штата (“NY”) и коды FIPS (36). Поставляемая MapInfo карта США имеет первые две колонки в качестве ключевых, так что запрашиваемые из источника данные также имеют или названия штатов или аббревиатуру.
- карта и ее ключевые колонки должны быть указаны в GeoDictionary. Geodictionary это файл (GEODICT.DCT по умолчанию) который указывает путь к этой информации для MapX. Программа MIGM.EXE поставляется вместе с MapX для поддержки GeoDictionary.

Некоторые ключевые колонки из слоя карты могут не содержать уникальных значений. В качестве примера - В США много графств с именами “Warren” или “Washington”. Таким образом, необходимо больше информации, что бы дать точный адрес. Параметр `SecondaryGeoColumn` используется для указания колонки в источнике данных, который содержит необходимую информацию, во избежание путаницы. `GeoDictionary` содержит информацию о том, какие карты имеют неуникальные ключи.

Как только колонка определена как географическая из источника данных и определен слой карты, с которым осуществляется связь. Каждая строка источника данных анализируется. Файл с ссылками, поставляемый с MapX содержит информацию, указывающую где находятся все точки с ссылками. MapX поставляется в файлом ссылок для US ZipCodes, называется он `ZIPCODES.CPF`.

Этот тип связи данных создается с помощью параметра `BindLayer BindLayer` – объект `BindLayer` предьявляется для указания типа связывания и полей из используемого источника данных. С того момента, как данные из источника собраны и хранятся в MapX, несмотря на возможные изменения в самом источнике, MapX не воспроизведет эти изменения. Метод `DataSet.Refresh` может быть использован, что бы заставить MapX прочесть заново источник данных и собрать заново данные и хранить их. Это занимает много времени. `DataSet.Refresh` не работает для наборов данных которые создают новые слои с точками – подходящий путь для добавления данных к этим слоям состоит из 2-х шагов:

1. создание слоя точек, где каждая точка имеет ключевое значение
2. связывание данных с этим вновь созданным слоем точек (используя `DataSets.Add` снова). Второй набора данных `DataSet` может быть обновлен.

Свойства объекта

- Свойство `DataSet.Fields` (Объект `DataSet`)
- Свойство `DataSet.GeoField` (Объект `DataSet`)
- Свойство `DataSet.Layer` (Объект `DataSet`)
- Свойство `DataSet.Name` (Объект `DataSet`)
- Свойство `DataSet.RowCount` (Объект `DataSet`)
- Свойство `DataSet.ReadOnly` (Объект `Dataset`)
- Свойство `DataSet.SecondaryGeoField` (Объект `DataSet`)
- Свойство `DataSet.SourceRows` (Объект `DataSet`)
- Свойство `DataSet.Themes` (Объект `DataSet`)
- Свойство `DataSet.Type` (Объект `Dataset`)

Методы объекта

- Метод `DataSet.AddField` (Объект `DataSet`)

- Метод DataSet.Refresh (Объект DataSet)
- Метод DataSet.RowValues (Объект DataSet)
- Метод DataSets.Restore (Коллекция DataSets)
- Метод DataSet.Value (Объект DataSet)

Свойства коллекции

- Свойство Datasets.BuildSourceRows (Коллекция Datasets)
- Свойство DataSets.Count (Коллекция DataSets)
- Свойство DataSets.Item (Коллекция DataSets)

Методы коллекции

- Метод DataSets.Add (Коллекция DataSets)
- Метод DataSets.Remove (Коллекция DataSets)
- Метод DataSets.RemoveAll (Коллекция DataSets)

Метод DataSet.AddField (Объект DataSet)

Назначение

Позволяет полю (' колонке') быть добавленным к dataset, это выражение содержащее функции, операторы и поля набора данных (только из текущего набора данных). Метод добавляет новое поле к коллекции набора данных для подписывания, тематических слоев и др..

Синтаксис

ОБЪЕКТ.AddField (Name, Expression)

Часть	Описание
ОБЪЕКТ	Это объект DataSet.
Name	Имя нового поля.
Expression	Созданное выражение.

Примечание

Новое поле будет иметь тип или miTypeNumeric или miTypeString в зависимости от возвращаемого типа выражения.

Новое поле будет иметь тип объединения (близких значений) miAggregationSum для числовых полей и miAggregationIndividual для строковых полей.

Поле добавляется в конец коллекции полей для набора данных и является временным. Это означает, что по завершении сеанса работы поле исчезнет.

Свойство `Datasets.BuildSourceRows` (Коллекция `Datasets`)

Назначение

Наборы данных `Datasets` созданные из коллекции `Datasets` с этим свойством, установленным на `TRUE`, поддерживают метод `Dataset.GetSourceRows`. Те наборы данных созданные из коллекции `Datasets` с этим свойством, установленным на `FALSE` не поддерживают этот метод, и вызывают `Datasets.GetSourceRows` тех объектов `Dataset` которые образуются в результате исключения.

Примечание

Имеющиеся наборы данных или поддерживают или не поддерживают метод `Datasets.GetSourceRows` в течение всего времени своего существования. Изменение свойств коллекции наборов данных не действует на объекты наборов данных, созданных ранее из коллекции наборов данных.

Свойство `DataSet.Fields` (Объект `DataSet`)

Назначение

Сбор полей `Fields` для набора данных.

Смотрите также

Объект и коллекция `Field`

Свойство `DataSet.GeoField` (Объект `DataSet`)

Назначение

Номер колонки с географическими ключами в коллекции `Fields`. Это целое значение и свойство открытое только для чтения.

Свойство DataSet.Layer (Объект DataSet)

Назначение

Связывает объект Layer этого набора данных.

Примечание

Когда добавлен набор данных DataSet, данные всегда прикрепляются к слою. Это свойство указывает на слой.

Слой может иметь более чем один набор данных, связанный с ним. Когда слой удаляется, все наборы данных, с ним связанные, тоже удаляются.

Пример на C++

```
// DataSet.Layer Property
// LabelProperties.Dataset Property
// LabelProperties.DataField Property
void CSampleProjectView::LabelLayer() {
    // Label the Layer that "My Dataset" is bound to with
    // the data from My Dataset's first column
    try {
        CMapXDataset myDataset = m_Map.GetDatasets().Item("My Dataset");
        myDataset.GetLayer().GetLabelProperties().SetDataset(myDataset) ;
        myDataset.GetLayer().GetLabelProperties().SetDataField(1);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

```
}
```

Пример на Visual Basic

```
Private Sub LabelLayer()
    ' Label the Layer that "My Dataset" is bound to with
    ' the data from My Dataset's first column
    Dim myDataset as MapXLib.Dataset
    Set myDataset = Map1.Datasets.Item("My Dataset")
    myDataset.Layer.LabelProperties.Dataset = myDataset
    myDataset.Layer.LabelProperties.DataField = 1
End Sub
```

Свойство DataSet.Name (Объект DataSet)

Назначение

Имя набора данных. Это строковая величина.

Примечание

Имя это путь ссылающийся на этот набор данных, когда используются такие методы, как DataSets.Item или DataSets.Remove. Имя должно быть уникальным, в наборах данных больше таких имен не должно быть.

Метод DataSet.Refresh (Объект DataSet)

Назначение

Обновляет данные из источника.

Синтаксис

```
DataSet.Refresh
```

Примечание

Этот метод используется для обновления карты, если исходные данные изменялись.

Когда метод Refresh запускается, данные пересчитываются из исходного источника данных и собираются заново используя методы сборки, определенные в наборе данных.

Любые тематические слои, базирующиеся на наборе данных будут обновлены при изменении исходных данных.

Внимание: Этот метод не работает с наборами данных типа `miBindLayerTypeXY`

Пример

```
`Create the default theme on the dataset
ds.Themes.Add
Map1.Refresh
```

Метод Dataset.RowValues (Объект Dataset)

Назначение

Определяет, какой будет строка: длинным строчным ID, строковым серийным ключом или объектом Feature, получающим ключ извне.

Синтаксис

```
[RowValues=]OBJECT.RowValues(Row)
```

Части Описание

OBJECT Это объект Dataset.

Row Переменная: Может быть FeatureID (целое), FeatureKey (строковая), или объект Feature.

Свойство DataSet.RowCount (Объект DataSet)

Назначение

Возвращает количество строк в наборе данных. Это целое значение. Свойство открыто только для чтения.

Примечание

С того момента, как данные связаны со слоем, RowCount действительно возвращает количество объектов на слое. Таким образом, счетчик строк набора данных может иметь нулевое значение, если нет данных, которые были бы связаны.

Свойство Dataset.ReadOnly (Объект Dataset)

Назначение

Свойство, доступное только для чтения, показывающее можно ли обновлять набор данных или нет. В данное время только наборы данных miDatasetLayer являются обновляемыми. Наборы данных обновляются при использовании Layers.UpdateFeature или Feature.Update с коллекцией RowValues.

Смотрите также

Объект RowValue и коллекция RowValues

Метод Layer.UpdateFeature

Метод Feature.Update

Метод DataSets.Restore (Коллекция DataSets)

Назначение

Этот метод может быть использован для восстановления набора данных из карты, которая хранится в файле с использованием OLE. Если набор данных содержит внешние данные, содержание набора данных полностью не восстановится. DataSets.Restore может использоваться для восстановления набора данных (и связанных с ними тематических слоев) при передаче карты во внешние данные.

Синтаксис

```
[ DataSet = ]OBJECT.Restore ( Name, SourceData )
```

Часть	Описание
-------	----------

OBJECT	Коллекция наборов данных DataSets.
--------	------------------------------------

Name	Строковый параметр. Это имя набора данных, подлежащего восстановлению; это должно быть то имя, которое существовало, когда карта создавалась.
SourceData	Переменная. Это должен быть тот же параметр, который был, когда сохраняемый набор данных DataSet был добавлен к исходному DataSets.Add. Этот источник данных затем использовался для восстановления содержания набора данных.

Смотрите также

Метод DataSets.Add

Свойство DataSet.SecondaryGeoField (Объект DataSet)

Назначение

Номер колонки являющейся второй ключевой колонкой в коллекции колонок Fields. Будет иметь значение 0 если не было определено SecondaryGeoField, когда добавлялись данные. Это целая величина и открыто свойство только для чтения.

Свойство DataSet.SourceRows (Объект DataSet)

Назначение

Коллекция SourceRows содержащая список строк из источника данных, которые были сопоставлены с указанной строкой (specified row).

Смотрите также

Объект SourceRow и коллекция SourceRows

Свойство Feature.FeatureKey

Свойство DataSet.Themes (Объект DataSet)

Назначение

Коллекция Themes для набора данных.

Смотрите также

Объект и коллекция Theme

Свойство DataSet.Type (Объект DataSet)

Назначение

Свойство открытое только для чтения, возвращающее тип набора данных. Величина этого свойства будет одним из значений DataSetTypeConstants.

Пример на Visual Basic

```
Dim ds As MapXLib.Dataset
Set ds = Map1.Datasets(1)
If ds.Type = miDataSetODBC Then
    ' ... then this is an ODBC dataset
ElseIf ds.Type = miDataSetDAO Then
    ' ... then this is a DAO dataset
End If
```

Свойство DataSet.Value (Объект DataSet)

Назначение

Возвращает значения для определенных строк и колонок в наборе данных. Это свойство предусмотрено по умолчанию для объекта DataSet.

Синтаксис

[value =] OBJECT.Value (Row, Column)

Часть Описание

OBJECT	Это объект DataSet.
Row	Строка в MapX для которой получается значение. Переменная: может быть: *Ключ (Key of the feature) *Имя (Name of the Feature) *объект Feature
Column	Переменная: может быть именем колонки или индексом.

Примечание

Величины, возвращаемые из набора данных MapX DataSet, не являются оригинальными исходными данными. С того момента, как исходные данные собраны и связаны со слоем для формирования набора данных MapX, номера строк соотносятся с объектами на слое Layer, а не с номерами строк исходных оригинальных данных. Таким образом, ячейки в набора данных могут иметь нулевые значения, если нет данных, связанных с объектами. Нулевое значение возвращается в переменную при установке типа переменной VT_NULL.

Вы можете использовать объект Feature как параметр Row, с того момента, как объект однозначно идентифицирует одну строку в Layer который связан с набором данных DataSet.

Пример на C++

```
// Dataset.Value Method

void CSampleProjectView::QueryFeature(CTreeCtrl&
queryResults,CMapXFeature& ftr,CMapXDataset& srcData) {

    // This function builds a tree control with all of the values in the
    dataset

    // Example output:

    // Cheyenne

    //     Capital: Cheyenne

    //     State: WY
```

```

//      FIPS_Code: 56
//      Pop_1990: 50008
//      Num_HU_90: 21859
CString fieldValue;
CString fieldName;
long fieldCount;
COleVariant ftrVt, valueVt;
HTREEITEM parentItem;

// First, use the feature's name as the parent node
// (all of the value nodes will be children of this node)
parentItem = queryResults.InsertItem(ftr.Name);

ftrVt.vt = VT_DISPATCH;
ftrVt.pdispVal = ftr;

try {
    fieldCount = srcData.GetFields().GetCount();
    // loop over all of the fields in the dataset
    for(long i = 1; i <= fieldCount; i++) {
        // Fetch the value from the dataset and add it to the Tree control
        valueVt = srcData.GetValue(ftrVt, i);
        if(valueVt.vt != VT_NULL) { // There is a value for the field
            valueVt.ChangeType(VT_BSTR);
            fieldValue = valueVt.bstrVal;
            SysFreeString(valueVt.bstrVal);
        }
    }
}

```



```

        // Add the string "Field: Value" to the tree
        fieldLine.Format("%s:
%s", (LPCSTR)srcData.GetFields().Item(i).GetName(), (LPCSTR)field Value);
        queryResults.InsertItem(fieldLine, parentItem);
    }
}
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Пример на Visual Basic

```

Private Sub QueryFeature(ftr As MapXLib.Feature, srcData As _
MapXLib.Dataset)
    ' This function builds a tree control with all of the _
    values in the dataset
    ' Example output:
    ' Cheyenne
    '   Capital: Cheyenne
    '   State: WY
    '   FIPS_Code: 56
    '   Pop_1990: 50008

```

```

'    Num_HU_90: 21859

Dim value As Variant

Dim parentItem As Node

' First, use the feature's name as the parent node
' (all of the value nodes will be children of this node)
Set parentItem = TreeView1.Nodes.Add(, , ftr.Name, ftr.Name)

' loop over all of the fields in the dataset
For Each Field In srcData.Fields
    ' Fetch the value from the dataset and add it to the _
    Tree control
    value = srcData.value(ftr, Field.Name)
    ' Add the string "Field: Value" to the tree
    TreeView1.Nodes.Add parentItem, tvwChild, parentItem.Key _
    & Field.Name & value, Field.Name & ": " & value
Next
End Sub

```

Смотрите также

Свойство Feature.FeatureKey

Свойство Feature.Name

Метод DataSets.Add (Коллекция DataSets)

Назначение

Создает определенный набор данных и добавляет его в коллекцию.

Синтаксис

`DataSets.Add (Type, SourceData, [Name], [Geofield], [SecondaryGeofield], [BindLayer], [Fields], [Dynamic])`

Часть	Описание
Type	Тип набора данных, который был добавлен. Берет значение <code>DataSetTypeConstants</code> .
SourceData	Это ссылка на данные и она разная в зависимости от типа Type – <code>IDAORRecordset</code> , <code>ODBCQueryInfo</code> , объект <code>Layer</code> или ничего.
Name	Строка, однозначно определяющая набор данных. Это необязательный параметр, и если он не определен, используется имя в форме <code>DataSetN</code> .
Geofield	Имя или индекс колонки, которая содержит географическую информацию. Это необязательный параметр, и если он не определен, <code>MapX</code> ищет по всем полям, что бы найти , похожее на то, что описано в <code>GeoDictionary</code> .
SecondaryGeofield	Имя или индекс колонки, которая содержит уточненную географическую информацию. Это необязательный параметр.
BindLayer	Определяет слой карты, к которому будут прикреплены данные, или используется <code>BindLayerObject</code> если входящие данные будут преобразованы в специальный файл (point reference file) (такой как <code>ZipCodes</code>) или они содержат значения долгота/широта. Это необязательный параметр, и если он не определен, <code>MapX</code> ищет слой, к которым будут прикреплены данные в <code>GeoDictionary</code> .
Fields	Это объект <code>Fields</code> который является коллекцией объектов <code>Field</code> . Объекты эти используются для описания того, какие поля из исходных данных будут использоваться и какая функция объединения данных используется, если одному объекту на карте сопоставляется несколько записей. Это необязательный параметр, если он не определен, считываются все колонки, а в случае нескольких записей сопоставленных одному объекту, производится суммирование значений.
Dynamic	Переменная: Логическая величина, которая определяет, являются ли связываемые данные динамическими. Это необязательный параметр, если он пропущен, то по умолчанию будет <code>False</code> , это означает, что связываемые данные статичны (то есть, <code>MapX</code> будет копировать

необходимые данные, когда база данных открыта). Если вы определите True, MapX будет получать доступ к данным напрямую, только когда данные потребуются (например при подписывании объектов). Если Вы указываете True но набор данных не поддерживает динамические колонки, исключение будет отвергнуто.

Внимание:

- Если существует двусмысленность (типа нескольких наборов данных), пользователь может установить обработчик событий. Если обработчик событий не установлен, MapX выберет первый набор данных. Смотрите описания события ResolveDataBind.
- Тип набора данных: miDatasetLayer – используется для создания набора данных, который ссылается на поля из таблицы MapInfo. Параметр SourceData это объект Layer.

Примечание

Если используется автоматическое определение логического условия, может быть найдено более чем одно совпадение. Если Вы хотите обработать этот случай, Вы можете использовать обработчик событий ResolveDataBind. С другой стороны, MapX выберет первый попавшийся из совпавших.

SecondaryGeofield требуется, когда слой, который в наборе данных используется для связывания, имеет ключевую колонку, которая не является уникальной. В этом случае используется вторая колонка с ключевыми значениями.

Колонка GeoField должна быть уникальной, если создается новый слой точек (BindLayer.LayerType является miBindLayerTypeXY или miBindLayerTypePointRef). Колонка GeoField будет использоваться, чтобы дать имена объектам на новом слое точек.

Повторяющиеся значения будут сведены в одну точку, а сами значения будут суммироваться.

Если определена коллекция Fields, то параметры Geofield SecondaryGeofield будут ссылаться на колонки в коллекции Fields а не исходные данные.

Пример на Visual Basic

```
`This sample demonstrates the simplest use of the Datasets.Add
`method. With this usage, you specify only the data source and
`type. MapX automatically determines which column represents
`the geographic information, as well as which layer to bind that
`information to.
```

`Also demonstrated here is the simplest use of the Themes.Add method. In this usage, where all arguments are omitted, MapX automatically determines a name for the theme, which column to use to create the theme, and what type of theme to create. In this case, MapX will create a ranged theme on the first numeric column in the USA table (TOTPOPHIS).

```
Private Sub Command1_Click()
```

```
    Dim db As Database
```

```
    Dim rs As Recordset
```

```
    Dim ds As Dataset
```

```
    `First, get the table that will contain our source data
```

```
    `Set db = DBEngine.Workspaces(0).OpenDatabase
```

```
    `"C:\Program Files\MapInfo MapX\Data\Mapstats.mdb")
```

```
    `Set rs = db.OpenRecordset("USA")
```

```
    `This line will create a dataset named "Dataset1"
```

```
    `Set ds = Map1.Datasets.Add(miDataSetDAO, rs)
```

```
    `Create the default theme on the dataset
```

```
    `ds.Themes.Add
```

```
End Sub
```

`This sample demonstrates the use of Datasets.Add and the

`RequestData event supported by MapX to build an unbound

`dataset, which allows MapX to access data whose format is known

`only to the programmer. In this example, the data is in the form
 `of a two-dimensional array in VB.

```
Private Const kNumberOfRows = 3
Dim theData(1 To 3, 1 To 2) As Variant
Private Sub Form_Load()
    theData(1, 1) = "ME"           `Fill in the data to
    theData(2, 1) = "NH"           `be used by RequestData
    theData(3, 1) = "VT"
    theData(1, 2) = 100
    theData(2, 2) = 200
    theData(3, 2) = 300
    Map1.ZoomTo 800, -70.26, 44.05 `Zoom in on New England
End Sub
Private Sub Command1_Click()
    Dim flds As New MapXLib.Fields
    Dim ds As Dataset
    `Describe the structure of the unbound dataset:
    flds.Add "State", "State", miAggregationIndividual, _
    miTypeString
    flds.Add "Sales", "Sales", miAggregationSum, miTypeNumeric
    `Create the unbound dataset. The "RequestData" event will be
    `triggered to get the data to be used.
    Set ds = Map1.Datasets.Add(miDataSetUnbound, Nothing, _
    "My Dataset", "State", , "USA", flds)
    `Create a theme based on the "Sales" column in the
```

```

`unbound dataset
ds.Themes.Add miThemeGradSymbol, "Sales", "My Theme"
End Sub

Private Sub Map1_RequestData(ByVal DataSetName As String, _
ByVal Row As Long, ByVal Field As Integer, _
Value As Variant, Done As Boolean)
    Done = False
    If DataSetName <> "My Dataset" Or Row > kNumberOfRows Then
        Done = True
    Else
        Value = theData(Row, Field)
    End If
End Sub

```

Пример на Visual Basic (Safe Array)

```

Private Sub AddData_Click()
    ' Dim the array that will hold my data
    Dim MyData(5 To 10, 2 To 4) As Variant

    ' Now populate the array
    MyData(5, 2) = "ny"
    MyData(6, 2) = "ny"
    MyData(7, 2) = "ca"
    MyData(8, 2) = "tx"
    MyData(9, 2) = "nc"
    MyData(10, 2) = "sc"

```

```

MyData(5, 3) = 100
MyData(6, 3) = 300
MyData(7, 3) = 123
MyData(8, 3) = 345
MyData(9, 3) = 455
MyData(10, 3) = 1231
MyData(5, 4) = 10000
MyData(6, 4) = 2000000
MyData(7, 4) = 30000
MyData(8, 4) = 12300
MyData(9, 4) = 34500
MyData(10, 4) = 45500

' add a dataset
Map1.Datasets.Add miDatasetSafeArray, MyData

' create a default theme
Map1.Datasets(1).Themes.Add

End Sub

```

Пример на C++

```

void CSampleProjectView::AddSet(CDaoRecordset& rs) {
    // This sample demonstrates the simplest use of the
    // Datasets.Add method. With this usage, you specify only the
    // data source and type. MapX automatically determines which
    // column represents the geographic information, as well as

```



```

// which layer to bind that information to. Also demonstrated
// here is the simplest use of the Themes.Add method. In this
// usage, where all arguments are omitted, MapX automatically
// determines a name for the theme, which column to use to
// create the theme, and what type of theme to create. In
// this case, MapX will create a ranged theme on the first
// numeric column in the USA table (TOTPOPHIS)
// This example assumes you have a DAO recordset of valid data
// open in the rs parameter
ColeVariant rsVariant;
CMapXDataset ds;
rsVariant.vt = VT_DISPATCH;
rsVariant.pdispVal = rs.m_pDAORecordset;

try {
    // Create a dataset named "Dataset1"
    ds = m_Map.GetDatasets().Add(miDataSetDAO,rsVariant);
    // Create the default theme on the dataset
    ds.GetThemes().Add();
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}

```

```

}

// This sample uses Datasets.Add and the RequestData event to
// build an unbound dataset, which allows MapX to access data
// whose format is known only to the programmer.

struct StateData { char name[3]; int value; };
StateData MapData[3] = {
    {"ME",100},
    {"NH",200},
    {"VT",300}};

void CSampleProjectView::Start() {
    ColeVariant FieldsVariant;
    CMapXFields flds;
    CMapXDataset ds;

    // Create a new, empty fields collection
    if(!flds.CreateDispatch(flds.GetClsid())) {
        TRACE0("Failed to Create Fields Collection");
        return;
    }

    try {
        flds.Add(ColeVariant("State"),ColeVariant("State"),
            ColeVariant((long)miAggregationIndividual),
            ColeVariant((long)miTypeString));
    }
}

```

```

flds.Add(ColeVariant("Value"),ColeVariant("Value"),
ColeVariant((long)miAggregationSum),
ColeVariant((long)miTypeNumeric));

FieldsVariant.vt = VT_DISPATCH;

FieldsVariant.pdispVal = flds.m_lpDispatch;

ds = m_Map.GetDatasets().Add(miDataSetUnbound,
COptionalVariant(),ColeVariant("My Dataset"),
ColeVariant("State"),COptionalVariant(),
ColeVariant("USA"),FieldsVariant,COptionalVariant());

ds.GetThemes().Add(miThemeGradSymbol,"Sales","My Theme");
} catch (ColeDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (ColeException *e) {
    e->ReportError();
    e->Delete();
}
}

void CSampleProjectView::OnRequestData(LPCTSTR DataSetName, long Row,
short Field, VARIANT FAR* Value, BOOL FAR* Done) {
    *Done = FALSE;

    if(!strcmp(DataSetName,"My Dataset") || Row > 3)
        *Done = TRUE;

    else {
        if(Field == 1) {

```

```

        Value->vt = VT_BSTR;

        Value->bstrVal = CString(MapData[Row].name).
        AllocSysString();

    } else {

        Value->vt = VT_INT;

        Value->iVal = MapData[Row].value;

    }

}

}

```

Смотрите также

Событие ResolveDataBind

Объект Field и коллекция Fields

Свойство DataSets.Count (Коллекция DataSets)

Назначение

Подсчитывает количество объектов DataSet в коллекции. Это свойство открытое только для чтения.

Пример на Visual Basic

```

Dim iCount as Integer

iCount = Map1.Datasets.Count

Debug.Print "There are " & iCount & "Datasets attached to _
the map."

- или -

Debug.Print "There are " & Map1.Datasets.Count & "Datasets _
attached" _ & " to the map."

```

Пример на C++

```

void CSampleProjectView::CountDataset() {
    CString msg;

    try {
        msg.Format("There are %d Datasets attached to the
map.", m_Map.GetDatasets().GetCount());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }

    AfxMessageBox(msg);
}

// Datasets.Item Property
CMapXDataset CSampleProjectView::GetMyDataset() {
    try {
        return m_Map.GetDatasets().Item("My Dataset");
        // or return m_Map.GetDatasets().Item(1);
    } catch (COleDispatchException *e) {
        e->ReportError();
    }
}

```

```

        e->Delete();
    } catch (ColeException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Свойство **DataSets.Item** (Коллекция **DataSets**)

Назначение

Возвращает отдельный объект DataSet из коллекции. Это переменная, и Вы можете задать имя набора данных или индекс в виде номера. По умолчанию это свойство для коллекции DataSets.

Примеры

Visual Basic:

```

Dim ds As DataSet
Set ds = Map1.Datasets("My Dataset")    or   Set ds = Map1.Datasets(1)

```

Delphi:

```

Var ds : variant;
Begin
    ds := Map1.Datasets.Item("MyDataset");  or   ds :=
        Map1.Datasets.Item(1);
End;

```

Метод **DataSets.Remove** (Коллекция **DataSets**)

Назначение

Удаляет определенный объект DataSet из коллекции DataSet.

Внимание: Если Вы удаляете объект, коллекция индексов перенумеровывается, заполняя возникшую брешь в номерах.

Синтаксис

`DataSets.Remove(index)`

Часть	Описание
--------------	-----------------

index	имя набора данных DataSet или индекс от 1.
-------	--

Примечание

Все тематические слои (themes), базирующиеся на этом наборе данных так же удаляются.

Пример на Visual Basic

```
Map1.Datasets.Remove "My Dataset"
```

- or -

```
Map1.Datasets.Remove 1
```

Пример на C++

```
void CSampleProjectView::RemoveMyDataset() {
    try {
        m_Map.GetDatasets().Remove("My Dataset");
        // or m_Map.GetDatasets().Remove(1);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Метод **DataSets.RemoveAll** (Коллекция **DataSets**)

Назначение

Удаляет все объекты Dataset из коллекции.

Синтаксис

DataSets.RemoveAll

Пример на Visual Basic

```
Private Sub btnDatasetsRemoveAll_Click()
    Dim nDataSets As Integer
    Dim ds As Dataset
    nDataSets = Map1.Datasets.Count
    Set ds = Map1.Datasets.Add(miDataSetLayer, Map1.Layers(1))
    nDataSets = Map1.Datasets.Count
    ' remove all of the datasets
    Map1.Datasets.RemoveAll

End Sub
```

Пример на C++

```
void CSampleProjectView::RemoveAllDatasets() {
    CMapXLayer lyr;
    COleVariant lyrVariant;
    CMapXDataset ds;

    try {
        lyr=m_Map.GetLayers().Item(1);
```



```

    lyrVariant.vt = VT_DISPATCH;
    lyrVariant.pdispVal = lyr.m_lpDispatch;

    // Add a dataset from a layer
    m_Map.GetDatasets().Add(miDataSetLayer, lyrVariant);

    // Now remove all datasets
    m_Map.GetDatasets().RemoveAll();
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Объект Datum

Объект Datum представляет Datum - номер региона для системы координат, используемый объектом CoordSys.

Методы объекта

- Метод Datum.Set
- Метод Datum.SetFromList

Свойства объекта

- Свойство Datum.Eccentricity
- Свойство Datum.Ellipsoid
- Свойство Datum.Flattening
- Свойство Datum.PrimeMeridian
- Свойства Datum.RotateX, RotateY, RotateZ
- Свойство Datum.ScaleAdjus
- Свойства Datum.SemiMajorAxis, SemiMinorAxis
- Свойства Datum.ShiftX, ShiftY, ShiftZ

Примечание

Все свойства открыты только для чтения. Что бы изменить Datum, используйте метод Set или метод SetFromList.

Что бы получить объект Datum, в объекте CoordSys обратитесь к свойству Datum.

Свойство Datum.Eccentricity (Объект Datum)

Назначение

Свойство открыто только для чтения; возвращает величину двойной точности, это эксцентриситет региона.

Примечание

В уравнениях эксцентриситет обычно обозначается буквой e .

Свойство Datum.Ellipsoid (Объект Datum)

Назначение

Свойство открыто только для чтения; возвращает величину - короткое целое, соответствующую идентификатору эллипсоида региона, типа 28 для "WGS 84."

Свойство `Datum.Flattening` (Объект `Datum`)

Назначение

Свойство открыто только для чтения; возвращает число двойной точности, представляющее выполаживание региона.

Примечание

В уравнениях выполаживание (flattening) всегда отображается символом f .

Свойство `Datum.PrimeMeridian` (Объект `Datum`)

Назначение

Свойство открыто только для чтения; возвращает число двойной точности, представляющее долготу первого меридиана региона, в градусах к востоку от Гринвича.

Примечание

Свойство `PrimeMeridian` обычно равно 0, поскольку большинство datum используют Гринвичский меридиан в качестве первого. Таким образом, некоторые datums используют другие места в качестве первого меридиана. Например, NTF datum использует Париж как первый меридиан, который имеет 2.33722917 градусов к востоку от Гринвича. Если Вы используете NTF datum в системе координат, все значения долготы в такой системе координат будут относительно Парижа вместо Гринвича.

Свойства `Datum.RotateX`, `Datum.RotateY`, `Datum.RotateZ` (Объект `Datum`)

Назначение

Свойство открыто только для чтения; возвращает число двойной точности, представляющее угол, в градусах-секундах, на который повернут эллипсоид региона вокруг каждой из его осей.

Свойство Datum.ScaleAdjust (Объект Datum)

Назначение

Свойство открыто только для чтения; возвращает число двойной точности, представляющее корректировку масштабного фактора региона (в миллионных долях, приспособленную к размеру эллипсоида).

Примечание

В уравнениях, корректировка масштабного фактора обычно изображается буквой *m*.

Свойства Datum.SemiMajorAxis, SemiMinorAxis (Объект Datum)

Назначение

Свойство открыто только для чтения; возвращает число двойной точности, представляющее главные полуоси - большую и малую. Длина в метрах.

Примечание

В уравнениях большая ось обозначается буквой “*a*”, а малая ось обозначается буквой *b*.

Метод Datum.Set (Объект Datum)

Назначение

Устанавливает свойства, которые входят в объект Datum.

Синтаксис

OBJECT.Set (Ellipsoid, ShiftX, ShiftY, ShiftZ, RotateX, RotateY, RotateZ, ScaleAdjust, PrimeMeridian)

Часть

Описание

OBJECT

Объект Datum.

Ellipsoid

Короткое значение; должно быть номером эллипсоида (таким как 28 для эллипсоида ”WGS 84”).

ShifX, ShiftY, ShiftZ	Величины двойной точности, это расстояния в метрах на которые осуществлен сдвиг вдоль каждой из осей.
RatateX, RotateY, RotateZ	Величины двойной точности, это углы (в градусах-секундах) на которые повернут эллипсоид вдоль каждой из осей.
ScaleAdjust	Величина двойной точности, представляющая корректировку масштабного фактора (количество, в миллионных долях, от размера эллипсоида).
PrimeMeridian	Величина двойной точности, представляющая долготу первого меридиана. Обычно равна нулю; смотрите раздел PrimeMeridian.

Примечание

Все аргументы необходимо задавать.

Смотрите также

Метод Datum.SetFromList

Метод Datum.SetFromList (Объект Datum)

Назначение

Устанавливает все свойства объекта Datum для указанного региона из списка регионов.

Синтаксис

ОБЪЕКТ.Set DatumNumber

Часть	Описание
--------------	-----------------

ОБЪЕКТ	Объект Datum.
--------	---------------

DatumNumber	Короткое целое; должно быть одним из номеров поддерживаемых регионов (типа 62 для "NAD 27" для континентальной части США).
-------------	--

Свойства Datum.ShiftX, ShiftY, ShiftZ (Объект Datum)

Назначение

Свойство открыто только для чтения; возвращает число двойной точности, представляющее расстояние в метрах на которое эллипсоид сдвинут вдоль каждой из его осей.

Объект и коллекция Feature

Каждый слой Layer имеет коллекцию выбранных объектов Feature (Layer.Selection). Выборка является коллекцией особенностей объектов. Объекты Feature сообщают такие особенности слоя, как заголовок, например, New York или Chicago. Коллекция выборок имеет методы и свойства для осуществления разных типов выбора.

Внимание: Если выбрана какая-либо совокупность объектов карты, и одновременно сделана выборка с другого слоя, то выделение в первом случае снимется. Например, в наборе данных "World Countries", выделите United States и затем, нажав клавишу control, выделите океан. Выделение с United States снимется. Однако эти оба объекта попадут в Selection Collections для соответствующих слоев.

Коллекция Features представляет коллекцию объектов Feature из одного слоя. Все объекты в такой коллекции обязательно должны быть с одного слоя. Вы можете получить объект при вызове одного из методов поиска на слое (Layer Search), который возвращает коллекцию Features. Объект Features будет содержать объекты, которые были на слое в момент вызова метода поиска. Добавление или удаление объектов на слое не изменит уже существующую коллекцию автоматически.

Свойства объекта

- Свойство Feature.Area (Объект Feature)
- Свойство Feature.Bounds (Объект Feature)
- Свойство Feature.Caption (Объект Feature)
- Свойство Feature.CenterX, CenterY (Объект Feature)
- Свойство Feature.FeatureID (Объект Feature)
- Свойство Feature.FeatureKey (Объект Feature)

- Свойство `Feature.KeyValue` (Объект `Feature`)
- Свойство `Feature.LabelPoint` (Объект `Feature`)
- Свойство `Feature.Layer` (Объект `Feature`)
- Свойство `Feature.Length` (Объект `Feature`)
- Свойство `Feature.Name` (Объект `Feature`)
- Свойство `Feature.Nodes` (Объект `Feature`)
- Свойство `Feature.Parts` (Объект `Feature`)
- Свойство `Feature.Perimeter` (Объект `Feature`)
- Свойство `Feature.Point` (Объект `Feature`)
- Свойство `Feature.Smooth` (Объект `Feature`)
- Свойство `Feature.Style` (Объект `Feature`)
- Свойство `Feature.Type` (Объект `Feature`)

Методы объекта

- Метод `Feature.Attach` (Объект `Feature`)
- Метод `Feature.Clone` (Объект `Feature`)
- Метод `Feature.Offset` (Объект `Feature`)
- Метод `Feature.Update` (Объект `Feature`)

Свойства коллекции

- Свойство `Features.Bounds` (Коллекция `Features`)
- Свойство `Features.Count` (Коллекция `Features`)
- Свойство `Features.Item` (Коллекция `Features`)

Методы коллекции

- Метод `Features.Add` (Коллекция `Features`)
- Метод `Features.AddByID` (Коллекция `Features`)
- Метод `Features.Clone` (Коллекция `Features`)
- Метод `Features.Common` (Коллекция `Features`)
- Метод `Features.Remove` (Коллекция `Features`)
- Метод `Features.Replace` (Коллекция `Features`)
- Метод `Features.RemoveByID` (Коллекция `Features`)

Пример На Visual Basic

Пример на Visual Basic создает коллекцию объектов (Features), при выполнении поиска в определенной окружности с заданным радиусом, затем выбранные объекты (features) возвращаются.

```
Dim fs as Features

Dim pt As New MapXLib.Point

pt.Set -96.731, 39.728

set fs = map1.Layers(1).SearchWithinDistance(pt,_
5,miUnitMile,miSearchTypePartiallyWithin)

map1.Layers(1).Selection.Replace fs
```

Пример на C++

```
void CSampleProjectView::SelectObjects50Miles() {
    CMapXFeatures fs;
    CMapXPoint pt;
    if(!pt.CreateDispatch(pt.GetClsid())) {
        TRACE0("Failed to Create Point object");
        return;
    }
    try {
        pt.Set(-96.731,39.728);

        fs = m_Map.GetLayers().Item(1).SearchWithinDistance(pt,50,miUnitMile
,miSearchTypePartiallyWithin);
        m_Map.GetLayers().Item(1).GetSelection().Replace(fs);
    } catch (COleDispatchException *e) {
```



```

        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Смотрите также

Коллекция Selection

Свойство Feature.Area (Объект Feature)

Назначение

Свойство открытое только для чтения, которое возвращает площадь объекта типа полигон; для других типов объектов оно не действует.

Примечание

Для контроля того, в каких единицах возвращается площадь, установите свойство Map.AreaUnit.

Применимо только для полигонов. Измеряется в единицах площади. Если Вы изменяете объект на слое и еще не использовали оператор Update чтобы записать изменения, то это свойство возвратит значение площади относящееся к уже "измененному" объекту (объект в том виде как он есть в памяти).

Метод Feature.Attach (Объект Feature)

Назначение

Присоединяет отдельный объект к карте, так что к этому объекту применяется координатная система карты.

Синтаксис

ОБЪЕКТ.Attach (Map)

Часть Описание

object Объект Feature.

Map Объект Map.

Примечание

Когда Вы создаете отдельный объект, Вам надо присоединить такой объект к карте прежде чем Вы воспользуетесь каким-либо свойством или методом. Прикрепление объекта к карте объединяет координатную систему карты с объектом.

Таким образом, если Вы создаете отдельный объект, вызываете методу объекта FeatureFactory, Вам не надо вызывать метод Attach. (Методы FeatureFactory автоматически прикрепляют объекты к карте.).

Пример на Visual Basic

Следующий пример Visual Basic демонстрирует как присоединить к карте отдельный объект.

```
Dim f as new Feature
f.Attach Map1
f.Type = miFeatureTypeSymbol
f.Point.Set Map1.CenterX, Map1.CenterY
Map1.Layers(1).AddFeature f
```

Пример на C++

Следующий пример на C++ показывает как создать новый символ или текстовый объект. Метод Attach используется чтобы связать координатную систему с отдельным объектом.

```
// create a new text or symbol feature to our
// scratch layer in the ontolused event handler,
// depending on which tool was used

void CSampleDlg::OnToolUsedMap1(short ToolNum, double X1, double Y1,
double X2, double Y2, double Distance, BOOL Shift, BOOL Ctrl, BOOL FAR*
EnableDefault)
```

```

{
    if (ToolNum == MYTOOL_SYMBOL || ToolNum == MYTOOL_TEXT) {
        try {
            CMapXFeature f;

            // create a new symbol or text feature
            // first create the standalone feature object
            f.CreateDispatch(f.GetClsid());

            // attach standalone feature to map, as required in V3
            f.Attach(m_ctrlMapX.GetDispatch());
            if (ToolNum == MYTOOL_SYMBOL) {
                f.SetType(miFeatureTypeSymbol);
            }
            else {
                f.SetType(miFeatureTypeText);
                f.SetCaption("Text");
            }

            // set the style and location of the feature
            f.SetStyle(m_ctrlMapX.GetDefaultStyle());
            f.GetPoint().Set(X1,Y1);

            // now add it to our layer
            CMapXLayer layer = m_ctrlMapX.GetLayers().Item("scratch layer");
            layer.AddFeature (f);
        }
        catch (COleDispatchException *e) {

```

```

        e->ReportError();
        e->Delete();
    }
    catch (CObjectException *e) {
        e->ReportError();
        e->Delete();
    }
}
}

```

Метод **Feature.Clone** (Объект **Feature**)

Назначение

Создает копию данного объекта в виде нового, независимого объекта **Feature**.

Синтаксис

```
[=Feature]Feature.Clone
```

Свойство **Feature.Bounds** (Объект **Feature**)

Назначение

Возвращает объект **Rectangle** представляющий размер географического объекта (минимального описанного прямоугольника).

Пример на C++

```

// Feature.Bounds Property
void CSampleProjectView::ViewNY() {
    // Set the map view to view New York state
    try {

```

```

    CMapXFeature NYState =
Map.GetLayers().Item("USA").GetFind().Search("NY");
    // If FindRC returns an exact match
    if(NYState.GetFindRC()%10==1)
        m_Map.SetBounds(NYState.GetBounds());
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Пример на Visual Basic

```

Private Sub ViewNY()
    ' Set the Map view to view New York state
    Dim NYState As MapXLib.FindFeature
    Set NYState = Map1.Layers.Item("USA").Find.Search("NY")
    If NYState.FindRC Mod 10 = 1 Then
        Map1.Bounds = NYState.Bounds
    End If
End Sub

```

Примечание

В ранних версиях MapX, это свойство называлось MBR.

Если Вы изменяете объект на слое, и Вы еще не применили оператор Update для сохранения изменений, то все равно это свойство вернет параметры измененного объекта (объекта такого, какой он в памяти).

Свойство Feature.Caption (Объект Feature)

Назначение

Строковая величина, открытая только для чтения, представляющая содержание текстового объекта. Это свойство не применяется, если объект не текстовый Text.

Синтаксис

[= string] OBJECT.Caption

Часть	Описание
-------	----------

OBJECT	Объект Feature.
--------	-----------------

string	Строковая величина, размером до 255.
--------	--------------------------------------

Пример на Visual Basic

```
. Dim f As New Feature
Dim fNew As Feature
' Add a new text object to layer 1
f.Attach Map1
f.Type = miFeatureTypeText
f.Point.Set Map1.CenterX, Map1.CenterY
f.Caption = "This is a text object"
Set fNew = Map1.Layers(1).AddFeature(f)
' Change the text caption
fNew.Caption = "Changed Text"
fNew.Update
```

Пример на C++

```
void CSampleProjectView::TextFeatureCreate() {
    CMapXFeature f,fNew;
```

```

// Create new feature
if(!f.CreateDispatch(f.GetClsid())) {
    TRACE0("Failed to Create Feature object");
    return;
}

try {
    // Add the new text object to layer 1
    f.SetType(miFeatureTypeText);
    f.GetPoint().Set(m_Map.GetCenterX(),m_Map.GetCenterY());
    f.SetCaption("This is a text object");
    fNew = m_Map.GetLayers().Item(1).AddFeature(f);
    // Now change the text caption
    fNew.SetCaption("Changed Text");
    fNew.Update();
} catch (ColeDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (ColeException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Смотрите также

Объект Feature

Свойства Feature.CenterX, CenterY (Объект Feature)

Назначение

Величины двойной точности, открытые только для чтения, показывающие координаты X и Y центроида объекта Feature.

Примечание

Если Вам нужен быстрый доступ к центроиду объекта, то используйте свойство LabelPoint, вместо свойств CenterX и CenterY.

Свойство Feature.FeatureID (Объект Feature)

Назначение

Содержит индекс ID объекта. Каждый объект на слое содержит свой уникальный индекс ID. Это целая величина и доступная только для чтения. Это свойство, существующее по умолчанию для объекта Feature.

Внимание: Хотя это свойство еще работает в MapX, рекомендуем использовать Feature.FeatureKey.

Свойство Feature.FeatureKey (Объект Feature)

Назначение

Возвращает ключ (Key) объекта Feature. Каждый такой объект на слое содержит уникальный ключ внутри слоя. Это строковая величина, доступная только для чтения. Это замена старому свойству Feature.FeatureID (которое еще работает, но рекомендуется использовать свойство FeatureKey).

Внимание: Ключ объекта может использоваться как параметр так же как и FeatureID.

Смотрите также

Метод Selection.SelectByRegion

Dataset.Value

Dataset.Sourcerows

Метод Layer.Updatefeature

Метод Layer.Deletefeature

Метод Features.AddbyID

Метод Features.RemovebyID

Метод Selection.SelectByID

Свойство Feature.KeyValue (Объект Feature)

Назначение

Это строковое свойство (string property) использующееся для обновления или создания колонки в таблице MapInfo, когда добавляется или обновляется объект или слой.

Пример на C++

```
// Feature.KeyField Property
// Layer.KeyValue Property

void CSampleProjectView::ModifyByPoint(double X,double Y) {
    // This example assumes that you have a layer named "Customer Layer"
    // with a field named "CUST_NAME". If there is a feature underneath
    the point
    // (X,Y), the CUST_NAME field is changed to "MapInfo Corp"
    try {
        CMapXLayer srcLayer = m_Map.GetLayers.Item("Customer Layer");
        CMapXFeatures ftrsUnderPoint;
        CMapXPoint pt;
        if(!pt.CreateDispatch(pt.GetClsid()) {
            TRACE0("Failed to Create Point object");
            return;
        }
    }
```

```

    }

    pt.Set(X,Y);

    // Find all of the features underneath the point (X,Y) in "Customer
    Layer"

    ftrsUnderPoint = srcLayer.SearchAtPoint(pt);

    // Now, all feature.KeyValue changes will deal with the CUST_NAME
    field

    srcLayer.SetKeyField("CUST_NAME");

    for(long i=1;i<=ftrsUnderPoint.GetCount();i++) {
        AfxMessageBox("Old Value: " + ftrsUnderPoint.Item(i).GetName() +
"\nNew Value: MapInfo Corp");

        // change the value in the table to "MapInfo Corp"

        ftrsUnderPoint.Item(i).SetKeyValue("MapInfo Corp");

        ftrsUnderPoint.Item(i).Update();
    }
} catch (COleDispatchException *e) {
    e->ReportError();

    e->Delete();
} catch (COleException *e) {
    e->ReportError();

    e->Delete();
}
}

```

Пример на Visual Basic

```

Private Sub ModifyByPoint(X As Double,Y As Double)
    ' This example assumes that you have a layer named
    ' "Customer Layer"
    ' with a field named "CUST_NAME". If there is a feature
    ' underneath the point
    ' (X,Y), the CUST_NAME field is changed to "MapInfo Corp"
    Dim srcLayer as MapXLib.Layer
    Dim ftrsUnderPoint as MapXLib.Features
    Dim pt As New MapXLib.Point

    pt.Set X,Y
    Set srcLayer = Map1.Layers.Item("Customer Layer")

    ' Find all of the features underneath the point (X,Y) in
    ' "Customer Layer"
    Set ftrsUnderPoint = srcLayer.SearchAtPoint(pt)

    ' Now, all feature.KeyValue changes will deal with
    ' the CUST_NAME field
    srcLayer.KeyField = "CUST_NAME"

    For Each ftr in ftrsUnderPoint
        MsgBox "Old Value: " & ftr.Name & "    New Value: MapInfo _
        Corp"
        ' change the value in the table to "MapInfo Corp"
    
```

```

    ftr.KeyValue = "MapInfo Corp"

    ftr.Update

Next

End Sub

```

Примечание

Имя колонки, которое устанавливается или запрашивается определяется значением свойства KeyField объекта Layer, из которого взят изменяемый объект слоя. Хотя это свойство является строковой величиной, сам стринг будет преобразован в тот тип, из которого состоит колонка. Это произойдет при обновлении колонки или ее создании.

Только самое последнее значение свойства KeyValue используется, когда добавляются или изменяются объекты. Это происходит если Вы устанавливаете ключевое значение (keyvalue), изменяете свойство ключевой колонки на другую колонку и затем заново устанавливаете ключевое значение, то только второе значение будет обновлено для изменяемого объекта в таблице Mapinfo.

Смотрите также

Layer.KeyField

Объект Layer

Свойство Feature.LabelPoint (Объект Feature)

Назначение

Объект Point, который является центроидом в координатах карты (точка подписи) текстового объекта.

Смотрите также

Объект Point

Свойство Feature.Layer (Объект Feature)

Назначение

Свойство только для чтения - объект Layer, слой, из которого взят конкретный объект. Может иметь нулевое значение для отдельного (еще не прикрепленного к слою) объекта.

Смотрите также

Объект Layer

Свойство Feature.Length (Объект Feature)

Назначение

Свойство открытое только для чтения, которое возвращает длину линейного объекта в виде величины двойной точности. Не применяется для нелинейных объектов.

Пример на C++

```
// Feature.Distance Method

void CSampleProjectView::FindDistance() {

    // Use the Feature.Length method to find the total
    // distance of all the roads that the user selected

    CMapXSelection RoadsSelected;

    double TotalDistance = 0;

    long RoadCount;

    CString msg;

    try {

        m_Map.SetMapUnit(miUnitMile);
```

```

RoadsSelected = m_Map.GetLayers().Item("US Highways").
GetSelection();

RoadCount = RoadsSelected.GetCount();

for(long i=1;i<=RoadCount;i++) {
    // for each individual road, add its length to the total
    TotalDistance += RoadsSelected.Item(i).GetLength();
}
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}

msg.Format("The total distance was %f miles",TotalDistance);
AfxMessageBox(msg);
}

```

Пример на Visual Basic

```

Private Sub FindDistance_Click()

    ' Use the Feature.Length property to find the total distance
    ' of all the roads that the user selected

    Dim RoadsSelected As Selection

    Dim TotalDistance as Double

    Dim Road as Feature

    TotalDistance = 0

```

```

Map1.MapUnit = miUnitMile

Set RoadsSelected = Map1.Layers.Item("US Highways").Selection
For Each Road In RoadsSelected
    TotalDistance = TotalDistance + Road.Length
Next
MsgBox "The total distance was " & TotalDistance & " miles"
End Sub

```

Примечание

Для контроля за единицами измерения, в которых возвращается длина, установите свойство Map.MapUnit.

Если Вы изменяете объект на слое и еще не сделали процедуру Update для сохранения изменений, то это свойство вернет уже измененную длину объекта, так как объект существует в памяти.

Смотрите также

Feature.Perimeter

Свойство Feature.Name (Объект Feature)

Назначение

Свойство, доступное только для чтения, возвращает имя объекта Feature. Это имя берется из поля LabelProperties.DataField (значение этого поля также появляется как подпись объекта).

Смотрите также

Свойство LabelProperties.DataField

Свойство Feature.Nodes (Объект Feature)

Назначение

Это свойство открытое только для чтения, обращающееся к узлам объекта так что за один проход может опросить все узлы объекта и затем поместить их в один непрерывный блок памяти.

Синтаксис

[SafeArray=] OBJECT.Nodes ([CSys])

Часть Описание

OBJECT Это место для объекта Feature.

CSys Объект CoordSys определяющий систему координат, в которой будут возвращены узлы объекта. По умолчанию это Map.NumericCoordSys.

Пример

Следующий пример показывает выдачу сохраненного массива. Пусть Arr это массив возвращаемый после вызова Nodes обращенного к полилинии, состоящей из двух сегментов, где первый сегмент содержит 3 узла а второй сегмент содержит 2 узла.

```
//Arr is a 7 x 2 array

Arr(1,1) = # of nodes in the segment 1 (in this case 3)
Arr(2,1) = X Coord of segment 1, node 1
Arr(3,1) = Y Coord of segment 1, node 1
Arr(4,1) = X Coord of segment 1, node 2
Arr(5,1) = Y Coord of segment 1, node 2
Arr(6,1) = X Coord of segment 1, node 3
Arr(7,1) = Y Coord of segment 1, node 3

Arr(1,2) = # of nodes in the segment 2 (in this case 2)
Arr(2,2) = X Coord of segment 2, node 1
Arr(3,2) = Y Coord of segment 2, node 1
```



```
Arr(4,2) = X Coord of segment 2, node 2
```

```
Arr(5,2) = Y Coord of segment 2, node 2
```

```
Arr(6,2) = Undefined
```

```
Arr(7,2) = Undefined
```

Пример на Visual Basic

```
// The following VB Code will print all the nodes for the first
//      object in a selection on the USA layer to the debug window.
Private Sub Command2_Click()

    Dim arrVals As Variant

    Dim obj As MapXLib.Feature

    Dim iNodeCnt As Integer

    Dim NodesLBound As Integer

    Dim NodesUBound As Integer

    Set obj = Map1.Layers("USA").Selection(1)

    arrVals = obj.Nodes

    NodesLBound = LBound(arrVals, 1)

    NodesUBound = UBound(arrVals, 1)

    PolysLBound = LBound(arrVals, 2)

    PolysUBound = UBound(arrVals, 2)

    For i = PolysLBound To PolysUBound

        Debug.Print "Begin Poly #"; i

        iCoordCnt = arrVals(NodesLBound, i) * 2

        For j = NodesLBound + 1 To iCoordCnt Step 2
```

```

        Debug.Print "Pt#"; j / 2; "(";
        arrVals(j, i); ","; arrVals(j + 1, i); ")"

    Next j

    Debug.Print "End Poly #"; i

    Debug.Print ""

Next i

End Sub

```

Метод **Feature.Offset** (Объект **Feature**)

Назначение

Перемещает объект относительно его текущей позиции.

Синтаксис

ОБЪЕКТ.Offset(deltaX, deltaY)

Части Описание

ОБЪЕКТ Это место для объекта Feature.

deltaX Расстояние в координатах карты, на которое перемещается указатель по оси x.

deltaY Расстояние в координатах карты, на которое перемещается указатель по оси y.

Свойство **Feature.Parts** (Объект **Feature**)

Назначение

Свойство только для чтения, возвращает объект Parts. Действует для линий и многоугольников.

Смотрите также

Объект Parts

Свойство **Feature.Perimeter** (Объект **Feature**)

Назначение

Свойство открытое только для чтения, которое возвращает периметр объекта полигон; не действует для других типов объектов.

Примечание

Для контроля за единицами измерения, в которых возвращается длина, установите свойство `Map.MapUnit`.

Если Вы изменяете объект на слое и еще не сделали процедуру `Update` для сохранения изменений, то это свойство вернет уже измененную длину объекта, так как объект существует в памяти.

Смотрите также

`Feature.Length`

Свойство **Feature.Point** (Объект **Feature**)

Назначение

Объект `Point`, который является местом в координатах карты, расположение объекта типа символ или текст.

Смотрите также

Объект `Point`

Метод **Feature.Smooth** (Объект **Feature**)

Назначение

Метод `Feature.Smooth` позволяет Вам сгладить полилинию, превратив ее в непрерывную кривую. Для сглаживания полилинии установите значение свойства на `True`.

Свойство **Feature.Style** (Объект **Feature**)

Назначение

Свойство для чтения/записи. Стил ь объекта.

Смотрите также

Объект **Style**

Примечание

Объект **Style** по умолчанию имеет значение '0' для каждого свойства объекта стиля.

Свойство **Feature.Type** (Объект **Feature**)

Назначение

Содержит тип объекта. Это значение **FeatureTypeConstants**, и оно открыто для чтения/записи.

Пример

```
Dim f As New Feature
Dim fNew As Feature

'Add a new text object to layer 1
f.Attach Map1
f.Type = miFeatureTypeText
f.Point.Set Map1.CenterX, Map1.CenterY
f.Caption = Map1.Layers(1).AddFeature(f)

'Change the text caption
fNew.Caption = "Changed Text"
fNew.Update
```

Смотрите также

FeatureFactory.CreateText

Метод Feature.Update (Объект Feature)**Назначение**

Обновляет слой с теми изменениями, которые сделаны над объектом Feature.

Синтаксис

```
[boolean=] OBJECT.Update ( [UpdateFeature] , [RowValues] )
```

Части**Описание**

OBJECT Это место для объекта Feature.

UpdateFeature Логическая величина, показывающая надо ли обновлять
 колонку объекта. По умолчанию равна true.

RowValues Представляет новые значения данных.

Внимание: Колонка объектов хранит географические данные для объекта Feature.

Пример на C++

```
// Feature.Offset Method
// Feature.Update Method
// define DRAG_TOOL_ID 40
void CSampleProjectView::OnCreate() {
    // ...
    m_Map.CreateCustomTool(DRAG_TOOL_ID,miToolTypeLine,miSizeAllCur sor);
    // ...
}
```

```

void CSampleProjectView::OnToolUsed(short ToolNum, double X1, double Y1,
double X2, double Y2, double Distance, BOOL Shift, BOOL Ctrl, BOOL*
EnableDefault)
{
    if(ToolNum==DRAG_TOOL_ID) {

        // Offset the feature underneath the mouse by however much the user
dragged

        // This example gets the features from a layer named "My Layer"

        // All of the changes are permanent and cannot be undone; do not use
this on

        // any permanent layers unless you wish to permanently alter the map
layer

        CMapXPoint pt;

        CMapXFeatures FtrsUnderPt;

        if(!pt.CreateDispatch(pt.GetClsid())) {

            TRACE0("Couldn't Create Point object");

            return;

        }

        try {

            pt.Set(X1,Y1);

            FtrsUnderPt = m_Map.GetLayers().Item("My
Layer").SearchAtPoint(pt);

            // Take the first item in the collection,even if there are more
than one

            // We do this because if there are two items on top of each other,
they

```

```

        // would be inseparable if all the items returned by SearchAtPoint
were used

        // This way, only one of them is moved, and the user will be able
to move

        // one off of the other

        if(FtrsUnderPt.GetCount()>=1) {

            CMapXFeature ftr = FtrsUnderPt.Item(1);

            ftr.Offset(X2-X1,Y2-Y1);

            ftr.Update();

        }

    } catch (COleDispatchException *e) {

        e->ReportError();

        e->Delete();

    } catch (COleException *e) {

        e->ReportError();

        e->Delete();

    }

}

}

```

Пример на Visual Basic

```

Const DRAG_TOOL_ID = 40

Private Sub Form_Load()

    ' ...

    Map1.CreateCustomTool DRAG_TOOL_ID,miToolTypeLine,miSizeAllCursor

    ' ...

End Sub

```

```

Private Sub Map1_ToolUsed(ByVal ToolNum As Integer, ByVal X1 _
As Double, ByVal Y1 As Double, ByVal X2 As Double, ByVal Y2 _
As Double, ByVal Distance As Double, ByVal Shift As Boolean, _
ByVal Ctrl As Boolean, EnableDefault As Boolean)

    If ToolNum = DRAG_TOOL_ID Then

        ' Offset the feature underneath the mouse by however much
        ' the user dragged
        ' This example gets the features from a layer named "My
        ' Layer"
        ' All of the changes are permanent and cannot be undone;
        ' do not use this on
        ' any permanent layers unless you wish to permanently
        ' alter the map layer

        Dim pt As New MapXLib.Point
        Dim FtrsUnderPt As MapXLib.Features
        Dim ftr As MapXLib.Feature

        pt.Set X1, Y1

        Set FtrsUnderPt = Map1.Layers.Item("My Layer").SearchAtPoint(pt)

        ' Take the first item in the collection, even if there
        ' are more than one
        ' We do this because if there are two items on top of
        ' each other, they
        ' would be inseparable if all the items returned by
        ' SearchAtPoint were used
        ' This way, only one of them is moved, and the user
        ' will be able to move

```



```

' one off of the other

If FtrsUnderPt.Count = 1 Then
    Set ftr = FtrsUnderPt.Item(1)
    ftr.Offset X2 - X1, Y2 - Y1
    ftr.Update
End If
End If
End Sub

```

Метод **Features.Add** (Коллекция **Features**)

Назначение

Добавляет объект **Feature** или все объекты из объекта **Features** в коллекцию. (UNION set operation).

Синтаксис

ОБЪЕКТ.Add (Source)

Часть Описание

ОБЪЕКТ Представляет объект **Features**.

Source Объект **Feature** или коллекция **Features** для добавления к коллекции **Features** в ОБЪЕКТ.

Пример на **Visual Basic**

```

' Select every state which contains a Top 20 city.

Dim ftr As Feature

Dim ftrs As Features

Dim usaLayer As Layer

Set usaLayer = Map1.Layers.Item("USA")

```

```

Set ftrs = usaLayer.NoFeatures

For Each ftr In Map1.Layers.Item("US Top 20 Cities").AllFeatures
    ftrs.Add usaLayer.SearchAtPoint(ftr.Point)
Next

usaLayer.Selection.Replace ftrspn

```

Смотрите также

Объект Feature и коллекция Features

Метод Features.AddByID (Коллекция Features)

Назначение

Добавляет объект Feature с указанным FeatureKey к коллекции Features.

Синтаксис

ОБЪЕКТ.AddByID (FeatureKey)

Часть	Описание
-------	----------

ОБЪЕКТ	Это место для объекта Features.
--------	---------------------------------

FeatureKey	FeatureKey объекта, добавляемого к коллекции Features.
------------	--

Внимание: Feature.FeatureKey заменяет собой Feature.FeatureID. FeatureID пока еще работает, но рекомендуется пользоваться новым свойством FeatureKey.

Смотрите также

Свойство `Feature.FeatureKey`

Свойство `Features.Bounds` (Коллекция `Features`)

Назначение

Возвращает объект `Rectangle` представляющий географическое пространство всех объектов в коллекции (минимальный описанный прямоугольник).

Синтаксис

`OBJECT.Bounds`

Вместо `object` разместите коллекцию `Features`.

Пример на C++

```
// Features.Bounds Property
void CSampleProjectView::ViewSelection(CMapXLayer lyr) {
    // set the map to view the current selection
    CMapXFeatures selectedFeatures;
    try {
        selectedFeatures = lyr.GetSelection().Clone();
        m_Map.SetBounds(selectedFeatures.GetBounds());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

```
}
```

Пример на Visual Basic

```
Private Sub ViewSelection(lyr As MapXLib.Layer)
    ' set the map to view the current selection
    Dim selectedFeatures As Features
    Set selectedFeatures = lyr.Selection.Clone
    Map1.Bounds = selectedFeatures.Bounds
End Sub
```

Примечание

Это свойство полезно, если Вы хотите изменить масштаб карты так, чтобы показать все объекты в коллекции.

Метод **Features.Clone** (Коллекция **Features**)

Назначение

Создает копию коллекции в виде другого объекта **Features**.

Синтаксис

ОБЪЕКТ.Clone (source)

Части Описание

ОБЪЕКТ Это место для коллекции **Features**.

source Это набор объектов **Features** которые будут копироваться с коллекцией, ОБЪЕКТ.

Метод **Features.Common** (Коллекция **Features**)

Назначение

Объединяет эту коллекцию и другой объект **Features** так, что эта коллекция содержит только те объекты, которые есть и там и там. (INTERSECT set operation).

Синтаксис

ОБЪЕКТ.Common (source)

Части Описание

ОБЪЕКТ Это место для коллекции **Features**.

source Это набор объектов **Features** которые будут объединены с коллекцией, ОБЪЕКТ.

Пример на **Visual Basic**

```
' Select every state that is within 500 miles of New Jersey AND
' within 500 miles of the point that was clicked.

Private Sub Map1_ToolUsed(ByVal ToolNum As Integer, ByVal X1 As Double,
ByVal Y1 As Double, ByVal X2 As Double, ByVal Y2 As Double, ByVal
Distance As Double, ByVal Shift As Boolean, ByVal Ctrl As Boolean,
EnableDefault As Boolean)

    If ToolNum = MY_TOOL Then

        Dim ftrs As Features

        Dim newJersey As Feature

        Dim usaLayer As Layer

        Dim pt As New Point

        pt.Set X1, Y1

        Set usaLayer = Map1.Layers.Item("USA")

        Set newJersey = usaLayer.Find.Search("NJ")
```

```
Set ftrs = usaLayer.SearchWithinDistance(pt, 500, miUnitMile,
miSearchTypePartiallyWithin)
```

```
ftrs.Common usaLayer.SearchWithinDistance(newJersey, 500,
miUnitMile, miSearchTypePartiallyWithin)
```

```
usaLayer.Selection.Replace ftrs
End If
End Sub
```

Свойство Features.Count (Коллекция Features)

Назначение

Количество объектов в коллекции.

Свойство Features.Item (Коллекция Features)

Назначение

Возвращает объект Feature из коллекции – пока поддерживается только числовая индексация. Это свойство по умолчанию присуще коллекции Features.

Метод Features.Remove (Коллекция Features)

Назначение

Удаляет объект Feature или все объекты из объекта Features из данной коллекции. (SUBTRACT set operation).

Синтаксис

OBJECT.Remove (source)

Части Описание

ОБЪЕКТ Это место для объекта Features.

source Это совокупность объектов Features которые будут удалены из коллекции, ОБЪЕКТ.

Пример на Visual Basic

```
' Select every state which _does not_ contain a Top 20 city.

Dim usaLayer As Layer

Dim ftrs As Features

Dim ftr As Feature

Set usaLayer = Map1.Layers.Item("USA")

Set ftrs = usaLayer.AllFeatures

For Each ftr In Map1.Layers.Item("US Top 20 Cities").AllFeatures
    ftrs.Remove usaLayer.SearchAtPoint(ftr.Point)
Next

usaLayer.Selection.Replace ftrs;
```

Метод Features.RemoveByID (Коллекция Features)

Назначение

Удаляет объект Feature с указанным индексом FeatureID из коллекции Features.

Синтаксис

ОБЪЕКТ.RemoveByID(*FeatureKey)

Часть Описание

ОБЪЕКТ Это место для объекта Features.

FeatureKey *FeatureKey объекта, который будет удален из коллекции Features. Этот ключ заменяет собой старый параметр FeatureID. FeatureID до сих пор

работает, но рекомендуется пользоваться новым параметром `FeatureKey`.

Метод `Features.Replace` (Коллекция `Features`)

Назначение

Перемещает содержимое коллекции с объектом `Feature` или всеми объектами из объекта `Features`.

Синтаксис

ОБЪЕКТ.`Replace` (source)

Части Описание

ОБЪЕКТ Это место для объекта `Features`.

source Это объект `Feature` или коллекция `Features`, содержание которых будет замещено содержанием "feature".

Объект `FeatureFactory`

Этот объект позволяет Вам создавать новые объекты карты, или создавать объекты при применении операций к существующим объектам (таким как буферизация).

Методы Объекта

- Метод `FeatureFactory.BufferFeatures` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CombineFeatures` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateArc` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateCircularRegion` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateEllipticalRegion` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateLine` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateRegion` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateSymbol` (Объект `FeatureFactory`)
- Метод `FeatureFactory.CreateText` (Объект `FeatureFactory`)

- Метод FeatureFactory.EraseFeature (Объект FeatureFactory)
- Метод FeatureFactory.IntersectFeatures (Объект FeatureFactory)
- Метод FeatureFactory.IntersectionPoints (Объект FeatureFactory)
- Метод FeatureFactory.IntersectionTest (Объект FeatureFactory)

Примечание

Большинство из этих методов возвращают отдельные, неприкрепленные к слою объекты. Эти объекты карты автоматически присоединяются к карте (то есть они получают ту же систему координат, что и карта). Другими словами Вам не надо применять метод Attach к тем объектам, которые возвращаются этими методами.

Что бы получить объект FeatureFactory, сошлитесь на свойство Map.FeatureFactory.

Пример на Visual Basic

```
Dim f As MapXLib.Feature
Dim p As New MapXLib.Point

' Use the map's current center as the point coordinates
' p.Set Map1.CenterX, Map1.CenterY
' Create a text feature and add it to layer 1
Set f = Map1.Layers(1).AddFeature(Map1.FeatureFactory. _
CreateText(p, "Some Text"))
```

Пример на C++

```
void CSampleProjectView::FeatureFactoryTextCreate() {
    CMapXFeature ftr;
    CMapXPoint pnt;
    COleVariant pntVt;

    if(!pnt.CreateDispatch(pnt.GetClsid())) {
        TRACE0("Failed to Create Point object");
        return;
    }
}
```

```

    }

    try {

        // Use the map's current center as the point's coordinates
        pnt.Set(m_Map.GetCenterX(),m_Map.GetCenterY());

        pntVt.vt = VT_DISPATCH;
        pntVt.pdispVal = pnt.m_lpDispatch;

        // Create a text feature and add it to layer 1
        ftr = m_Map.GetLayers().Item(1).
        AddFeature(m_Map.GetFeatureFactory().
        CreateText(pntVt,"Some Text"));
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Метод FeatureFactory.BufferFeatures (Объект FeatureFactory)

Назначение

Возвращает отдельный (неприкрепленный) объект Feature представляющий буферную зону.

Синтаксис

OBJECT.BufferFeatures (Source, Distance, [Units] , [Resolution])

Часть	Описание
ОБЪЕКТ	Объект <code>FeatureFactory</code> .
Source	Объект <code>Feature</code> или объект коллекции <code>Features</code> . В буфер могут попадать точки, линии или области; текстовые объекты в буфер не включаются.
Distance	Размер буфера, в единицах, указанных в аргументе <code>units</code> . Если параметр <code>distance</code> положителен, полигон буфера больше чем объект(ы) источника <code>source</code> . Когда буферизация осуществляется над полигонами, Вы можете указать отрицательную дистанцию <code>distance</code> , которая создаст буферный полигон, который меньше чем полигон источника.
Units	Переменная: Значение <code>MapUnitConstants</code> , такое как <code>miUnitMile (0)</code> , которое определяет единицы измерения для аргумента <code>distance</code> . Если значение пропущено, то свойство <code>MapUnit</code> используется по умолчанию
Resolution	Переменная: Положительное целое определяющее номер узлов, используемых при создании буфера. Это количество узлов для окружности в операции создания буфера, а не общее количество узлов для целого региона; смотрите описание ниже. Если это значение пропущено, то разрешение (<code>resolution</code>) определяется свойство <code>DefaultConversionResolution</code> .

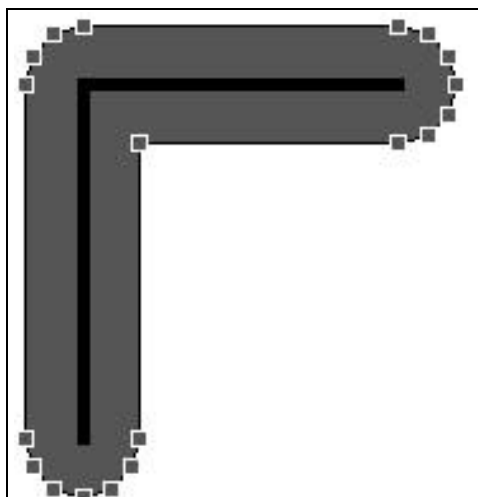
Примечание

Этот метод возвращает один полигон буферной зоны, независимо от того, основана ли буферизация на одном типе объектов `Feature` или на коллекции `Features`. Если Вы хотите создать отдельный буфер для каждого объекта из коллекции, Вам надо создать цикл для всей коллекции и использовать метод `BufferFeatures` для каждого объекта.

Аргумент `resolution` определяет, то как много узлов будет в полигоне буфера. Разрешение задается в количестве узлов для окружности; это хорошо иллюстрируется следующим примером. Картинка слева показывает два объекта: Черную, L-образную линию, и серый полигон буферной зоны. Полигон был создан при образовании буфера вокруг полилинии.

В этой иллюстрации, маркеры в виде черных квадратиков отмечают каждый узел в полигоне буфера. Этот буфер был создан при разрешении 12 узлов на окружность. Обратите внимание, что конец каждой полилинии окружен полукругом, и каждый полукруг состоит из 6 сегментов.

Если увеличить параметр `resolution`, то соответственно увеличится количество узлов в каждом таком полукруге.



Обратите внимание, что `resolution` не представляет общее количество узлов в полигоне буфера. Например, полигон буферной зоны, показанный на рисунке, имеет более 12 узлов, хотя разрешение определено как 12 узлов на окружность. При задании большего разрешения (`resolution`) буферная зона будет выглядеть более гладкой, но это требует большего времени для расчетов и места в памяти для хранения объекта полигон буфера.

Смотрите также

Метод `FeatureFactory.CreateCircularRegion`

Метод `FeatureFactory.CombineFeatures` (Объект `FeatureFactory`)

Назначение

Возвращает отдельный, непривязанный к слою, объект `Feature` представляющий комбинацию (объединение) нескольких линейных объектов или нескольких объектов типа полигон.

Синтаксис

`ОБЪЕКТ.CombineFeatures(feature1 [, feature2])`

Часть **Описание**

`object` Объект `FeatureFactory`

feature1	Объект Feature или коллекция объектов Features.
feature2	Переменная: Объект Feature или коллекция объектов Features.

Примечание

Если feature1 это коллекция Features, то аргумент feature2 является необязательным. Если feature1 это объект Feature, то аргумент feature2 является необходимым. Вы не можете комбинировать линейный объект в объект типа полигон. Поддерживаются следующие виды преобразований:

- Вы можете объединять два или более линейных объекта (объекты, чье свойство Type это miFeatureTypeLine), в этом случае результат тоже будет линейным объектом.
- Вы можете объединять два или более объекта типа полигон (тип: miFeatureTypeRegion), в этом случае результат будет тоже объект типа полигон.

Метод FeatureFactory.CreateArc (Объект FeatureFactory)

Назначение

Создает линейные объекты формы дуги, и возвращает дугу в виде одиночного(неприкрепленного к слою) объекта Feature.

Синтаксис

ОБЪЕКТ.CreateArc(Point1, Point2, [Angle] , [Distance] , [Resolution] , [Style])

Часть Описание

ОБЪЕКТ	Объект FeatureFactory.
Point, Point2	Два объекта типа точка, представляющие начало и конец дуги.
Angle	Переменная: Угол, в градусах. Не обязательный параметр; если он пропущен, по умолчанию будет угол 90 градусов, создастся симметричная дуга.
Distance	Переменная: Длина линии уступа (offset line), выраженная в как количество "единиц дуги" (где одна единица дуги это расстояние между точками Point1 и Point2). Увеличение этого значения приводит к появлению более изогнутой дуги; при задании значения 0 появится прямая линия. Необязательный параметр; если он пропущен, по умолчанию будет значение 1.

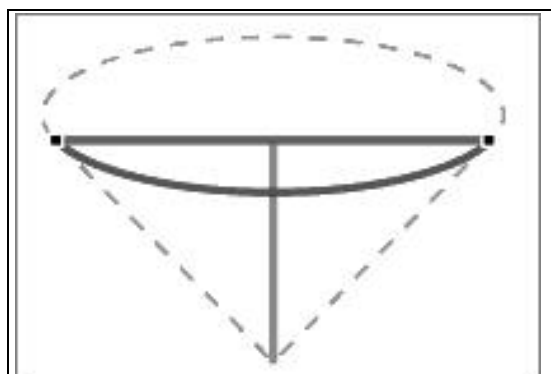
Ressolution	Переменная: Положительное целое указывающее на количество узлов, используемых при создании эллипса; максимальное значение это 32,763. Необязательный параметр; если он пропущен, то разрешение будет контролироваться свойством DefaultConversionResolution.
Style	Переменная: Объект Style который определяет вид объекта. Необязательный параметр; если он пропущен, никакой стиль не будет установлен.

Примечание

Этот метод возвращает линейный объект (чье свойство Type это miFeatureTypeLine) по форме он выглядит как дуга. Обратите внимание: Если текущее значение DisplayCoordSys не совпадает с текущим значением NumericCoordSys, дуга может появиться с искаженной формой.






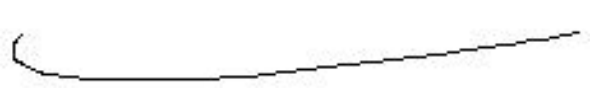
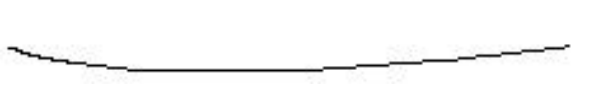
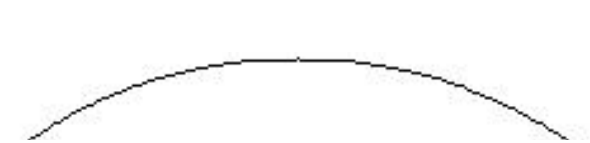
Чтобы рассчитать, как будет выглядеть дуга, представим линию, соединяющую Point1 и Point2 (горизонтальная линия на диаграмме слева). Далее, представим вторую линию (вертикальная линия на диаграмме) которая начинается из центральной точки первой линии. Длина второй линии определяется аргументом Offset, и ее угол относительно первой линии определяется аргументом Angle. Чтобы эта линия была перпендикулярна первой, задайте угол 90 или 270 градусов.

Образует треугольник, определяющийся точками Point1, Point2 и конечной точкой второй линии. Этот треугольник действует как "контрольный полигон", в том смысле, что он определяет форму дуги. Дуга заполняет такой треугольник, касаясь двух его боковых сторон:



Дуга располагается сверху над линией между точками Point1 и Point2. Чтобы расположить дугу ниже этой линии, аргумент Angle определите как 270 градусов).

Каждая комбинация значений Angle и Offset порождает единственную по форме дугу. Если Вы всегда используете одну и ту же комбинацию значений Angle и Offset, то получающиеся дуги будут иметь всегда одну форму, независимо от расстояния между Point1 и Point2. Некоторые примеры приведены ниже

Угол	Offset	Результат
30	1	
30	0.5	
90	1	
90	0.5	
90	0.25	
165	1	
165	0.5	
270	0.5	

Смотрите также

Метод `FeatureFactory.CreateEllipticalRegion`

Метод **FeatureFactory.CreateCircularRegion** (Объект **FeatureFactory**)

Назначение

Создает объект типа полигон в форме круга, и возвращает его в виде отдельного, неприкрепленного к слою, объекта **Feature**. Такой объект может быть добавлен к слою, использоваться в процедуре поиска и т.д.

Синтаксис

ОБЪЕКТ.CreateCircularRegion(**Type**, **Point**, **Distance**, [**Units**] , [**Resolution**] , [**Style**])

Часть	Описание
-------	----------

ОБЪЕКТ	Объект FeatureFactory .
---------------	--------------------------------

Type	Краткое целое: Величина CircleTypeConstants : Может быть miCircleTypeScreen (значение: 0) или miCircleTypeMap (1). Смотрите ниже.
-------------	--

Point	Объект Point : Представляет собой место - центр круга, указывается в текущей системе координат.
--------------	--

Distance	Величина двойной точности: Радиус круга, в единицах определенных аргументом Units .
-----------------	--

Units	Переменная: Значение MapUnitConstant , такое как miUnitMile (0), которое идентифицирует единицу измерения для аргумента Distance . Не обязательный параметр; если он пропущен, то свойство MapUnit будет контролировать единицы измерения.
--------------	--

Resolution	Переменная: Целое положительное, определяющее количество узлов, используемых для создания окружности; максимальное число 32763. Не обязательный параметр; если он пропущен, по свойство DefaultConversionResolution будет контролировать разрешение. Чем больше разрешение, там более сглаженным будет полигон, но при этои растет время обработки и место на диске для хранения создаваемого объекта.
-------------------	---

Style	Переменная: Объект Style , который определяет стилевое оформление объекта. Если он отсутствует, стилового оформления не будет.
--------------	---

Примечание

Если **Type** имеет значение **miCircleTypeScreen**, то **MapX** создает полигон, появляющийся еа экране в виде окружности. Таким образом, различные точки на окружности могут не иметь одинаковое географическое расстояние от центра окружности. **MapX** вычисляет точки вдоль

оси x в единицах Distance от центра. Это расстояние преобразуется в экранные единицы расстояния (пиксели) и используется как радиус для расчета остальных узлов окружности. Обратите внимание: Если Вы меняете координатную систему карты после создания полигона, полигон может стать искаженным, отличающимся от окружности.

Если Type это miCircleTypeMap, то MapX создает полигон, точки которого, имеют все одинаковое географическое расстояние от центра. Таким образом, в зависимости от кривизны земной поверхности, системы координат и проекции карты, полигон может принять вид отличный от окружности.

Смотрите также

Метод BufferFeatures

Метод CreateEllipticalRegion

Метод FeatureFactory.CreateEllipticalRegion (Объект FeatureFactory)

Назначение

Создает объект типа полигон, в форме эллипса и возвращает его в виде одиночного объекта Feature, не привязанного к слою. Объект может быть добавлен к слою, может использоваться в процедуре поиска и т.д.

Синтаксис

OBJECT.CreateEllipticalRegion(Rectangle, [Angle] , [Resolution] , [Style])

Часть	Описание
OBJECT	Объект FeatureFactory.
Rectangle	Объект Rectangle, представляющий минимальный описанный вокруг объекта прямоугольник (MBR), внутри которого будет размещен эллипс в текущей системе координат.
Angle	Переменная: Угол вращения для эллипса. Эллипс вращается относительно своего центра.
Resolution	Переменная: Положительное целое, определяющее количество узлов, используемых при создании эллипса, наибольшее значение 32763. Если это значение отсутствует, то свойство DefaultConversionResolution будет контролировать разрешение
Style	Переменная: Объект Style, который определяет стилевое

оформление объекта. Не обязательный параметр; если он отсутствует, стилевое оформление отсутствует.

Пример на C++

```
// FeatureFactory.CreateEllipticalRegion Method
void CSampleProjectView::CreateEllipse() {
    // Create a new ellipse region and add it to a temporary features
    layer
    CMapXRectangle rect;
    CMapXFeature createdEllipse;
    if(!rect.CreateDispatch(rect.GetClsid())) {
        TRACE0("Failed to Create rectangle object");
        return;
    }
    try {
        rect.Set(m_Map.GetCenterX(),m_Map.GetCenterY(),m_Map.GetCenterX
        (+30,m_Map.GetCenterY()+10);
        createdEllipse =
        m_Map.GetFeatureFactory().CreateEllipticalRegion(rect);
        m_Map.GetLayers().Item("Temp Layer").AddFeature(createdEllipse);
    } catch(COleDispatchException* e) {
        e->ReportError();
        e->Delete();
    } catch(COleException* e) {
        e->ReportError();
        e->Delete();
    }
}
```

Пример на Visual Basic

```
Private Sub CreateEllipse()
    ' Create a new ellipse region and add it to a temporary
    ' features layer
    Dim rect As New MapXLib.Rectangle
    Dim createdEllipse As Feature

    rect.Set Map1.CenterX, Map1.CenterY, Map1.CenterX
    + 30, Map1.CenterY + 10
    Set createdEllipse = Map1.FeatureFactory.CreateElliptical _
Region(rect, 0, 24, Map1.DefaultStyle)
    Map1.Layers.Item("Temp Layer").AddFeature createdEllipse
End Sub
```

Примечание

Если текущее значение DisplayCoordSys не совпадает с текущим значением NumericCoordSys, то эллипс может появиться в искаженной форме.

Смотрите также

Метод CreateArc

Метод CreateCircularRegion

Метод FeatureFactory.CreateLine (Объект FeatureFactory)

Назначение

Возвращает отдельный объект Feature (линейный объект), построенный из коллекции точек.

Синтаксис

ОБЪЕКТ.CreateLine([Points] , [Style])

Часть**Описание**

ОБЪЕКТ

Объект FeatureFactory.

Points

Переменная: Коллекция Points, представляющая набор точек, используемых для построения линейного объекта.

Необязательный параметр; если он отсутствует, точек к линейному объекту добавлено не будет.

Style

Переменная: Объект Style который определяет стилевое оформление объекта. Не обязательный параметр; если он отсутствует, стилевое оформление отсутствует.

Метод FeatureFactory.CreateRegion (Объект FeatureFactory)**Назначение**

Возвращает отдельный объект Feature (объект полигон), построенный из коллекции точек.

Синтаксис

ОБЪЕКТ.CreateRegion([Points] , [Style])

Часть**Описание**

ОБЪЕКТ

Объект FeatureFactory.

Points

Переменная: Коллекция Points, представляющая набор точек, которые будут определять полигон. Если объект Rectangle создает полигон, то используются четыре точки для углов прямоугольника. Необязательный параметр; если он отсутствует, точек к объекту добавлено не будет.

Style

Переменная: Объект Style который определяет стилевое оформление объекта. Не обязательный параметр; если он отсутствует, стилевое оформление отсутствует.

Метод FeatureFactory.CreateSymbol (Объект FeatureFactory)

Назначение

Возвращает отдельный объект Feature (точечный объект), построенный из коллекции точек.

Синтаксис

ОБЪЕКТ.CreateSymbol([Point] , [Style])

Часть	Описание
ОБЪЕКТ	Объект FeatureFactory.
Point	Переменная: объект Point, представляющий позицию x/y того места, где будет помещен символ. Не обязательный параметр; если он отсутствует, то местоположение объекта не установлено.
Style	Переменная: Объект Style который определяет стилевое оформление объекта. Не обязательный параметр; если он отсутствует, стилевое оформление отсутствует.

Метод FeatureFactory.CreateText (Объект FeatureFactory)

Назначение

Возвращает отдельный, неприкрепленный к слою объект Feature (текстовый объект).

Синтаксис

ОБЪЕКТ.CreateText([Point] , [Caption] , [Position] , [Style])

Часть	Описание
ОБЪЕКТ	Объект FeatureFactory.
Point	Переменная: объект Point, представляющий место, куда будет прикреплен текст. Не обязательный параметр; если он отсутствует, то местоположение объекта не установлено.
Caption	Переменная: Строковая величина, определяющая текст. Не обязательный параметр; если он отсутствует, подпись объекта не будет установлена.
Position	Величина PositionConstants; определяет начальную позицию

текста относительно места его прикрепления. Не обязательный параметр; если он отсутствует, по умолчанию `miPositionTL` (то есть место прикрепления текста будет левым верхним углом текста).

Этот аргумент действует только, если определены оба аргумента `Point` и `Caption`.

Style

Переменная: Объект `Style` который определяет стилевое оформление объекта. Не обязательный параметр; если он отсутствует, стилевое оформление отсутствует.

Примечание

Вы определяете местоположение текстового объекта, указывая `Point` и `Position`. Обратите внимание, что свойство `Position` может использоваться только для указания начала позиции текста; объект `Feature` не снабжен свойством `Position`, что позволяет Вам изменять ориентацию текста. Этот метод создает текстовый объект, и нет возможности в нем менять ориентацию текста относительно точки его прикрепления.

Метод `FeatureFactory.EraseFeature` (Объект `FeatureFactory`)

Назначение

Возвращает отдельный, не привязанный к слою объект `Feature` при "стирании" одной области объекта из другого объекта.

Синтаксис

`OBJECT.EraseFeature(SourceFeature, EraserFeature)`

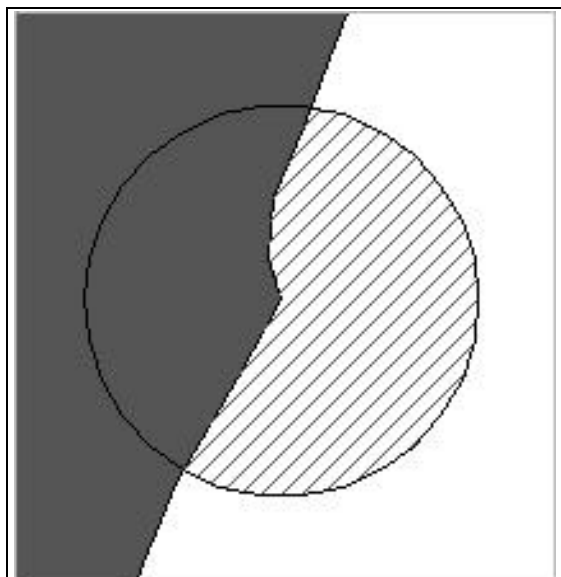
Часть	Описание
OBJECT	Объект <code>FeatureFactory</code> .
SourceFeature	Объект <code>Feature</code> , часть которого Вы хотите стереть; это может быть линейный объект или полигон.
EraserFeature	Объект <code>Feature</code> , который представляет область, которую Вы хотите удалить из <code>SourceFeature</code> . Это должен быть объект типа полигон.

Примечание

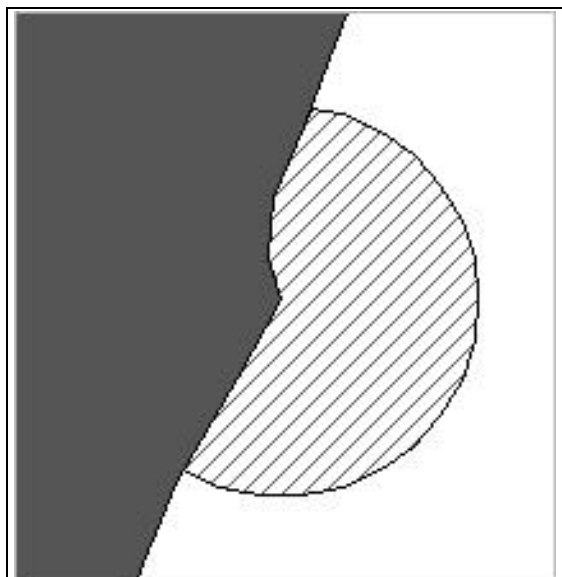
Этот метод создает объект Feature при вычитании EraserFeature из SourceFeature. Если два объекта непересекаются, то ничего не сотрется, объект возвращаемый этим методом будет идентичным SourceFeature.

Пример

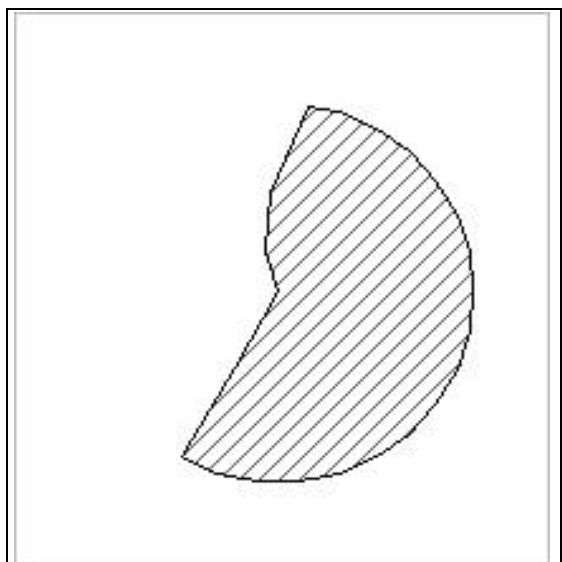
Этот рисунок показывает полигон в виде круга, созданный методом BufferFeatures, В этом примере, полигон в виде окружности покрывает береговую линию; область серого цвета представляет береговую часть, а белая область представляет воду.



Чтобы стереть часть полигона типа окружность, используйте метод EraseFeature. Если Вы делаете полигон типа окружность вашего SourceFeature, и делаете серый регион вашего EraserFeature, Вы сотрете часть окружности.



Этот рисунок показывает то, что останется от полигона типа окружность после вызова метода `EraseFeature`.



Метод `EraseFeature` удаляет часть объекта, которая перекрывает другой объект. Для осуществления обратной задачи (удалить часть объекта, которая не перекрыта), используйте метод `IntersectFeatures`.

Метод `FeatureFactory.IntersectFeatures` (Объект `FeatureFactory`)

Назначение

Возвращает обособленный объект, не связанный со слоем, который представляет пересечение нескольких объектов.

Синтаксис

```
[Feature=]OBJECT.IntersectFeatures( feature1 [feature2] )
```

Часть	Описание
OBJECT	Объект <code>FeatureFactory</code> .
feature1	Объект <code>Feature</code> или объект коллекция <code>Features</code> , представляющий объекты типа линия или полигон.
feature2	Переменная: объект <code>Feature</code> или объект коллекция <code>Features</code> , представляющие объекты типа полигон или линия. Этот аргумент требуется только если аргумент <code>feature1</code> является просто одиночным объектом <code>Feature</code> .

Примечание

Этот метод пересекает объект `Feature` или `Features` с другим объектом `Feature` или `Features`, и возвращает результирующий объект в виде отдельного, неприкрепленного к слою, объекта.

Вы не можете использовать точечный или текстовый объект с этим методом. Поддерживаются только следующие комбинации типов объектов:

- Вы можете пересекать регионы другими регионами. Если регионы перекрываются, объект, возвращаемый этим методом тоже будет полигоном. Если полигоны не перекрываются, но соприкасаются границами, то пересечение будет полилинией, представляющей границу соприкосновения полигонов.
- Вы можете пересечь полигоны и линии. Если полигон покрывает как минимум часть линии, пересечение будет частью линии, которая покрыта полигоном.
- Вы можете пересечь линии другими линиями. Если две линии пересекаются, то пересечение будет точкой.

Обратите внимание, что два объекта могут совсем не пересекаться, в этом случае, величина, возвращаемая этим методом является пустым объектом; такой объект не может быть прикреплен к слою.

Метод `IntersectFeatures` возвращает часть объекта, который перекрывает другой объект. Чтобы получить часть объекта, который не перекрыт, используйте метод `EraseFeature`.

Метод `FeatureFactory.IntersectionPoints` (Объект `FeatureFactory`)

Назначение

Возвращает коллекцию точек (`Points`), содержащую точки, в которых пересекаются два объекта.

Синтаксис

`ОБЪЕКТ.IntersectionPoints(feature1, feature2, [flag])`

Часть	Описание
<code>object</code>	Объект <code>FeatureFactory</code> .
<code>feature1</code>	Объект <code>Feature</code> .
<code>feature2</code>	Объект <code>Feature</code> .
<code>flag</code>	Переменная: Величина <code>IntersectionPointConstants</code> ; смотрите ниже. Необязательный параметр; если он пропущен, по умолчанию будет <code>miIntersectAll</code> .

Примечание

Если два объекта непересекаются, коллекция `Points`, создаваемая этим методом будет пустой.

Аргумент `flag` может быть одним из трех значений `IntersectionPointConstants`, как описано ниже.

Параметр	Значение	Описание
<code>miIntersectCrossings</code>	9	Возвращает набор точек, в которых линейный сегмент из одного объекта пересекает линейный сегмент другого объекта, но объекты не имеют узлов в точках пересечения.
<code>miIntersectCommon</code>	10	Возвращает набор точек, в которых линейный сегмент из одного объекта пересекает линейный

		сегмент другого объекта и оба объекта имеют узел в точке пересечения.
miIntersectAll	11	Возвращает все точки, в которых объекты пересекаются, независимо от того, имеют ли они общие узлы в точке пересечения. Другими словами, возвращает все точки miIntersectCommon и все пересечения miIntersectCrossings.

Метод FeatureFactory.IntersectionTest (Объект FeatureFactory)

Назначение

Возвращает логическое значение, определяющее, удовлетворяют ли два объекта тесту на пересечение.

Синтаксис

ОБЪЕКТ.IntersectionTest(feature1, feature2, [flag])

Часть	Описание
ОБЪЕКТ	Объект FeatureFactory.
feature1	Объект Feature.
feature2	Объект Feature.
flag	Переменная: Величина IntersectionTestConstants; смотрите ниже. Необязательный параметр; если он пропущен, по умолчанию будет miIntersectFeature.

Примечание

Этот метод проверяет два объекта, что бы определить, пересекаются ли они. Возможны три варианта результатов теста; аргумент flag определяет, какой из типов тестирования осуществлен.

Параметр	Значение	Описание
miIntersectCentroidWithinFeature	0	Тест возвращает True если центрост feature1 попадает в feature2.
miIntersectFeature	1	Тест возвращает True если два объекта пересекаются в любой точке или имеют общие узлы.
miIntersectEntirelyWithinFeature	2	Тест возвращает True если feature1 целиком находится в feature2.

Объект и коллекция Field

Каждый DataSet имеет коллекцию полей (свойство DataSet.fields). Коллекция Fields представляет поля объекта DataSet. Коллекция Fields также используется как параметр для DataSets.Add и также для Themes.Add.

Свойства объекта

- Свойство Field.AggregationFunction
- Свойство Field.Name
- Свойство Field.Type

Свойства коллекции

- Свойство Fields.Count
- Свойство Fields.Item

Методы коллекции

- Метод Fields.Add
- Метод Fields.Remove
- Метод Fields.RemoveAll

Смотрите также

Объект DataSet

Метод DataSets.Add

Метод Themes.Add

Свойство Field.AggregationFunction (Объект Field)

Назначение

Это свойство определяет как MapX вычисляет значение в поле Field когда несколько значений привязаны к одному объекту. Это свойство доступное только для чтения, и оно имеет значение одной из констант AggregationFunctionConstants. Свойство инициализируется в методе Fields.Add.

Смотрите также

Метод Fields.Add

Свойство Field.Name (Объект Field)

Назначение

Имя поля(колонки). Это строковая величина. Это свойство открытое только для чтения. Это свойство, существующее по умолчанию для объекта Field.

Свойство Field.Type (Объект Field)

Назначение

Тип поля. Для этого берется значение FieldTypeConstants. Это свойство открытое только для чтения.

Метод Fields.Add (Коллекция Fields)

Назначение

Добавляет поле к коллекции Fields.

Синтаксис

```
[ Field= ]OBJECT.Add (DataSourceCol, [Name], [AggregateFunction], [Type])
```

Части

Описание

OBJECT	Это объект Fields.
DataSourceCol	Имя или индекс колонки с данными.
Name	Имя добавляемого поля (колонки).
AggregateFunction	Тип объединяющей функции. Значение: AggregationFunctionConstants
Type	Тип данных в колонке. Берется значение FieldTypeConstants.

Примечание

Коллекция Fields собирается при использовании метода DataSets.Add. Параметр Fields берется из коллекции Fields, и создается с использованием этого метода Add.

Параметр Type используется только с набором данных Unbound Datasets. Он игнорируется для всех других типов наборов данных DataSet.

Объединяющая функция по умолчанию это miAggregationIndividual для колонок со строковыми величинами и miAggregationSum для числовых колонок.

Метод Add не может использоваться для коллекции полей набора данных с того момента, как этот набор данных создан.

Смотрите также

Метод DataSets.Add

Свойство DataSets.Fields

Пример на Visual Basic

```
'This sample demonstrates the use of Datasets.Add and the RequestData
`event supported by MapX to build an unbound dataset, which
`allows MapX
`to access data whose format is known only to the programmer.
`In this
`example, the data is in the form of a two-dimensional array
`in VB.
```

```

Private Const kNumberOfRows = 3

Dim theData(1 To 3, 1 To 2) As Variant

Private Sub Form_Load()

    theData(1, 1) = "ME"                `Fill in the data to
    theData(2, 1) = "NH"                `be used by RequestData
    theData(3, 1) = "VT"

    theData(1, 2) = 100
    theData(2, 2) = 200
    theData(3, 2) = 300

    Map1.ZoomTo 800, -70.26, 44.05      `Zoom in on New England

End Sub

Private Sub Command1_Click()

    Dim flds As New MapXLib.Fields
    Dim ds As Dataset

    `Describe the structure of the unbound dataset:
    flds.Add "State", "State", miAggregationIndividual, _
    miTypeString
    flds.Add "Sales", "Sales", miAggregationSum, miTypeNumeric
    `Create the unbound dataset. The "RequestData" event will be
    `triggered to get the data to be used.
    Set ds = Map1.Datasets.Add(miDataSetUnbound, Nothing, _
    "My Dataset", "State", , "USA", flds)
    `Create a theme based on the "Sales" column in the unbound dataset
    ds.Themes.Add miThemeGradSymbol, "Sales", "My Theme"

```

```

End Sub

Private Sub Map1_RequestData(ByVal DataSetName As String, _
    ByVal Row As Long, ByVal Field As Integer, _
    Value As Variant, Done As Boolean)

    Done = False

    If DataSetName <> "My Dataset" Or Row > kNumberOfRows Then
        Done = True
    Else
        Value = theData(Row, Field)
    End If
End Sub

```

Свойство Fields.Count (Коллекция Fields)

Назначение

Число полей в коллекции Fields. Это свойство доступно только для чтения.

Свойство Fields.Item (Коллекция Fields)

Назначение

Возвращает объект Field из коллекции Fields. Берется только параметр, определяющий какое нужно поле. В качестве параметра можно указать или имя поля или его индекс (1 - первое значение). Это свойство, существующее по умолчанию для коллекции Fields.

Пример на Visual Basic

```

Dim fld As Field

Set fld = Map1.Datasets("My Dataset").Fields(1)

```


Пример на Delphi

```

Var fld: variant;

Begin

    fld:= Map1.Datasets.Item("MyDataset").Fields.Item(1);

End;

```

Пример на C++

```

void CSampleProjectView::Item() {
    try {
        CMapXField fld = m_Map.GetDatasets().Item("My
Dataset").GetFields().Item(1);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
}

```

Метод Fields.Remove (Коллекция Fields)

Назначение

Удаляет указанный объект Field из коллекции Fields.

Внимание: Если Вы удаляете объект, индексы коллекции перенумеровываются, заполняя образовавшийся пробел.

Синтаксис

OBJECT.Remove (index)

Часть

Описание

OBJECT

Это объект Fields.

index

Имя объекта Field или индекс, начиная с 1.

Пример на Visual Basic

```
`This sample demonstrates the use of the Add and Remove
`methods on the Fields collection. For a more practical
`example of how to use the Add method, look at the samples
`for the Datasets.Add method.
Private Sub Command1_Click()
    `Create a new, empty Fields collection. Note the
    `declaration as "MapXLib.Fields". This is to
    `prevent conflicts with the DAO "Fields" object.
    Dim flds As New MapXLib.Fields
    `Add several fields to the collection
    flds.Add "State", "State", miAggregationIndividual
    flds.Add "Sales", "Sum_of_sales", miAggregationSum
    flds.Add "Household", "Avg_Household", miAggregationAverage
    `Now, remove the third field from the collection:
    flds.Remove 3
End Sub
```

Пример на C++

```

void CSampleProjectView::AddRemove() {
    CMapXFields flds;

    // Creates a new, empty fields collection
    if(!flds.CreateDispatch(flds.GetClsid())) {
        TRACE0("Failed to Create Fields collection");
        return;
    }

    try {
        // Add several fields to the collection
        flds.Add("State","State",miAggregationIndividual);
        flds.Add("Sales","Sum_of_sales",miAggregationSum);
        flds.Add("Household","Avg_Houshold",miAggregationAverage);

        // Now, remove the third field from the collection
        flds.Remove(3);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Метод **Fields.RemoveAll** (Коллекция **Fields**)

Назначение

Удаляет все объекты **Field** из коллекции.

Синтаксис

ОБЪЕКТ.RemoveAll

Пример на **Visual Basic**

```
Private Sub btnFieldsRemoveAll_Click()
    Dim nFields As Integer
    Dim flds As New MapXLib.Fields
    nFields = flds.Count
    ' add two new fields to the fields collection
    flds.Add "State", "State", miAggregationIndividual, _
    miTypeString
    flds.Add "Sales", "Sales", miAggregationSum, miTypeNumeric
    nFields = flds.Count
    ' remove all of the fields
    flds.RemoveAll
    nFields = flds.Count
End Sub
```

Пример на **C++**

```
void CSampleProjectView::RemoveAllFields() {
    CMapXFields flds;
    if(!flds.CreateDispatch(flds.GetClsid())) {
        TRACE0("Failed to Create Fields collection");
    }
}
```

```

        return;
    }

    try {
        // Add two new fields to the fields collection
        flds.Add(ColeVariant("State"),ColeVariant("State"),
        ColeVariant((long)miAggregationIndividual),
        ColeVariant((long)miTypeString));

        flds.Add(ColeVariant("State"),ColeVariant("State"),
        ColeVariant((long)miAggregationSum),
        ColeVariant((long)miTypeNumeric));

        flds.RemoveAll(); // Remove all of the fields
    } catch (ColeDispatchException *e) {
        e->ReportError();

        e->Delete();
    } catch (ColeException *e) {
        e->ReportError();

        e->Delete();
    }
}

```

Объект Find

Назначение

Объект Find позволяет поиск объектов. Уточнение происходит за счет второй таблицы (second polygon table), используется карта улиц и алгоритм интерполяции.

Свойства объекта

- Свойство Find.Abbreviations (Объект Find)
- Свойство Find.CloseMatchMax (Объект Find)
- Свойство Find.ClosestAddress (Объект Find)
- Свойство Find.FindDataset (Объект Find)
- Свойство Find.FindField (Объект Find)
- Свойство Find.OtherBoundary (Объект Find)
- Свойство Find.RefineDataset (Объект Find)
- Свойство Find.RefineField (Объект Find)
- Свойство Find.RefineLayer (Объект Find)

Методы объекта

- Метод Find.SearchEx (Объект Find)
- Метод Find.Search (Объект Find)

Пример на Visual Basic

```
Dim FoundObj As FindFeature
Set Map1.Layers("US Major Cities").Find.RefineLayer = _
Map1.Layers("USA")
Set FoundObj = Map1.Layers("US Major _
Cities").Find.Search("Columbia", "SC")
If (FoundObj.FindRC Mod 10 = 1) Then
    Map1.Zoom = 100
    Map1.CenterX = FoundObj.CenterX
```

```

    Map1.CenterY = FoundObj.CenterY
Else
    MsgBox "No exact match found. " + Str$(FoundObj.FindRC)
End If

```

Пример на C++

```

void CSampleProjectView::SimpleFind() {
    try {
        CMapXFindFeature FoundObject = m_Map.GetLayers().
        Item("US Major Cities").GetFind().Search("Columbia","SC");
        if(FoundObject.GetFindRC() %10 == 1) {
            m_Map.SetZoom(100);
            m_Map.SetCenterX(FoundObject.GetCenterX());
            m_Map.SetCenterY(FoundObject.GetCenterY());
        } else
            AfxMessageBox("No exact match found.");
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Пример поиска с использованием Find Field и добавлением таблицы MI к коллекции Datasets на Visual Basic

```

Dim FoundObj As FindFeature

```

```

Dim Layer As Layer
Dim ds As Dataset
Dim Finds As Dataset
Map1.Layers.Add "C:\program files\mapinfo _
professional\data\usa\states.tab"
Set Layer = Map1.Layers("states")
Set Findds = Map1.Datasets.Add(miDataSetMITable, Layer)
Set Map1.Layers("states").Find.FindDataset = Findds
Set Map1.Layers("states").Find.FindField = Findds.Fields(2)
Set FoundObj = Map1.Layers("states").Find.Search("New York", _
Text2)
If (FoundObj.FindRC Mod 10 = 1) Then
    Map1.Zoom = 500
    Map1.CenterX = FoundObj.CenterX
    Map1.CenterY = FoundObj.CenterY
Else
    MsgBox "No exact match found. " + Str$(FoundObj.FindRC)
End If

```

Пример поиска с использованием Find Field и добавлением таблицы MI к коллекции Datasets на C++

```

void CSampleProjectView::ComplexFind(CString SearchString) {
    CMapXFindFeature FoundObj;
    CMapXLayer Layer;
    COleVariant LayerVariant;
    CMapXDataset ds, Findds;

```



```

try {

    m_Map.GetLayers().Add("C:\\Program Files\\MapInfo
    MapX\\Maps\\states.tab");

    Layer= m_Map.GetLayers().Item("states");

    LayerVariant.vt = VT_DISPATCH;

    LayerVariant.pdispVal = Layer.m_lpDispatch;


    Finddds = m_Map.GetDatasets().
    Add(miDataSetLayer,LayerVariant);

    m_Map.GetLayers().Item("states").GetFind().
    SetFindDataset(Finddds);

    m_Map.GetLayers().Item("states").GetFind().
    SetFindField(Finddds.GetFields().Item(2));


    FoundObj = m_Map.GetLayers().Item("states").GetFind().
    Search(SearchString,"New York");

    if(FoundObj.GetFindRC() % 10 == 1) {

        m_Map.SetZoom(500);

        m_Map.SetCenterX(FoundObj.GetCenterX());

        m_Map.SetCenterY(FoundObj.GetCenterY());

    } else

        AfxMessageBox("No exact match found.");

} catch (COleDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (COleException *e) {

    e->ReportError();

    e->Delete();

```

```

    }
}

// Geoset object and collection (Page 72)
void CSampleProjectView::Geoset() {
    try {
        CMapXGeoset gs = m_Map.GetGeosets().Item("United States");
        m_Map.SetCenterX(gs.GetCentroid().GetX());
        m_Map.SetCenterY(gs.GetCentroid().GetY());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

// LabelProperties Object (Page 77)
void CSampleProjectView::EditLabelProperties(CDaoRecordset&
SourceRecordset) {
    COleVariant variant;
    CMapXDataset ds;

    variant.vt = VT_DISPATCH;
    variant.pdispVal = SourceRecordset.m_pDAORecordset;

```

```

try {
    ds = m_Map.GetDatasets().Add(miDataSetDAO,variant);

    m_Map.GetLayers().Item("usa").GetLabelProperties().
        SetDataset(ds);

    m_Map.GetLayers().Item("usa").GetLabelProperties().
        SetDataField(ds.GetFields().Item("GEONAME"));
} catch (COleDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Свойство Find.Abbreviations (Объект Find)

Назначение

Позволяет пользователю набирать ST, STR или STREET, использовать принятые сокращения в английском языке, и MapX будет сравнивать все с ST. По умолчанию имеет значение TRUE.

Свойство Find.CloseMatchMax (Объект Find)

Назначение

Возвращает максимальное количество совпадений, загруженных в FindResult из SearchEx. По умолчанию равно 8.

Свойство Find.ClosestAddr (Объект Find)

Назначение

Логическое – подбирает последний номер дома, если запрашиваемый номер не найден.

Например, если пользователь запрашивает дом 120 ELM ST, но MapX знает только о домах 1–100 ELM ST, то MapX подберет номер дома 100 ELM ST. По умолчанию имеет значение FALSE.

Свойство Find.FindDataset (Объект Find)

Назначение

Поиск совпадающего набора данных полей по ключу. Если ключ не определен, используется первичный ключ слоя.

Свойство Find.FindField (Объект Find)

Назначение

Поиск совпадающего поля из Набора данных. Используется вместе со свойством FindDataset.

Смотрите также

Свойство FindDataset

Метод Find.SearchEx (Объект Find)

Назначение

Метод SearchEx расширяет функциональные возможности поиска. Это средство поиска для нахождения полного совпадения, возвращаемого в коллекцию. Разработчик может показать результат пользователю в виде списка. Вначале поиск пытается вернуть точное совпадение при тех параметрах, которые заданы методом. Если точного совпадения не найдено, то список MatchedFeature будет пустым и коллекция FindMatches будет содержать все полные совпадения, найденные методом FindMatch.

Синтаксис

[FindResult=]OBJECT.SearchEx (Address, [Boundary])

Части Описание

OBJECT Это объект Find.

Address Строковая величина, имя объекта или уличного адреса, который надо найти. Например, “Rensselaer, 6 Georgian Ct,” “London” или “3 Elm Street”.

Boundary Строковая величина, уточненное имя района, ограничивающего искомый объект. Используется когда указан слой по которому проводится уточнение. Например, “NY”, “12211”.
Если уточняющий слой установлен, то этот аргумент необходим, в других случаях он необязателен.

Пример

```
Public Sub SearchEx(txtCity As String, txtState As String)

Dim lyrCity As Layer

Dim dsCity As Dataset

Dim dsState As Dataset

Dim objFindResult As FindResult

Dim intNumCols As Integer

Dim intNumRows As Integer

Dim counter As Integer

Dim firstcounter As Integer

' This file ships as sample data with MapX.

Set lyrCity = Map1.Layers("US Minor Cities")

Set lyrState = Map1.Layers("USA")

' Add the datasets

Set dsCity = Map1.Datasets.Add(miDataSetLayer, lyrCity)
```

```

Set dsState = Map1.Datasets.Add(miDataSetLayer, lyrState)

'   Set the city dataset to be the find dataset
Set lyrCity.Find.FindDataset = dsCity
Set lyrCity.Find.FindField = dsCity.Fields("city")
'   Set the state dataset to be the refining dataset
Set lyrCity.Find.RefineLayer = lyrState
Set lyrCity.Find.RefineDataset = dsState
Set lyrCity.Find.RefineField = dsState.Fields("state")

Set objFindResult = lyrCity.Find.SearchEx(txtCity, txtState)

If objFindResult.MultipleMatches = True Then

    For Each objMatchCandidate In objFindResult.Matches
        firstcounter = firstcounter + 1
        List1.AddItem objMatchCandidate.Name

        intNumRows = objFindResult.Matches.Count
        intNumCols = dsCity.Fields.Count
        CityGrid.Cols = intNumCols
        CityGrid.Rows = intNumRows
        For counter = 1 To intNumCols
            CityGrid.Row = firstcounter - 1
            CityGrid.Col = counter - 1
            CityGrid.Text = dsCity.Value(objMatchCandidate.FeatureID,
            counter)
        Next
    Next

```

```

    Next
Else
    List1.AddItem objFindResult.MatchedFeature.Name
    intNumCols = dsCity.Fields.Count
    CityGrid.Cols = intNumCols

    For counter = 1 To intNumCols
        CityGrid.Col = counter - 1
        CityGrid.Text = dsCity.Value(
            objFindResult.MatchedFeature.FeatureID, counter)
    Next
End If

End Sub

```

Смотрите также

FindResult

FindMatches

FindMatch

Свойство Find.OtherBoundary (Объект Find)

Назначение

Логическое – когда используется уточняющая граница, это свойство определяет как MapX обходится с отсутствием совпадающих значений внутри уточняющей границы.

Когда установлено False, Find.Search будет возвращать только совпадающие значения если они происходят внутри уточняющей границы и возвращать отсутствие совпадений во всех других случаях. Когда установлено True, Find.Search будет сначала возвращать совпадения

внутри определяющей границы, если таких нет, будет возвращать совпадения за пределами уточняющей границы. По умолчанию устанавливается False.

Смотрите также

Свойство Find.RefineLayer

Метод Find.Search

Свойство Find.RefineDataset (Объект Find)

Назначение

Когда используется RefineLayer, Вы можете использовать геоколонку для сравнения или Вы можете вместо этого использовать Dataset и Field.

Свойство Find.RefineField (Объект Find)

Назначение

Используется вместе с RefineDataset.

Свойство Find.RefineLayer (Объект Find)

Назначение

Слой используемый для поиска по составному ключу. Уточнение полезно при многократном сравнении.

Например, в карте улиц может быть несколько BROADWAY – с различными значениями zipcode. Поэтому можно использовать для уточнения слой Zipcode. Когда идет поиск, то укажите улицу и почтовый код.

Смотрите также

Свойство Find.RefineField

Свойство Find.RefineDataset

Метод Find.Search (Объект Find)

Назначение

Ищет слой для адреса или для имени объекта на карте; возвращает объект FindFeature.

Синтаксис

ОБЪЕКТ.Search (Address [Boundary])

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Find.
Address	Строковая величина, имя объекта или почтовый адрес, который Вы ищете. Например, “Rensselaer, 6 Georgian Ct,”, “London” или “3 Elm Street”.
Boundary	Строковая величина, имя объекта границы, по которому проводится уточненный поиск. Используется, когда указан слой для уточненного поиска. Например, “NY”, “21135”. Если уточняющий слой установлен, то этот аргумент необходим, в противном случае от необязателен.

Например, смотрите раздел Find в электронном справочнике.

Смотрите также

Объект FindFeature

Объект FindFeature

Объект FindFeature это суперкласс Feature с добавлением строкового значения последнего поиска, который будет возвращен.

FindFeature возвращается методом поиска объекта Find и содержит все свойства объекта Feature и свойства FindRC.

Свойства объекта

- Свойство FindFeature.FindRC

Смотрите также

Объект Feature

Свойство FindFeature.FindRC (Объект FindFeature)

Назначение

FindRC это результирующий код операции поиска.

Для получения X,Y и/или индекса объекта ID используются свойства, CenterX, CenterY, FeatureID (смотрите, Feature objects. Если адрес был найден FeatureID будет 0.

<u>Значения</u>	<u>Смысл</u>
xx1	Точное совпадение
xx2	Используется вычитание из файла аббревиатур
xx3 (-)	Точное совпадение не найдено
xx4 (-)	Не указано имя объекта; совпадение не найдено
x1x	Сторона улицы неопределена
x2x (+ / -)	Номер дома был в пределах min/max
x3x (+ / -)	Номер дома был не в пределах min/max
x4x (+ / -)	Номер дома не был указан
x5x (-)	Улицы не пересекаются
x6x (-)	Совпадающие строки не имеют географического объекта

1xx (+ / -)	Имя найдено в другом регионе
2xx (-)	Имя найдено более чем в одном регионе
3xx (+ / -)	Не был указан точно регион и одно совпадение было найдено
4xx (-)	Не был указан регион и найдено много совпадений
5xx (+)	Имя было найдено более одного раза в указанном регионе

Объект FindMatch и коллекция FindMatches

Объект FindMatch возвращает совпадение, полученное из метода SearchEx определяющего полное совпадение или точное совпадение. Коллекция FindMatches пополняется наиболее точными совпадениями.

Свойства объекта

- Свойство FindMatch.Name (Объект FindMatch)
- Свойство FindMatch.FeatureID (Объект FindMatch)
- Свойство FindMatch.Score (Объект FindMatch)

Свойства коллекции

- Свойство FindMatches.Count (Коллекция FindMatches)
- Свойство FindMatches.Item (Коллекция FindMatches)

Смотрите также

Объект FindResult

Метод SearchEx

Свойство FindMatch.Name (Объект FindMatch)

Назначение

Возвращает имя объекта (например, улицы).

Свойство FindMatch.FeatureID (Объект FindMatch)

Назначение

Возвращает индекс ID объекта, если такой существует. Если не существует, возвращается ноль.

Свойство FindMatch.Score (Объект FindMatch)

Назначение

Возвращает количество совпадений или ноль если совпадений нет, например, -- 100 (точных совпадений).

Свойство FindMatches.Count (Коллекция FindMatches)

Назначение

Это свойство только для чтения, показывающее количество совпадений в коллекции.

Свойство FindMatches.Item (Коллекция FindMatches)

Назначение

Возвращает совпадающий адрес из коллекции FindMatches. Совпадающие адреса (объекты) расположены в коллекции FindMatches по порядку (лучшее совпадение первое).

Синтаксис

[FindMatch=]OBJECT.Item (index)

Части	Описание
-------	----------

OBJECT	Это коллекция FindMatches
--------	---------------------------

index	Цельночисленный индекс возвращаемого совпадения.
-------	--

Объект FindResult

Объект FindResult возвращает информацию о FindRC в форме свойств, которые облегчают доступ к искомому результату. FindResult содержит коллекцию объектов, содержащих множество найденных совпадений в том порядке, как они были найдены. Если найдено точное совпадение, то коллекция будет содержать только объект ExactMatch.

Свойства объекта

- Свойство FindResult.AddressRangeOutOfRange (Объект FindResult)
- Свойство FindResult.ExactMatch (Объект FindResult)
- Свойство FindResult.FindRC (Объект FindResult)
- Свойство FindResult.IntersectionNotFound (Объект FindResult)
- Свойство FindResult.MatchedFeature (Объект FindResult)
- Свойство FindResult.Matches (Объект FindResult)
- Свойство FindResult.MultipleMatches (Объект FindResult)
- Свойство FindResult.RefineRegion (Объект FindResult)
- Свойство FindResult.Substitute (Объект FindResult)

Смотрите также

Метод SearchEx

Свойство FindResult.AddressOutOfRange (Объект FindResult)

Назначение

Возвращает логическую величину если адресный номер находится в допустимом диапазоне.

Свойство FindResult.ExactMatch (Объект FindResult)

Назначение

Возвращает TRUE если возвращено точное совпадение.

Свойство `FindResult.FindRC` (Объект `FindResult`)

Назначение

Ищет слой для уличных адресов или для имени объекта на карте; возвращает объект `FindFeature`.

Свойство `FindResult.IntersectionNotFound` (Объект `FindResult`)

Назначение

Это свойство только для чтения, возвращает логическую величину, показывающую, пересекаются ли улицы.

Свойство `FindResult.MatchedFeature` (Объект `FindResult`)

Назначение

Возвращает объект точного совпадения `ExactMatch`.

Свойство `FindResult.Matches` (Объект `FindResult`)

Назначение

Возвращает итоговую коллекцию `Match`.

Свойство `FindResult.MultipleMatches` (Объект `FindResult`)

Назначение

Возвращает логическую величину, если найдено множество совпадений.

Свойство `FindResult.RefineRegion` (Объект `FindResult`)

Назначение

Возвращает логическую величину, если указан уточняющий регион поиска.

Свойство `FindResult.Substitute` (Объект `FindResult`)

Назначение

Возвращает логическую величину, если использована замена из файла с аббревиатурами.

Объект `Geoset` и коллекция `Geosets`

Объект `Geoset` построен из объекта `Map` и позволяет Вам определять рабочий набор. Рабочий набор `GeoSet` это коллекция слоев карт и их настроек. Объект коллекция `Geosets` это коллекция объектов `Geoset`.

Свойства объекта

- Свойство `Geoset.Centroid`
- Свойство `Geoset.PathName`
- Свойство `Geoset.UserName`

Свойства коллекции

- Свойство `Geosets.Count`
- Свойство `Geosets.Item`

Пример

```
Dim gs As Geoset
gs = Map1.Geosets("United States")
Map1.CenterX = gs.Centroid.X
```

```
Map1.CenterY = gs.Centroid.Y
```

Смотрите также

Объект Map

Свойство Geoset.Centroid (Объект Geoset)

Назначение

Объект Point определяет географический центр рабочего набора

Смотрите также

Объект Point

Свойство Geoset.PathName (Объект Geoset)

Назначение

Определяет путь к месту на компьютере где находится рабочий набор.

Свойство Geoset.UserName (Объект Geoset)

Назначение

Определяет имя рабочего набора.

Свойство Geosets.Count (Коллекция Geosets)

Назначение

Количество объектов Geoset в коллекции. Это свойство открытое только для чтения.

Свойство `Geosets.Item` (Коллекция `Geosets`)

Назначение

Возвращает объект `Geoset` из коллекции. Это переменная, и Вы можете указать имя `Geoset.Name` или числовой индекс от 1 до `Geosets.Count`. Это свойство, существующее по умолчанию для коллекции `Geosets`.

Объект `Graphic`

Назначение

Каждая аннотация `Annotation` содержит объект `Graphic` (Свойство `Annotation.Graphic`) в котором хранится символ или текст и местоположение.

Свойства объекта

- Свойство `Graphic.Caption`
- Свойство `Graphic.Position`
- Свойство `Graphic.Style`
- Свойство `Graphic.X`
- Свойство `Graphic.Y`

Смотрите также

Объект `Annotation`

Свойство `Graphic.Caption` (объект `Graphic`)

Назначение

Содержит текст если аннотация имеет тип `miTextAnnotation`. Это строковая величина.

Смотрите также

Свойство `Annotation.Type`

`AnnotationTypeConstants`

Свойство `Graphic.Position` (объект `Graphic`)**Назначение**

Содержит значение, указывающее, как текст должен быть расположен относительно координат X,Y. Используется только если тип аннотации `miTextAnnotation`. Свойство берет значение `PositionConstants`, по умолчанию оно равно `miPositionTL`.

Смотрите также

Свойство `Annotation.Type`

`PositionConstants`

Свойство `Graphic.Style` (объект `Graphic`)**Назначение**

Объект `Style` содержащий стиль символа или текст. Это свойство открытое только для чтения.

Смотрите также

Объект `Style`

Свойство `Graphic.X` (объект `Graphic`)**Назначение**

Содержит координату X аннотации. Это величина двойной точности и представляет собой географическую долготу. Если тип аннотации это `miTextAnnotation`, то позиция аннотации определяется координатами X,Y в сопряжении со свойством `Position`. Если тип аннотации это `miSymbolAnnotation`, координаты X,Y относятся к центру символа.

Смотрите также

Свойство `Annotation.Type`

`AnnotationTypeConstants`

Свойство `Graphic.Position`

Свойство `Graphic.Y` (объект `Graphic`)**Назначение**

Содержит координату Y аннотации. Это величина двойной точности, географическая широта. Если тип аннотации это `miTextAnnotation`, то позиция аннотации определяется координатами в объединении со свойством `Position`. Если тип аннотации это `miSymbolAnnotation`, координаты X,Y относятся к центру символа.

Смотрите также

Свойство `Annotation.Type`

`AnnotationTypeConstants`

Свойство `Graphic.Position`

Объект `IndividualValueCategory`, коллекция `IndividualValueCategories`

Настройки тематической карты отдельных значений хранятся в коллекции `IndividualValueCategories`, которая является коллекцией объектов `IndividualValueCategory` — один объект для каждого уникального значения в тематической карте.

Чтобы получить коллекцию `IndividualValueCategories`, обратитесь к свойству `ThemeProperties.IndividualValueCategories`.

Свойства объекта

- Свойство `IndividualValueCategory.NumItems`
- Свойство `IndividualValueCategory.Style`

- Свойство IndividualValueCategory.Value

Свойства коллекции

- Свойство IndividualValueCategories.AllOthersCategory
- Свойство IndividualValueCategories.Count
- Свойство IndividualValueCategories.Item

Смотрите также

Объект Legend

Свойство ThemeProperties.IndividualValueCategories

Свойство IndividualValueCategories.AllOthersCategory (Коллекция IndividualValueCategories)

Возвращает один объект IndividualValueCategory из коллекции; этот объект описывает категорию для всех диапазонов а не для отдельных значений тематической карты. Объект IndividualValueCategory возвращается со свойством AllOthersCategory, и имеет свойство Value, которое не определено и по умолчанию имеет значение пустого строка (“”). Таким образом, Вы можете установить его свойство Style как показано в примере ниже. Текст, который появляется в легенде для объекта AllOthersCategory может быть установлен со свойством LegendTexts.AllOthersText.

Пример

```
Dim ds As Dataset
Dim lyr as Layer
Dim thm As MapXLib.Theme
Dim styl As MapXLib.Style
Dim AllOthers As MapXLib.IndividualValueCategory
Set lyr = Map1.Layers.Add("states.tab", 1)
Set ds = Map1.Datasets.Add(miDataSetLayer, Map1. _
Layers("STATES"))
```

```
ds.Themes.Add miThemeIndividualValue, "STATE",  
"My Theme", False  
Set thm = ds.Themes("My Theme")  
thm.Visible = True  
  
Set styl = Map1.DefaultStyle  
styl.PickRegion  
  
Set AllOthers = _  
    thm.ThemeProperties.IndividualValueCategories. _  
    AllOthersCategory  
AllOthers.Style = styl
```

Свойство `IndividualValueCategories.Count` (коллекция `IndividualValueCategories`)

Назначение

Целая величина, открытая только для чтения, показывающая количество уникальных индивидуальных значений в тематическом слое.

Свойство `IndividualValueCategories.Item` (коллекция `IndividualValueCategories`)

Назначение

Возвращает объект `IndividualValueCategory` из коллекции; который описывает один из рангов значений тематического слоя.

Свойство `IndividualValueCategory.NumItems` (Коллекция `IndividualValueCategories`)

Назначение

`NumItems` это количество объектов тематического слоя индивидуальных значений, которые попадают в данную категорию. Это целое число. Свойство открыто только для чтения.

Внимание: Это свойство действует только когда `MapX` автоматически генерирует тематику (`Theme.ComputeTheme = TRUE`). Если Вы вручную вычисляете тематическую карту (`Theme.ComputeTheme = FALSE`), то свойство вернет `NumItems NOT` что будет относиться к числу разделов в категории - то есть результат будет неопределен.

Свойство `IndividualValueCategory.Style` (Коллекция `IndividualValueCategories`)

Назначение

Оно определяет стиль для категории. Это объект `Style` и может быть применен к существующему объекту стиля или может устанавливать свойства стиля индивидуально. По умолчанию настройки обеспечивает высококонтрастный стиль.

Смотрите также

Объект `Style`

Свойство `IndividualValueCategory.Value` (объект `IndividualValueCategory`)

Назначение

Это свойство получает или устанавливает значения, используемые в тематическом слое индивидуальных значений. Это свойство может быть установлено, только если свойство `Theme.ComputeTheme` установлено на значение `FALSE`.

Внимание: установка свойства `Value` не действует на коллекцию `Legend.LegendTexts` для тематического слоя, которую нужно обновить отдельно.

Синтаксис

Свойство `IndividualValueCategory.Value`

Объект LabelProperties

Объект LabelProperties содержит свойства, которые контролируют, как прорисовываются подписи на слое.

Свойства объекта

- Свойство LabelProperties.DataField (Объект LabelProperties)
- Свойство LabelProperties.DataSet (Объект LabelProperties)
- Свойство LabelProperties.Duplicate (Объект LabelProperties)
- Свойство LabelProperties.LabelMax (Объект LabelProperties)
- Свойство LabelProperties.LabelZoom (Объект LabelProperties)
- Свойство LabelProperties.LabelZoomMax (Объект LabelProperties)
- Свойство LabelProperties.LabelZoomMiny (Объект LabelProperties)
- Свойство LabelProperties.LineType (Объект LabelProperties)
- Свойство LabelProperties.Offset (Объект LabelProperties)
- Свойство LabelProperties.Overlap (Объект LabelProperties)
- Свойство LabelProperties.Parallel (Объект LabelProperties)
- Свойство LabelProperties.PartialSegments (Объект LabelProperties)
- Свойство LabelProperties.Position (Объект LabelProperties)
- Свойство LabelProperties.Style (Объект LabelProperties)
- Свойство LabelProperties.Visible (Объект LabelProperties)

Пример

```
Dim DS As Object

Dim DB As Object

Dim RS As Object

Dim Temp As Object

Set DB = Workspaces(0).OpenDatabase("C:\Program _
Files\MapInfo MapV2\Data\Mapstats.mdb")
```



```

Set RS = DB.OpenRecordset("USA")

Set DS = Map1.Datasets.Add(miDataSetDAO, RS)

Set Map1.Layers("usa").LabelProperties.Dataset = DS

Set Temp = DS.Fields("GEONAME")

Set Map1.Layers("usa").LabelProperties.DataField = Temp

```

Смотрите также

Объект Layer

Свойство **LabelProperties.DataField** (объект **LabelProperties**)

Назначение

Устанавливает объект DataField то есть поле из которого берутся значения для подписей. Поле используемое для подписывания определяется как этим свойством, так и свойством DataSet определяющим какой набор данных DataSet используется. Для этого используется объект Field.

Пример на C++

```

// LabelProperties.Dataset Property
// LabelProperties.DataField Property
// Layer.AutoLabel Property
// LabelProperties.Position Property
// LabelProperties.Offset Method
void CSampleProjectView::LabelStates() {
    // Label the states with their complete name
    CMapXLayer USALayer = m_Map.GetLayers().Item("USA");
    CMapXDataset USData;
    COleVariant lyrVt;

    // Make a variant for the parameter to datasets.Add
    lyrVt.vt = VT_DISPATCH;
}

```

```

lyrVt.pdispVal = USALayer;
lyrVt.pdispVal->AddRef();

try {
    // Add the layer as a dataset to bring in the State_Name field
    USData = m_Map.GetDatasets().Add(miDataSetLayer,lyrVt);

    USALayer.SetAutoLabel(TRUE);

    // Label with the "State_Name" field of the dataset we just added
    USALayer.GetLabelProperties().SetDataset(USData);
    USALayer.GetLabelProperties().SetDataField(USData.GetFields().Item("State_Name"));

    // Position the label 4 points to the bottom right of the center of
the State
    USALayer.GetLabelProperties().SetPosition(miPositionBR);
    USALayer.GetLabelProperties().SetOffset(4);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Пример на Visual Basic

```

Private Sub LabelStates()
    ' Label the states with their complete name

    Dim USALayer As Layer
    Dim USData As Dataset

    Set USALayer = Map1.Layers.Item("USA")

    ' Add the layer as a dataset to bring in the State_Name field
    Set USData = Map1.Datasets.Add(miDataSetLayer, USALayer)

    ' Label with the "State_Name" field of the dataset we
    ' just added
    Set USALayer.LabelProperties.Dataset = USData
    Set USALayer.LabelProperties.DataField = _
    USData.Fields.Item("State_Name")

    ' Position the labels 4 points to the bottom right of the
    ' center of the State
    USALayer.LabelProperties.Position = miPositionBR
    USALayer.LabelProperties.Offset = 4
    USALayer.AutoLabel = True
End Sub

```

Смотрите также

Свойство LabelProperties.DataSet

Объект Field

Свойство LabelProperties.DataSet (объект LabelProperties)

Назначение

Устанавливает набор данных Dataset используемый для подписывания. Поле, используемое для подписывания определяется и этим свойством и свойством DataField указывающим, какое поле в наборе данных DataSet используется. Если не определить DataSet то подписывание состоится по первому полю.

Смотрите также

Свойство LabelProperties.DataField

Объект DataSet

Свойство LabelProperties.Duplicate (объект LabelProperties)

Назначение

Проверяет есть ли одновременно повторяющиеся подписи на карте. Это логическая величина, и по умолчанию она равна False.

Свойство LabelProperties.LabelMax (объект LabelProperties)

Назначение

Определяет максимальное количество подписей, автоматически генерируемых для слоя. Это целая величина, по умолчанию равная 100.

Свойство `LabelProperties.LabelZoom` (объект `LabelProperties`)

Назначение

Определяет, изменяется ли масштаб подписей при изменении масштаба карты. Определяет масштабный фактор - в каких пределах возможно отображение подписей. Если `LabelZooming` установлено, то используются значения, которые хранятся в свойствах `LabelZoomMax` и `LabelZoomMin`. Это логическая величина и по умолчанию она имеет значение `False`.

Смотрите также

Свойство `LabelProperties.LabelZoomMax`

Свойство `LabelProperties.LabelZoomMin`

Свойство `LabelProperties.LabelZoomMax` (объект `LabelProperties`)

Назначение

Если установлено `ZoomLayering` (Свойство `LabelProperties.LabelZoom`), то настоящее свойство указывает максимальное значение масштаба для отображения подписей. Для этого берется величина двойной точности в единицах карты (`Map.MapUnit`). Подписи появятся если `zoom >= min` и `< max`.

Смотрите также

Свойство `LabelProperties.LabelZoom`

Свойство `Map.MapUnit`

Свойство `LabelProperties.LabelZoomMin` (объект `LabelProperties`)

Назначение

Если установлено `ZoomLayering` (Свойство `LabelProperties.LabelZoom`), то настоящее свойство указывает минимальное значение масштаба для отображения подписей. Для этого берется

величина двойной точности в единицах карты (Map.MapUnit). Подписи появятся если `zoom >= min` и `< max`.

Смотрите также

Свойство `LabelProperties.LabelZoom`

Свойство `Map.MapUnit`

Свойство `LabelProperties.LineType` (объект `LabelProperties`)

Назначение

Определяет тип линии, используемой для связки подписи и объекта. Свойство используется когда `Offset` имеет большое значение. Берется значение `LineTypeConstants` и по умолчанию будет `miLineTypeNone`.

Смотрите также

`LineTypeConstants`

Свойство `LabelProperties.Offset`

Свойство `LabelProperties.Offset` (объект `LabelProperties`)

Назначение

Определяет расстояние между объектом и его подписью. Это полезно для символьных объектов, когда Вам нужна подпись на некотором расстоянии от объекта. Это целая величина, определяющая расстояние в пикселях. По умолчанию расстояние равно 5.

Свойство `LabelProperties.Overlap` (объект `LabelProperties`)

Назначение

Позволяет перекрываться подписям на одном слое. Это логическая величина, по умолчанию равная `False`.

Свойство `LabelProperties.Parallel` (объект `LabelProperties`)

Назначение

Определяет, должны ли подписи располагаться параллельно линейным объектам. Это логическая величина и по умолчанию она имеет значение `True`.

Свойство `LabelProperties.PartialSegments` (Объект `LabelProperties`)

Назначение

Установка этого свойства на `True` позволяет MapX подписывать полилинии, даже если в окне карты видна только малая часть полилинии. Таким образом, это свойство не работает для других объектов. Это логическая величина, по умолчанию равная `True`.

Свойство `LabelProperties.Position` (объект `LabelProperties`)

Назначение

Устанавливает положение подписи относительно объекта карты. Берется значение `PositionConstants`.

Значение по умолчанию зависит от `PredominantObjectType`:

`miFeatureTypeSymbol` это `miPositionCL`

`miFeatureTypeLine` это `miPositionBC`

`miFeatureTypeRegion` это `miPositionCC`

Свойство `LabelProperties.Style` (объект `LabelProperties`)

Назначение

Устанавливает стиль используемый для подписи. По умолчанию это `Helvetica 10 pt`.

Свойство `LabelProperties.Visible` (объект `LabelProperties`)

Назначение

Определяет, являются ли подписи видимыми для слоя. Подписи могут быть созданы используя свойство `Layer.Autolabel`, или вручную используя метод `Layer.LabelAtPoint` или используя инструмент подпись. Это свойство `Visible` определяет, являются ли все подписи для слоя видимыми или нет. Берется логическая величина, по умолчанию это `True`.

Объект `Layer` и коллекция `Layers`

Каждая карта имеет коллекцию слоев. Коллекция `Layers` хранится в объекте `Layers`. Коллекция `Layers` имеет методы и свойства используемые для добавления и удаления объектов `Layer` из коллекции.

Компьютерные карты организованы в слои. Каждый слой несет свою нагрузку. Каждый рабочий набор `GeoSet` изображается как отдельный слой. Каждый слой содержит различные объекты, такие как полигоны, точки, линии и текст.

Например, один слой может содержать границы штатов, второй слой может иметь символы, показывающие столицы, третий слой может содержать текстовые подписи. Объединив эти слои, Вы получаете карту. Можно одновременно отображать разное количество слоев. Слои карты формирует блоки карт в `MapX`. Как только Вы создали карту слоев, Вы можете настраивать слои различными способами, добавлять слои, удалять их и менять их порядок. Вы можете устанавливать все настройки слоя используя команды и свойства коллекции и объекта `Layer` а также объекта `LabelProperties`.

Изменение порядка следования слоев

Важно правильно располагать слои. Например, у Вас есть слой с точками, отображающими адреса покупателей и слой с данными переписи. Если слои расположены неправильно, то `MapX` может прорисовать точки первыми, а потом площадные объекты, которые перекроют эти точки и они будут невидны.

Масштаб слоев

Иногда необходимо отобразить слой в определенном масштабе. Zoom Layering определяет, в каких пределах слой будет видимым.

Например, у Вас есть слой улиц и слой почтовых кодов ZIP Code (границы). Когда Вы делаете масштаб в 10 миль, улицы выглядят как черные пятна. Это потому, что при таком масштабе линии улиц сливаются. Используйте Zoom Layering , чтобы сообщить MapX показывать слой улиц только в определенном масштабном диапазоне, например от 5 миль.

Масштабный фактор, то есть те пределы, в которых отображается слой, можно настраивать для каждого слоя.

Свойства объекта

- Свойство Layer.AutoLabel (Объект Объект Layer)
- Свойство Layer.Bounds (Объект Layer)
- Свойство Layer.CoordSys (Объект Layer)
- Свойство Layer.DataSets (Объект Layer)
- Свойство Layer.DrawLabelsAfter (Объект Layer)
- Свойство Layer.FileSpec (Объект Layer)
- Свойство Layer.Find (Объект Layer)
- Свойство Layer.KeyField (Объект Layer)
- Свойство Layer.LabelProperties (Объект Layer)
- Свойство Layer.Name (Объект Layer)
- Метод Layer.NoFeatures (Объект Layer)
- Свойство Layer.OverrideStyle (Объект Layer)
- Свойство Layer.PredominantFeatureType (Объект Layer)
- Свойство Layer.Selectable (Объект Layer)
- Свойство Layer.Selection (Объект Layer)
- Свойство Layer.ShowCentroids (Объект Layer)
- Свойство Layer.ShowLineDirection (Объект Layer)
- Свойство Layer.ShowNodes (Объект Layer)
- Свойство Layer.Style (Объект Layer)
- Свойство Layer.Type (Объект Layer)
- Свойство Layer.Visible (Объект Layer)
- Свойство Layer.ZoomLayer (Объект Layer)
- Свойство Layer.ZoomMax (Объект Layer)
- Свойство Layer.ZoomMin (Объект Layer)

Методы объекта

- Метод Layer.AddFeature method (Объект Layer)
- Метод Layer.AllFeatures method (Объект Layer)
- Метод Layer.BeginAccess method (Объект Layer)
- Метод Layer.ClearCustomLabels method (Объект Layer)
- Свойство Layer.Editable (Объект Layer)
- Метод Layer.EndAccess (Объект Layer)
- Метод Layer.Search (Объект Layer)
- Метод Layer.DeleteFeature (Объект Layer)
- Метод Layer.DrilldownAddFeatures (Объект Layer)
- Метод Layer.DrilldownRemoveFeatures (Объект Layer)
- Метод Layer.DrilldownReset (Объект Layer)
- Метод Layer.FeatureIDFromFeatureName (Объект Layer)
- Метод Layer.FeatureKeyFromFeatureName (Объект Layer)
- Метод Layer.GetDrilldownFeaturesByID (Объект Layer)
- Метод Layer.GetFeatureByID (Объект Layer)
- Метод Layer.GetFeatureByKey (Объект Layer)
- Метод Layer.Invalidate (Объект Layer)
- Метод Layer.LabelAtPoint (Объект Layer)
- Метод Layer.Refresh (Объект Layer)
- Метод Layer.SearchAtPoint (Объект Layer)
- Метод Layer.SearchWithinDistance (Объект Layer)
- Метод Layer.SearchWithinFeature (Объект Layer)
- Метод Layer.SearchWithinRectangle (Объект Layer)
- Метод Layer.UpdateFeature (Объект Layer)

Свойства коллекции

- Свойство Layers.AnimationLayer (Коллекция Layers)
- Свойство Layers.Bounds (Коллекция Layers)
- Свойство Layers.Count (Коллекция Layers)
- Свойство Layers.InsertionLayer (Layers object)
- Свойство Layers.Item (Коллекция Layers)

Методы коллекции

- Метод Layers.Add (Коллекция Layers)

- Метод `Layers.AddGeosetLayers` (Коллекция `Layers`)
- Метод `Layers.AddServerLayer` (Коллекция `Layers`)
- Метод `Layers.AddUserDrawLayer` (Коллекция `Layers`)
- Метод `Layers.ClearSelection` (Коллекция `Layers`)
- Метод `Layers.CreateLayer` (Коллекция `Layers`)
- Метод `Layers.LayersDlg` (Коллекция `Layers`)
- Метод `Layers.Move` (Коллекция `Layers`)
- Метод `Layers.Remove` (Коллекция `Layers`)
- Метод `Layers.RemoveAll` (Коллекция `Layers`)

Смотрите также

Объект `Map`

Метод `Layer.AddFeature` (Объект `Layer`)

Назначение

Создает и возвращает новый объект на слое, со свойствами, указанными в объекте `Source`. Этот метод полезен для редактирования объектов.

Синтаксис

`ОБЪЕКТ.AddFeature (Source, [RowValues])`

Часть

Описание

`ОБЪЕКТ`

Это объект `Layer`.

`Source`

`Source` это объект `Feature`.

`RowValues`

Это коллекция `RowValues` и содержит значения из колонки данных.

Смотрите также

Метод `Layer.AllFeatures`

Метод `Layer.DeleteFeature`

Метод `Layer.Invalidate`

Метод `Layer.NoFeatures`

Метод Layer.SearchAtPoint

Метод Layer.SearchWithinDistance

Метод Layer.SearchWithinFeature

Метод Layer.SearchWithinRectangle

Метод Layer.UpdateFeature

Свойство Feature.FeatureKey

Пример на Visual Basic

```
Private Sub Map1_ToolUsed(ByVal ToolNum As Integer, ByVal X1 _
As Double, ByVal Y1 As Double, ByVal X2 As Double, ByVal Y2 _
As Double, ByVal Distance As Double, ByVal Shift As Boolean, _
ByVal Ctrl As Boolean, EnableDefault As Boolean)
Dim f As New Feature
If ToolNum = miArrowTool Then
    f.Type = miFeatureTypeSymbol
    f.Style = Map1.DefaultStyle
    f.Point.Set X1, Y1
    Map1.Layers("Fred").AddFeature f
End If
End Sub
```

Пример на C++

```
void CSampleProjectView::OnToolUsed(short ToolNum, double X1, double Y1,
double X2, double Y2, double Distance, BOOL Shift, BOOL Ctrl, BOOL FAR*
EnableDefault) {
    CMapXFeature fea;

    if(!fea.CreateDispatch(fea.GetClsid())) {
```

```

        TRACE0("Failed to Create Feature object");
        return;
    }
    if (ToolNum==miArrowTool) {
        try {
            fea.SetType(miFeatureTypeSymbol);
            fea.SetStyle(m_Map.GetDefaultStyle());
            fea.GetPoint().Set(X1,Y1);
            m_Map.GetLayers().Item("Fred").AddFeature(fea);
        } catch (COleDispatchException *e) {
            e->ReportError();
            e->Delete();
        } catch (COleException *e) {
            e->ReportError();
            e->Delete();
        }
    }
}

///// Layer.SearchAtPoint method (Page 91)
void CSampleProjectView::OnMouseDown(short FAR* Button, short FAR*
Shift, float FAR* X, float FAR* Y) {
    double longitude,latitude;

    CMapXFeatures fs;

    CMapXPoint pnt;

    // Create the new point

```

```

if(!pnt.CreateDispatch(pnt.GetClsid())) {
    TRACE0("Failed to Create Point object");
    return;
}

try {
    if(m_Map.GetCurrentTool() == miArrowTool) {
        m_Map.ConvertCoord(X,Y,&longitude,&latitude,miScreenToMap);
        pnt.Set(longitude,latitude);

        fs = m_Map.GetLayers().Item("US Top 20
Cities").SearchAtPoint(pnt);
        if(fs.GetCount() > 0) {
            CString msg;
            msg.Format("%d: %s",fs.GetCount(),fs.Item(1).GetName());
            AfxMessageBox(msg);
        } else
            AfxMessageBox("Nothing Found");
    }
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

```
// Layers.AnimationLayer property (Page 99)
void CSampleProjectView::AnimationLayer() {
    try {
        m_Map.GetLayers().SetAnimationLayer(m_Map.GetLayers().Item(3));
        long LayerCount = m_Map.GetLayers().GetCount()
        for(long i=1;i<=LayerCount;i++) {
            if(m_Map.GetLayers().GetAnimationLayer().m_lpDispatch ==
m_Map.GetLayers().Item(i).m_lpDispatch) {
                //...
            }
        }
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Метод Layer.AllFeatures (объект Layer)

Назначение

Возвращает объект с карты со всеми его атрибутами.

Синтаксис

```
[ Features= ]OBJECT.AllFeatures ()
```

Смотрите также

Метод `Layer.AddFeature`

Метод `Layer.DeleteFeature`

Метод `Layer.Invalidate`

Метод `Layer.NoFeatures`

Метод `Layer.SearchAtPoint`

Метод `Layer.SearchWithinDistance`

Метод `Layer.SearchWithinFeature`

Метод `Layer.SearchWithinRectangle`

Метод `Layer.UpdateFeature`

Метод `BeginAccess` (Объект `Layer`)

Назначение

Метод `BeginAccess` открывает таблицу для операций чтения и записи. Это улучшает работу для повторяющихся операций со слоями и наборами данных. Вам надо вызывать `Layer.EndAccess` для каждого вызова `Layer.BeginAccess`.

Синтаксис

`ОБЪЕКТ.BeginAccess (BeginAccessType)`

Части

Описание

`ОБЪЕКТ` Это объект `Layer`.

`beginAccessType` Одна из констант `LayerBeginAccessConstants`, указывающая тип доступа (только для чтения или чтение-запись).

Примечание

Программы MapX написанные без `BeginAccess` и `EndAccess` будут работать как и раньше, но без улучшения работы для повторяющихся операций.

Вы также можете вставить в такую программу вызов `BeginAccess`, но только первый вызов будет иметь действие.

Смотрите также

`Layer.EndAccess`

Метод `Layer.ClearCustomLabels` (объект `Layer`)

Назначение

Удаляет все подписи с текущего слоя.

Синтаксис

`ОБЪЕКТ.ClearCustomLabels`

Свойство `Layer.Editable` (Объект `Layer`)

Это свойство определяет, является ли выбранный слой редактируемым (можно перемещать, менять размер или удалять объекты) Это логическая величина, по умолчанию `False`. Инструмент Стрелка (`ArrowTool`) может использоваться для перемещения или изменения размеров выделенных объектов. Клавиша `delete` удаляет выделенные объекты.

Метод `Layer.EndAccess` (Объект `Layer`)

Назначение

Метод `EndAccess` завершает операцию чтения/записи таблицы. Вам надо вызвать `Layer.EndAccess` после каждого вызова `Layer.BeginAccess`.

Синтаксис

`ОБЪЕКТ.EndAccess (endAccessType)`

Часть

Описание

`ОБЪЕКТ` Это объект `Layer`.

`endAccessType` Одна из констант `LayerEndAccessConstants`, указывающая тип

конца доступа.

Примечание

Если Layer.BeginAccess была вызвана с beginAccessType или miAccessRead или miAccessReadWrite, то параметр EndAccessType должен иметь значение miAccessEnd. Программы MapX написанные без использования BeginAccess и EndAccess будут работать как и раньше, но без улучшения работы с повторяющимися операциями.

Смотрите также

Метод BeginAccess

Метод Layer.Search (Объект Layer)

Назначение

Метод Layer.Search обладает мощностью запросов SQL. Составленное выражение оценивается для каждой строки в таблице Layers, и возвращается коллекция Features, те ее объекты, для которых составленное выражение верно.

Синтаксис

```
[ Features= ]OBJECT.Search (strWhere)
```

Части Описание

OBJECT Это объект Layer.

strWhere Строковая: Выражение оценивающее каждый объект.

Пример на Visual Basic

```
Dim lyrUSA As MapXLib.Layer
Dim ftrs As MapXLib.Features

Set lyrUSA = Map1.Layers.Add("USA")

Map1.Datasets.Add miDataSetDAO, Data1.Recordset, "TestDataset",
"GEOABBR", , lyrUSA, , False
```

```
Set ftrs = lyrUSA.Search("TOTPOP > 1000000")

'*** The features collection will contain all of the states that have a
total population greater than 1000000.
```

Смотрите также

Выражения MapX

Свойство `Layer.AutoLabel` (Объект `Layer`)

Назначение

Определяет, установлен ли режим автоматического подписывания. Это логическая величина, по умолчанию равная `False`.

Свойство `Layer.Bounds` (Объект `Layer`)

Назначение

Возвращает объект прямоугольник `Rectangle` минимальный описывающий прямоугольник для всех географических объектов на слое; не применяется к слоям, созданным пользователем.

Пример на C++

```
// Layer.Bounds Property

void CSampleProjectView::ViewEntireLayer(CMapXLayer& ViewLayer) {

    // Set the boundaries of the map view to be the boundaries of the
    given layer

    try {

        m_Map.SetBounds(ViewLayer.GetBounds());

    } catch (COleDispatchException *e) {

        e->ReportError();

        e->Delete();

    } catch (COleException *e) {
```

```

        e->ReportError();
        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub ViewEntireLayer(ViewLayer As MapXLib.Layer)
    ' Set the boundaries of the map view to be the boundaries
    ' of the given layer
    Map1.Bounds = ViewLayer.Bounds
End Sub

```

Примечание

Это свойство полезно если Вам надо изменить масштаб так, чтобы показать все объекты на слое.

Смотрите также

Layers.Bounds

Свойство Layer.CoordSys (Объект Layer)

Назначение

Возвращает объект CoordSys, показывающий систему координат в которой сохранен слой карты, открыт только для чтения.

Смотрите также

Объект CoordSys

Map.DisplayCoordSys

Map.NumericCoordSys

Свойство Layer.DataSets (Объект Layer)

Назначение

Это свойство открытое только для чтения, то есть коллекция DataSets которая связана с определенным слоем карты. Эта коллекция является подмножеством полной коллекции DataSets (Map.DataSets). Объекты внутри Layer.DataSets те же самые, что и в Map.DataSets.

Примечание

Слои, не имеющие наборов данных (такие как растровые или созданные пользователем UserDraw) будут возвращать пустую коллекцию.

Layer.DataSets не поддерживает методы Add или Restore. При использовании возникает исключительная ситуация.

Добавляет набор данных к Map.DataSets через Datasets.Add и также добавляет этот набор данных к коллекции DataSets связанного слоя. Удаление набора данных из коллекции Layer.DataSet также приводит к удалению его из Map.DataSets, и наоборот.

Смотрите также

Метод DataSets.Add

Map.DataSets

Метод Layer.DeleteFeature (Объект Layer)

Назначение

Удаляет объект и его атрибуты в базе данных со слоя.

Синтаксис

ОБЪЕКТ.DeleteFeature (FeatureKey)

Часть

Описание

ОБЪЕКТ

Это объект Layer.

FeatureKey

Объект Feature подлежащий удалению. Этот параметр может быть или объектом Feature или ключом FeatureKey.

Смотрите также

Метод `Layer.AddFeature`

Метод `Layer.AllFeatures`

Метод `Layer.Invalidate`

Метод `Layer.NoFeatures`

Метод `Layer.SearchAtPoint`

Метод `Layer.SearchWithinDistance`

Метод `Layer.SearchWithinFeature`

Метод `Layer.SearchWithinRectangle`

Метод `Layer.UpdateFeature`

Свойство `Layer.DrawLabelsAfter` (Объект `Layer`)

Назначение

Логическая величина. Когда установлено `True`, подписи для всех слоев начиная с нижнего (или последнего слоя, установленного этим свойством) включаются в сам этот слой и прорисовываются после прорисовки самого слоя. Это позволяет лучше контролировать какое место занимают подписи по отношению к объектам.

По умолчанию, слои имеют это свойство установленное на `False`, и подписи прорисовываются после самого верхнего слоя, но до слоя анимации.

Метод `Layer.DrilldownAddFeatures` (объект `Layer`)

Назначение

Добавляет объекты к сборочному слою (`Drilldown`).

Синтаксис

`ОБЪЕКТ.DrilldownAddFeatures (Level, FeatureKeys)`

Часть

Описание

ОБЪЕКТ	Объект Layer связанный со сборочным (Drilldown) слоем.
Level	Строковая величина, показывающая, какой уровень в сборочном слое содержит объекты, которые будут добавлены.
FeatureKeys	Переменная: Строковая величина или массив строковых величин, идентифицирующих индексы ID объектов сборочного слоя, которые будут добавлены. Обратите внимание, что эти индексы сборочного слоя ID (то есть значения из колонки во сборочном компоненте таблицы), не являются внутренними значениями FeatureID объектов MapX.

Примечание

Этот метод не может поменять содержание таблиц. Объекты, которые Вы добавляете, уже есть в таблице; при "добавлении" их к слою, Вы делаете его видимым.

Метод Layer.DrilldownRemoveFeatures (объект Layer)

Назначение

Удаляет объекты со сборочного слоя.

Синтаксис

ОБЪЕКТ.DrilldownRemoveFeatures (Level, FeatureKeys)

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Layer который соотносится со сборочным слоем (Drilldown layer).
Level	Строковая величина, указывающая какой уровень содержит объекты подлежащие удалению.
FeatureKeys	Переменная: Строковая величина или массив строковых величин, определяющих индексы сборочных объектов ID, подлежащих удалению. Обратите внимание, что эти индексы указывают первую колонку во сборочной таблице, и не являются внутренними значениями FeatureID объектов MapX Feature.

Примечание

Метод не изменяет содержание никаких таблиц. Объекты, которые Вы удаляете, не удаляются из таблиц; при ”удалении” их, Вы просто делаете их невидимыми.

МетодLayer.DrilldownReset (объект Layer)

Назначение

Стирает все объекты со сборочного слоя, и переинициализирует слой используя объекты из одного уровня.

Синтаксис

ОБЪЕКТ.DrilldownReset (Level)

Часть	Описание
ОБЪЕКТ	Объект Layer который соотносится со сборочным слоем.
Level	Строковая величина, указывающая, какой уровень должен появиться во сборочном слое. Если строковая величина пустая, слой будет очищен (все объекты будут удалены из сборочного слоя).

Примечание

Этот метод так переустанавливает сборочный слой, что будут показаны объекты с одного уровня. Вы можете использовать этот метод если хотите создать кнопку ”Reset” , которая переведет сборочный слой в его начальное состояние.

Метод Layer.FeatureIDFromFeatureName (Объект Layer)

Задается имя, а возвращается ID объекта с этим именем (если таких имен несколько, возвращает ID первого объекта с указанным именем). Имя определенное в текущее время будет иметь значение из первой колонки в таблицы Mapinfo. Если нет объектов с указанным именем, то возникает исключительная ситуация.

Синтаксис

[int=]ОБЪЕКТ.FeatureIDFromFeatureName (Name)

Часть	Описание
ОБЪЕКТ	Это объект Layer.
Name	Строковая величина, имя объекта, чье ID Вы ищите.

Метод Layer.FeatureKeyFromFeatureName (Объект Layer)

Назначение

Этот метод возвращает FeatureKey из FeatureName в объекте Layer.

Синтаксис

[BSTR=]OBJECT.FeatureKeyFromFeatureName (KeyValue)

Часть	Описание
ОБЪЕКТ	Это объект Layer.
KeyValue	Имя объекта. Имя это значение KeyField.

Смотрите также

Свойство Layer.KeyField

Свойство Layer.FileSpec (объект Layer)

Назначение

Это свойство содержит полное описание того места, где физически хранится слой. Это строковая величина, открытая только для чтения.

Свойство Layer.Find (объект Layer)

Назначение

Определяет объект Find для слоя.

Метод **Layer.GetDrilldownFeaturesByID** (объект Layer)

Назначение

Возвращает объект Feature по его индексу ID.

Синтаксис

ОБЪЕКТ.GetDrilldownFeaturesByID(strLevel, FeatureID)

Часть Описание

ОБЪЕКТ Объект Layer.

strLevel Имя уровня во сборочном слое.

FeatureID Строковая величина или массив строковых величин, определяющих какие объекты запрашиваются..

Метод **Layer.GetFeatureByID** (объект Layer)

Назначение

Возвращает объект Feature, по заданному индексу ID.

Синтаксис

ОБЪЕКТ.GetFeatureByID(FeatureKey)

Часть Описание

ОБЪЕКТ Объект Layer.

FeatureKey Строковая величина, определяющая объект. Это значение, возвращаемое свойством Feature.FeatureKey. Это замена свойству Feature.FeatureID (которое до сих пор работает, но рекомендуется использовать новое свойство FeatureKey).

Метод **Layer.GetFeatureByKey** (Объект Layer)

Назначение

Этот метод возвращает объект Feature с заданным ключом (Key) в объекте Layer.

Синтаксис

[Feature=]ОБЪЕКТ.GetFeatureByKey (FeatureKey)

Части Описание

ОБЪЕКТ Это объект Layer.

FeatureKey Строковая величина, определяющая объект. Это величина, возвращаемая свойством Feature.FeatureKey.

Смотрите также

Свойство Feature.FeatureKey

Метод Layer.Invalidate (объект Layer)**Назначение**

Вызывает перерисовку в прямоугольнике, заданном в координатах экрана.

Синтаксис

Layer.Invalidate InvalidRect

Смотрите также

Метод Layer.AddFeature

Метод Layer.AllFeatures

Метод Layer.DeleteFeature

Метод Layer.NoFeatures

Метод Layer.SearchAtPoint

Метод Layer.SearchWithinDistance

Метод Layer.SearchWithinFeature

Метод Layer.SearchWithinRectangle

Метод Layer.UpdateFeature

Свойство `Layer.KeyField` (объект `Layer`)

Назначение

Это строковое свойство указывает имя колонки (поля) в слое таблицы Mapinfo, которая будет установлена или возвращена свойством `KeyValue` объекта карты. По умолчанию берется первая колонка таблицы слоя. Это свойство не применяется к растрам или слоям, нарисованным пользователем. Будет выдана ошибка, если Вы попытаетесь установить `KeyField` для неправильного имени поля.

Метод `Layer.LabelAtPoint` (объект `Layer`)

Назначение

Подписывает отдельный точечный объект.

Синтаксис

ОБЪЕКТ.`LabelAtPoint` (x, y)

Часть	Описание
ОБЪЕКТ	Это объект <code>Layer</code> .
x	X координата точки (двойная точность).
y	Y координата точки (двойная точность).

Пример на C++

```
// Layer.LabelAtPoint Method

void CSampleProjectView::OnToolUsed(short ToolNum, double X1, double Y1,
double X2, double Y2, double Distance, BOOL Shift, BOOL Ctrl1, BOOL*
EnableDefault) {

    if (ToolNum==CUSTOM_LABEL_TOOL_ID) {

        CMapXPoint pt;

        if (!pt.CreateDispatch(pt.GetClsid())) {

            TRACE0("Failed to Create Point object");
```

```

        return;
    }
    try {
        pt.Set(X1,Y1);
        // Label any item clicked on in the topmost layer
        m_Map.GetLayers().Item(1).LabelAtPoint(pt);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
}
}

```

Пример на Visual Basic

```

Private Sub Map1_ToolUsed(ByVal ToolNum As Integer, ByVal X1 _
As Double, ByVal Y1 As Double, ByVal X2 As Double, ByVal Y2 _
As Double, ByVal Distance As Double, ByVal Shift As Boolean, _
ByVal Ctrl As Boolean, EnableDefault As Boolean)

    If ToolNum = CUSTOM_LABEL_TOOL_ID Then

        Dim pt As New Point

        pt.Set X1,Y1

        ' Label any item clicked on in the topmost layer
        Map1.Layers(1).LabelAtPoint pt

    End If

```

End Sub

Свойство Layer.LabelProperties (объект Layer)

Назначение

Свойства подписи для слоя.

Смотрите также

Объект LabelProperties

Свойство Layer.Name (объект Layer)

Назначение

Это свойство содержит имя слоя. Имя дается слою при его добавлении.

Метод Layer.NoFeatures (объект Layer)

Назначение

Возвращает пустые объекты для слоя.

Синтаксис

ОБЪЕКТ.NoFeatures()

Смотрите также

Метод Layer.AddFeature

Метод Layer.AllFeatures

Метод Layer.DeleteFeature

Метод Layer.Invalidate

Метод Layer.SearchAtPoint

Метод Layer.SearchWithinDistance

Метод Layer.SearchWithinFeature

Метод Layer.SearchWithinRectangle

Метод `Layer.UpdateFeature`

Свойство `Layer.OverrideStyle` (объект `Layer`)

Назначение

Определяет, нужно ли менять стиль объекта, используя свойство `Style`. Это логическая величина и по умолчанию равна `False`.

Смотрите также

Свойство `Layer.Style`

Свойство `Layer.PredominantFeatureType` (объект `Layer`)

Назначение

Свойство открытое только для чтения, возвращающее целое значение, представляющее одну из констант `FeatureTypeConstants`. Оно используется для определения того, какого типа объекты преобладают на слое.

Примечание

Если на слое одинаковое количество разных объектов (например, 2 символа и 2 полигона) то `predominantFetureType` использует следующие приоритеты определения преобладающих объектов.

- `symbol`
- `line`
- `region`
- `misc`

Метод `Layer.Refresh` (Объект `Layer`)

Назначение

Этот метод перемещает кэш со слоя. Это полезно для серверных слоев, для которых включено кэширование.

Синтаксис

ОБЪЕКТ.Refresh

Смотрите также

Объект LayerInfo

Метод Layer.SearchAtPoint (объект Layer)

Назначение

Возвращает объект коллекции Features.

- Полигоны - если точка внутри полигона, полигон находится в результирующей коллекции.
- Линии и символы - Ближайшие символ(ы) или линия (/и) (может быть увеличено их количество, если несколько точек имеют одинаковые координаты) в пределах 3 пикселей вокруг данной точки. Такие линии и символы добавляются к результирующей коллекции.

Синтаксис

ОБЪЕКТ.SearchAtPoint (Point)

Часть	Описание
x	X координата точки на карте.
y	Y координата точки на карте.

Смотрите также

- Метод Layer.AddFeature
- Метод Layer.AllFeatures
- Метод Layer.DeleteFeature
- Метод Layer.Invalidate
- Метод Layer.NoFeatures
- Метод Layer.SearchWithinDistance
- Метод Layer.SearchWithinFeature
- Метод Layer.SearchWithinRectangle
- Метод Layer.UpdateFeature

Пример

```

Private Sub Map1_MouseDown(Button As Integer, Shift As _
Integer, X As Single, Y As Single)

Dim Lon As Double

Dim lat As Double

Dim fs As Features

Dim pnt As New Point

If Map1.CurrentTool = miArrowTool Then

    Map1.ConvertCoord X, Y, Lon, lat, miScreenToMap

    pnt.Set Lon, lat

    Set fs = Map1.Layers("US Top 20 Cities").SearchAtPoint(pnt)

    If fs.Count > 0 Then

        MsgBox fs.Count & ": " & fs(1).Name

    Else

        MsgBox "Nothing found"

    End If

End If

End Sub

```

Метод Layer.SearchWithinDistance (объект Layer)**Назначение**

Ищет объекты карты в пределах определенного расстояния и возвращает результаты поиска в виде коллекции Features.

Синтаксис

OBJECT.SearchWithinDistance(Source, double Distance, short Units, short SearchType)

Часть **Описание**

OBJECT	Объект Layer.
Source	Объект Point или объект Feature, от которого производится поиск.
Distance	Расстояние поиска, в единицах карты, определенных параметром Units.
Units	Значение MapUnitConstants, такое как miUnitMile (0), указывающее единицы измерения для аргумента Distance.
SearchType	Значение SearchTypeConstants; определяет критерии поиска, используемые для определения, находится ли объект в пределах зоны поиска.

Примечание

Этот метод создает буферную зону вокруг объекта-источника, затем ищет внутри полигона буфера. После осуществления поиска, буфер отбрасывается; если надо сохранить буфер, используйте метод BufferFeatures.

Разрешение буфера (количество нодов) определяется свойством Map.DefaultConversionResolution.

Пример на Visual Basic

```
Private Sub SelectCustomers(state As String,pt As Point, _
distance As Double,units As Integer)

    ' Given a layer named "Customers Layer", find all customers
    ' in (state) that are within (distance) (units) from (pt)

    Dim custInState As Features
    Dim custInDistance As Features
    Dim customersFound As Features
    Dim stateFeature As FindFeature
    Dim custLayer As Layer

    Set custLayer = Map1.Layers("Customers Layer")

    ' find the state
```

```

Set stateFeature = Map1.Layers("USA").Find.Search(state)
If stateFeature.FindRC mod 10 <> 1 Then ' No exact match found
    MsgBox "Couldn't find state"
Else
    ' find all the customers within the distance
    Set custInDistance = custLayer.SearchWithinDistance _
        (pt,distance,units,miSearchTypeCentroidWithin)

    ' find all the customers in the state
    Set custInState = custLayer.SearchWithinFeature(
        stateFeature, miSearchTypeCentroidWitih)

    ' keep only the customers within the distance and
    ' within the state
    Set customersFound = custInDistance.Clone
    customersFound.Common custInState

    ' select these customers
    custLayer.Selection.Replace customersFound
End If
End Sub

```

Пример на C++

```

// Features.Clone Method
// Features.Common Method

// Layer.SearchWithinDistance Method
// Layer.SearchWithinFeature Method

```

```

void CSampleProjectView::SelectCustomers(CString& state,CMapXPoint&
pt,double distance,short units) {

    // Given a layer named "Customers Layer", find all
    // customers in (state) that are within (distance) (units)
    // from (pt)

    CMapXFeatures custInState;
    CMapXFeatures custInDistance;
    CMapXFeatures customersFound;
    CMapXFindFeature stateFeature;
    CMapXLayer custLayer;

    try {

        custLayer = m_Map.GetLayers().Item("Customers Layer").

        // find the state

        stateFeature = m_Map.GetLayers().Item("USA").GetFind().
        Search(state);

        if(stateFeature.GetFindRC() % 10 != 1) {

            // No exact match found

            AfxMessageBox("Couldn't find state");

            return;

        }

        // find all the customers within the distance

        custInDistance = custLayer.SearchWithinDistance(pt,
        distance,units,miSearchTypeCentroidWithin);
    }

```

```

        // find all the customers in the state
        custInState = custLayer.SearchWithinFeature(stateFeature,
            miSearchTypeCentroidWithin);

        // keep only the customers within the distance and
        // within the state
        customersFound = custInDistance.Clone();
        customersFound.Common(custInState);

        // select these customers
        custLayer.GetSelection().Replace(customersFound);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Смотрите также

Метод Layer.AddFeature

Метод Layer.UpdateFeature

Метод Layer.DeleteFeature

Метод Layer.Invalidate

Метод Layer.SearchWithinFeature

Метод Layer.SearchWithinRectangle

Метод Layer.SearchAtPoint

Метод Layer.AllFeatures

Метод Layer.NoFeatures

Метод Layer.SearchWithinFeature (Объект Layer)

Назначение

Возвращает коллекцию Features. SearchType имеет значения miSearchTypeCentroidWithin, miSearchTypePartiallyWithin, miSearchTypeEntirelyWithin

Синтаксис

ОБЪЕКТ.SearchWithinFeature (Feature, SearchType)

Часть	Описание
ОБЪЕКТ	Объект Layer.
Feature	Объект Feature используемый как основа поиска.
SearchType	SearchType это или miSearchTypeCentroidWithin или miSearchTypePartiallyWithin или miSearchTypeEntirelyWithin.

Смотрите также

Метод Layer.AddFeature

Метод Layer.AllFeatures

Метод Layer.DeleteFeature

Метод Layer.Invaliddate

Метод Layer.NoFeatures

Метод Layer.SearchAtPoint

Метод Layer.SearchWithinDistance

Метод Layer.SearchWithinRectangle

Метод Layer.UpdateFeature

Метод **Layer.SearchWithinRectangle** (Объект **Layer**)

Назначение

Возвращает коллекцию Features объектов, попадающих в указанный прямоугольник.

Синтаксис

OBJECT.SearchWithinRectangle (Rectangle, SearchType)

Часть	Описание
OBJECT	Объект Layer.
Rectangle	Объект Feature, используемый как основа поиска.
SearchType	SearchType это miSearchTypeCentroidWithin, miSearchTypePartiallyWithin, miSearchTypeEntirelyWithin.

Смотрите также

- Метод Layer.AddFeature
- Метод Layer.AllFeatures
- Метод Layer.DeleteFeature
- Метод Layer.Invalidate
- Метод Layer.NoFeatures
- Метод Layer.SearchAtPoint
- Метод Layer.SearchWithinDistance
- Метод Layer.SearchWithinFeature
- Метод Layer.UpdateFeature

Свойство **Layer.Selectable** (объект **Layer**)

Назначение

Определяет, является ли слой доступным для выборки с помощью инструментов выборки. Это свойство не действует на автоматический процесс выборки в коллекции выборки. Это логическая величина, по умолчанию равная True.

Свойство **Layer.Selection** (объект **Layer**)

Назначение

Коллекция выбранных со слоя объектов.

Смотрите также

Коллекция Selection

Свойство **Layer.ShowNodes** (Объект **Layer**)

Назначение

Определяет, нужно ли отображать узлы объектов в слое (маленькие рамки вокруг узлов). По умолчанию свойство равно False.

Свойство **Layer.ShowCentroids** (Объект **Layer**)

Назначение

Определяет, нужно ли отображать направление центроиды объектов (только полигонов). По умолчанию равно false.

Свойство **Layer.ShowLineDirection** (Объект **Layer**)

Назначение

Определяет, нужно ли отображать направление рисования линий и полилиний. По умолчанию равно false.

Свойство Layer.Style (объект Layer)

Назначение

Объект стиля, используемого для того чтобы все объекты появились оформленными в указаном стиле. Объект Style вызывается и заполняется новыми необходимыми значениями. Этот стиль используется, если свойство Override (единообразно) установлено на True.

Смотрите также

Свойство Layer.OverrideStyle

Свойство Layer.Type (Объект Layer)

Назначение

Новый тип свойства объекта Layer возможен, если Вы укажете, что слой имеет один из констант типов слоев:

LayerTypeConstants

```
miLayerTypeNormal = 0
miLayerTypeRaster = 2
miLayerTypeSeamless = 4
miLayerTypeUnknown = 5
miLayerTypeUserDraw = 6
```

Пример на C++

```
// Layer.Type Property
void CSampleProjectView::CountLayers() {
    long lCount, normCount=0, rasterCount=0, seamlessCount=0;
    try {
        lCount = m_Map.GetLayers().GetCount();
        for(long i=1; i<=lCount; i++) {
            switch(m_Map.GetLayers().Item(i).GetType()) {
```

```

        case miLayerTypeNormal:
            normCount++;
            break;
        case miLayerTypeRaster:
            rasterCount++;
            break;
        case miLayerTypeSeamless:
            seamlessCount++;
            break;
        // case miLayerTypeUnknown:
        // case miLayerTypeUserDraw:
        // case miLayerTypeDrilldown:
    };
}

} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}

msg.Format("There are %ld normal layers, %ld raster layers, and %ld
seamless layers.",normCount,rasterCount,seamlessCount);

AfxMessageBox(msg);
}

```

Пример на Visual Basic

```

Private Sub CountLayers()
    Dim normCount As Integer, rasterCount As Integer, _
    seamlessCount As Integer
    For Each Lyr In Map1.Layers
        Select Case Lyr.Type
            Case miLayerTypeNormal
                normCount = normCount + 1
            Case miLayerTypeRaster
                rasterCount = rasterCount + 1
            Case miLayerTypeSeamless
                seamlessCount = seamlessCount + 1
            'Case miLayerTypeUnknown
            'Case miLayerTypeUserDraw
            'Case miLayerTypeDrilldown
        End Select
    Next
    MsgBox "There are " & normCount & " normal layers, " _
    & rasterCount & " raster layers, and " & seamlessCount _
    & " seamless layers."
End Sub

```

Метод Layer.UpdateFeature (объект Layer)

Назначение

Обновляет указанный объект на слое, который до этого имел другие свойства. Этот метод полезен для редактирования объекта.

Синтаксис

OBJECT.UpdateFeature (FeatureKey, [Source], [RowValues])

Часть Описание

OBJECT	Объект Layer.
FeatureKey	Обновляемый объект Feature. Теперь применяется вместо параметра FeatureID. FeatureID до сих пор работает, но рекомендуется использовать новый параметр FeatureKey.
Source	Объект Feature из которого извлекаются свойства. По умолчанию это Target.
RowValues	Это коллекция RowValues, содержащая новые значения в столбцах данных.

Смотрите также

Метод Layer.AddFeature

Метод Layer.AllFeatures

Метод Layer.DeleteFeature

Метод Layer.Invalidate

Метод Layer.NoFeatures

Метод Layer.SearchAtPoint

Метод Layer.SearchWithinDistance

Метод Layer.SearchWithinFeature

Метод Layer.SearchWithinRectangle

Свойство Layer.Visible (Объект Layer)

Назначение

Определяет, является ли слой видимым. Это логическая величина, по умолчанию равная True.

Свойство `Layer.ZoomLayer` (Объект `Layer`)

Назначение

Определяет, есть ли масштабный фактор у слоя. Масштабный фактор определяет те пределы, в которых слой является видимым. Если масштабный фактор (`Zoom Layering`) включен, то значения пределов хранятся в `ZoomMax` и `ZoomMin`. Это логическая величина, по умолчанию равная `False`.

Смотрите также

Свойство `Layer.ZoomMin`

Свойство `Layer.ZoomMax`

Свойство `Layer.ZoomMax` (Объект `Layer`)

Назначение

Если установлен масштабный фактор (`ZoomLayering`) (Свойство `Layer.ZoomLayer`), то это свойство определяет максимальный размер в единицах длины окна карты, при котором слой видим. Это величина двойной точности, в единицах карты (`Map.MapUnit`).

Смотрите также

Свойство `Layer.ZoomLayer`

Свойство `Map.MapUnit`

Свойство `Layer.ZoomMin` (Объект `Layer`)

Назначение

Если установлен масштабный фактор (`ZoomLayering`) (Свойство `Layer.ZoomLayer`), то это свойство определяет минимальный размер в единицах длины окна карты, при котором слой видим. Это величина двойной точности, в единицах карты (`Map.MapUnit`).

Смотрите также

Свойство `Layer.ZoomLayer`

Свойство `Map.MapUnit`

Метод `Layers.Add` (Коллекция `Layers`)

Назначение

Создает объект типа слой для указанного файла, добавляет его к коллекции и отображает его на карте.

Синтаксис

`ОБЪЕКТ.Add ([LayerInfo], [Position])`

Часть Описание

ОБЪЕКТ Это объект `Layers`.

LayerInfo Переменная: Может быть или объектом `LayerInfo` или путем (`Pathname`) к таблице `MapInfo` (.ТАВ) добавляемой к слою.

Position Переменная: Его начальная позиция в списке слоев. Если не указано значение или указан 0, используется алгоритм автоматического позиционирования слоя.

Примечание

Слой автоматически позиционируется относительно других слоев карты. Например, слой с точками помещается выше слоя с полигонами.

Если строковая переменная обозначает параметр `LayerInfo`, то он интерпретируется следующим образом:

- расширение `.tab` воспринимается как файл таблицы `.tab`.
- любое другое расширение воспринимается как саморегистрирующийся растровый файл.
- отсутствие расширения воспринимается как имя пользователя в геословаре (`geodictionary user name`).

Пример

```
Map1.Layers.Add dlgFileOpen.filename, 2
```

Метод `Layers.AddGeosetLayers` (Коллекция `Layers`)

Назначение

Добавляет указанный рабочий набор `GeoSet` к текущей карте. Создает объекты `Layer` для каждого слоя, указанного в рабочем наборе `GeoSet` которого еще нет в коллекции `Layers`.

Синтаксис

`ОБЪЕКТ.AddGeosetLayers (GeosetName)`

Часть

`ОБЪЕКТ`

`GeosetName`

Описание

Это объект `Layers`.

“Дружественное” имя рабочего набора (‘friendly’ name) или полное описание местоположения файла.

Пример

```
Map.Layers.AddGeosetLayers "United States"
```

Смотрите также

Свойство `Map.Geoset`

Метод `Layers.AddServerLayer` (Коллекция `Layers`)

Назначение

Добавляет удаленные пространственные данные к карте (то есть слой объектов возвращается из удаленной базы данных типа `Oracle` или `Informix`).

Синтаксис

`ОБЪЕКТ.AddServerLayer (Name, ConnectString, Query, Cache, [Position], [Options])`

Часть

`ОБЪЕКТ`

`Name`

`ConnectString`

Описание

Объект коллекции `Layers`.

Строковая величина, определяющая имя создаваемого слоя (типа “Highways”).

Строковая величина - связь `ODBC`, указывающая параметры соединения

Query	такие как имя хоста, пароль, и др. Если строковая величина не дает всей информации, источник данных ODBC показывает диалог соединения Строковая величина, которая указывает GISSQL/Server-specific SQL запрос.
Cache	
Position	Дополнительная: Позиция в списке слоев. Для перемещения слоя на самый верх, укажите 1 (один).
Options	Дополнительная: Принимает поразрядно значения OR для LayerSrvLayerOptions, которое определяет поведение слоя сервера. По умолчанию принимает значения (miLayerCacheOn или miMBRSearchOn). Смысл в том, что в VB, оператор OR используется и для логического и для пошагового OR, а в C++ есть для этих случаев разные операторы. Тот кто встраивает MapX в приложение C++ должен знать как корректно использовать OR. Это позволяет контролировать некоторые элементы поведения слоев сервера (находящиеся в кэше и опции MBRsearch). Это не логические значения. Здесь берутся установки флагов (bitmask) как описано ниже. Используйте нумерованные значения и просто OR чтобы совместно определить значение SrvLayerOptions перед вызовом AddServerLayer

Примечание

Метод AddServerLayer включен для совместимости сверху вниз. Можно для этих целей также просто метод Layers.Add. Объект LayerInfo посланный в метод Add позволяет Вам указывать какой слой является серверным и контролировать кэш и MBR search.

Следующая таблица суммирует эффект параметра Options. Константы могут быть пошаговыми и использоваться вместе для комбинации настроек.

miLayerCacheOn	Прочитанные атрибуты и объекты будут храниться в памяти. Это дубирует интерактивность карты (например, масштабирование). Таким образом, самое позднее обновление данных не будет показано пока Layer.Refresh вызван.
miLayerCacheOff	Все данные будут выбираться из базы данных каждый раз при прорисовке слоя. Это менее эффективно но лучше для данных, привязанных ко времени (date data).
miLayerMBRSearchOn	Будет осуществлять MBRSearch поиск предикатов для добавления к n пространственному запросу, когда собираются данные.

`miLayerMBRSearch` Выключает предикат `MBRSearch` который присоединяется к запросу `ff` когда карта нарисована. Это полезно для запросов которые уже содержат пространственный предикат и когда добавочный пространственный предикат не нужен.

Смотрите также

`Layers.Add`

Объект `LayerInfo`

Метод `Layers.AddUserDrawLayer` (Коллекция `Layers`)

Назначение

Добавляет слой, в котором связанное приложение- контейнер (container application) может рисовать. Используется вместе с событием `DrawUserLayer`, которое запускается, когда на слое надо что-то нарисовать. Возвращает вновь созданный объект `Layer`.

Синтаксис

`ОБЪЕКТ.AddUserDrawLayer (Name, Position)`

Части Описание

`ОБЪЕКТ` Объект `Layers`.

`Name` Имя создаваемого слоя

`Position` Место в списке слоев, куда будет добавлен новый слой.

Свойство `Layers.AnimationLayer` (Коллекция `Layers`)

Слой анимации полезен, если есть периодическое обновление, например для приложений, работающих в режиме реального времени. Например, Ваше приложение работает с данными от GPS (Global Positioning Satellite) приемников, сообщающих текущие координаты перемещающихся объектов. Поскольку большая часть объектов карты не изменяется, для ускорения прорисовки положения перемещающихся объектов используется анимационный слой.

Первоначально `AnimationLayer` установлен на ноль. Вы можете назначить свойству объекта `Layer` такое свойство, которое сделает этот слой анимационным (это может быть обычный слой

или слой, нарисованный пользователем). Когда слою присваивается свойство `AnimationLayer`, он прорисовывается выше всех остальных слоев, включая слой аннотаций и выборок. Слой пока находится на первоначальном месте в списке слоев. Плавающие объекты типа легенды также будут прорисовываться выше анимационного слоя, хотя легенда и не будет перерисовываться аналогично слою анимации. Если как анимационный слой используется обычный слой, выборка и подписывание будут работать.

Что бы выключить анимационный слой, присвойте свойству значение ноль:

```
Set Map.Layers.AnimationLayer = nothing
```

Это действие вернет слой в обычное состояние, располагающийся в своем первоначальном месте в списке слоев.

Пример

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

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

Свойство `Layers.Bounds` (Коллекция `Layers`)

Назначение

Возвращает объект `Rectangle` представляющий все объекты попавшие в прямоугольник, кроме слоев, нарисованных пользователем (кроме слоев `UserDraw`).

Пример на C++

```
// Layers.Bounds Property

void CSampleProjectView::ViewAllLayers() {

    // Set the boundaries of the map view to be the boundaries of all of
    the layers

    try {
```

```

        m_Map.SetBounds(m_Map.GetLayers().GetBounds());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub ViewAllLayers_Click()
    ' Set the window boundaries of the map to be the boundaries
    ' of all of the layers
    Map1.Bounds = Map1.Layers.Bounds
End Sub

```

Примечание

Это свойство полезно, если Вы хотите подобрать масштаб карты так, чтобы на дисплее поместились все объекты со всех слоев.

Смотрите также

Свойство Layer.Bounds

Метод Layers.ClearSelection (Коллекция Layers)

Назначение

Отменяет выборку объектов на всех слоях коллекции Layers.

Смотрите также

Selection.ClearSelection

Свойство Layers.Count (Коллекция Layers)

Назначение

Содержит количество объектов Layer в коллекции. Это целая величина, открытая только для чтения.

Метод Layers.CreateLayer (Коллекция Layers)

Назначение

Метод CreateLayer позволяет создать временный или постоянный слой таблицы MapInfo. Созданная таблица имеет колонку для имени объекта, которая используется при подписывании и связывании данных. Когда Вы добавляется или обновляете объект, свойство Name помещается в колонку с именами объектов.

Синтаксис

OBJECT.CreateLayer (Name , [FileSpec] , [Position] , [KeyLength] , [CoordSys])

Часть	Описание
OBJECT	Объект коллекции Layers.
Name	Имя слоя на который ссылка (имя пользователя слоя).
FileSpec	Переменная: Путь к месту где будет создан слой. Имя файла должно включать расширение .tab. Другие файлы с расширениями (.map, .dat, и др.) создаются в том же каталоге, где и .tab. Если имя файла не задано, то создается временный слой, который будет удален когда карта или осх стираются.
Position	Переменная: Начальное положение в списке слоев. Если оно пропущено, положение определится специальным алгоритмом, размещающим слои по типам объектов.
KeyLength	Переменная: Длина колонки, добавленной к таблице, которая содержит имена объектов. Если она пропущена, по умолчанию

значение равно 32.

CoordSys Переменная: Объект CoordSys который определяет систему координат, в которой сохраняется новый слой. Необязательная величина; если пропущена, то используется свойство Map.NumericCoordSys.

Примечание

Метод возвращает объект Layer, который добавлен к коллекции. Новый слой создается с ключевой колонкой, называемой “GEONAME”.

Пример на Visual Basic

```
Dim lyr As MapXLib.Layer
' create temporary layer to be used as animation layer
' put in first position in layer list and use default keylength
Set lyr = Map1.Layers.CreateLayer("tempAnimate", ,1)
Set Map1.Layers.AnimationLayer = lyr
```

Пример на C++

```
void CSampleProjectView::CreateAnimationLayer() {
    CMapXLayer lyr;
    try {
        // Create temporary layer to be used as animation layer
        // Put in first position in layer list and use default keylength
        lyr=m_Map.GetLayers().CreateLayer("tempAnimate",NULL,1);
        m_Map.GetLayers().SetAnimationLayer(lyr);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
```

```

        e->ReportError();
        e->Delete();
    }
}

```

Свойство `Layers.InsertionLayer` (Объект `Layers`)

Назначение

Это свойство определяет в какой слой будут вставлены новые объекты, созданные инструментально. По началу `InsertionLayer` установлен на `NULL`. Вы можете присвоить объект `Layer` к свойству, чтобы сделать этот слой “вставляемым” слоем “`InsertionLayer`” (смотрите пример программы).

Пример на `Visual Basic`

```

'Make layer USA the insertion layer
dim lyrInsertion as MapXlib.Layer
set lyrInsertion = Map1.Layers("USA")

lyrInsertionLayer.Editable = True
Set Map1.Layers.InsertionLayer = lyrInsertion

```

Примечание

Вы не можете сделать “вставляемым” сквозной, растровый, спитый или создаваемый пользователем слой.

Редактируемые свойства слоя должны быть установлены на `true` перед тем как он станет “вставляемым”.

Только один слой может быть “вставляемым”.

Инструменты создания объектов `AddPoint`, `AddLine`, `AddRegion` и `AddPolyLine` будут создавать объекты на том слое, который объявлен “вставляемым”.

Вы не можете установить Map.CurrentTool на любой из инструментов создания объектов, пока Вы не укажете “вставляемый” слой.

Свойство Layers.Item (Коллекция Layers)

Назначение

Получает указанный объект Layer из коллекции. Объект Layer определяется по индексу (целое, от 1) или по имени объекта.

Метод Layers.LayersDlg (Коллекция Layers)

Назначение

Показывает диалог, в котором пользователь может добавлять слои, менять их порядок, и настраивать свойства слоев. Если пользователь нажимает ОК, то изменения, сделанные в диалоге, немедленно отображаются на карте.

Синтаксис

[Boolean=]OBJECT.LayersDlg ([HelpFile], [HelpID])

Часть	Описание
OBJECT	Коллекция Layers.
HelpFile	Имя файла справки (Windows Help). (дополнительная)
HelpID	Номер ID раздела файла справки. (дополнительная)

Примечание

Если Вы указываете и HelpFile и HelpID, то диалог включает в себя кнопку Help; если пользователь нажимает эту кнопку, MapX показывает соответствующий раздел Help .

Пример

Следующий оператор появляется в окне диалога Layer Control

```
Map1.Layers.LayersDlg
```

Метод `Layers.Move` (Коллекция `Layers`)

Назначение

Передвигает слой в коллекции `Layer` изменяя порядок, в котором прорисовываются слои.

Синтаксис

`ОБЪЕКТ.Move (From, To)`

Части	Описание
<code>ОБЪЕКТ</code>	Объект <code>Layers</code> .
<code>From</code>	Индекс слоя, который перемещается. Самый верхний слой это 1.
<code>To</code>	Новое положение слоя. Например, если перемещаете на второе место, то это 2..

Пример на C++

```
// Layers.Move Method
void CSampleProjectView::Rearrange() {
    // Move the lowest layer to the top
    try {
        m_Map.GetLayers().Move(m_Map.GetLayers().GetCount(), 1);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```


Пример на Visual Basic

```
Private Sub Rearrange()  
    ' Move the lowest layer to the top  
    Map1.Layers.Move Map1.Layers.Count,1  
End Sub
```

Примечание

Порядок слоев в коллекции Layer , в котором они прорисовываются. Когда слои добавляются используя Layers.Add или Layers.AddGeosetLayers, то слои вставляются по некоторому правилу в коллекцию слоев. Например, слои с точками помещаются выше слоев с площадными объектами, слои с площадными объектами выше растровых слоев и т.д.

Смотрите также

Коллекция Layers

Метод Layers. Remove (Коллекция Layers)

Назначение

Удаляет объект Layer из коллекции. Имеет эффект удаления слоя из карты.

Внимание: Если Вы удаляете объект, то индексы коллекции перенумеровываются, что бы заполнить пустое место.

Синтаксис

ОБЪЕКТ.Remove (Index)

Часть Описание

ОБЪЕКТ Объект Layers.

Index Или целочисленный индекс(начиная с 1) или имя удаляемого слоя.

Метод `Layers.RemoveAll` (Коллекция `Layers`)

Назначение

Удаляет все объекты `Layer` из коллекции.

Синтаксис

`Layers.RemoveAll`

Пример на Visual Basic

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

Пример на C++

```
void CSampleProjectView::RemoveAllLayers() {
    try {
        m_Map.GetLayers().RemoveAll();
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

}

Объект LayerInfo

Объект LayerInfo хранит параметры, которые будут переданы в метод Layers.Add и определяет типы из констант LayerInfoType.

Таблица описывает параметры, которые надо использовать для каждого типа слоев.

Описание LayerInfoType	Параметр	Требуется	Тип
miLayerInfoTypeTab	FileSpec	Yes	Строковая: Указание на файл .tab который будет открыт
	Name	No	Строковая: Имя добавляемого слоя
miLayerInfoTypeUserDraw	Name	Yes	Строковая: Имя слоя, создаваемого пользователем
miLayerInfoTypeRaster	FileSpec	Yes	Строковая: указание на саморегистрирующийся растр, который будет открыт
	Name	No	Строковая: Имя добавляемого слоя
miLayerInfoTypeShape	FileSpec	Yes	Строковая: Указание на .shp файл
	Name	No	Строковая: Имя добавляемого слоя
	CoordSys	Yes	Объект CoordSys: Система координат в которой хранится в .shp файле.
	Style	No	Объект Style: Стилль используется для

			рендеринга объекта.	
miLayerInfoTypeServer	Name	Yes	Строковая: Имя добавляемого слоя	
	ConnectionString	Yes	Строковая: Строка соединения ODBC (смотри AddServerLayer)	
	Query	Yes	Строковая: Запрос SQL (смотрите AddServerLayer)	
	ToolKit	Yes	Строковая: Настройка инструментария ToolKit , или ODBC или ORAINET	
		Cache	No	Строковая: Или ON или OFF. По умолчанию OFF.
		MBRSearch	No	Строковая: Или ON или OFF. По умолчанию OFF.
miLayerInfoTypeGeodictUserNa me	Name	Yes	Строковая: Имя добавляемого слоя (как он хранится в Geodictionary).	

Для любого типа LayerInfoType отличного от miLayerInfoTypeUserDraw или miLayerInfoTypeRaster, поддерживаются следующие настройки:

Any	AutoCreateDataset	No	Числовое: если 1, то набор данных типа miDatasetLayer добавлен после добавления слоя. Если 0 (по умолчанию), то набор данных добавляться не будет.
Any	DatasetName	No	Строковая: Если параметр AutoCreateDataset равен 1, то это будет имя нового набора данных. Если строка не определена, используется то имя, которое по умолчанию.
Any	Cache	No	Строковая: Или ON или OFF, указывает, кэшируются ли данные, загружаемые из серверных слоев (server layers). По

умолчанию равно OFF.

Свойства объекта

- Свойство LayerInfo.Type (Объект LayerInfo)

Методы объекта

- Метод LayerInfo.AddParameter (Объект LayerInfo)

Пример

```
Dim LayerInfoObject As New LayerInfo
LayerInfoObject.Type = miLayerInfoTypeTab
LayerInfoObject.AddParameter "FileSpec", "c:\Program _
Files\MapInfo MapX 4.0\Maps\worldcap.tab"
LayerInfoObject.AddParameter "Name", "My New Layer"
Map1.Layers.Add LayerInfoObject
```

Смотрите также

Метод Layers.Add

Свойство LayerInfo.Type (Объект LayerInfo)

Назначение

Это свойство процедуры чтения/записи, чье значение это одна из констант LayerTypeConstants

Смотрите также

Метод LayerInfo.AddParameter

Метод Layers.Add

LayerInfoTypeConstants

Метод `LayerInfo.AddParameter` (Объект `LayerInfo`)

Назначение

Метод `Addparameter` добавляет установку параметров, требуемых для данного вызова метода `Layers.Add`. Необходимые параметры определяются по свойствам типов объекта `LayerInfo`.

Синтаксис

ОБЪЕКТ.`AddParameter` (name, value)

Части Описание

ОБЪЕКТ Объект `LayerInfo`.

name Имя параметра.

value Значение параметра.

Смотрите также

Метод `Layers.Add`

Объект `LayerInfo`

Объект `Legend`

Назначение

Каждый объект `Theme` имеет объект `Legend` (`Theme.Legend`). Объект `Legend` содержит свойства, определяющие настройки легенды. Каждый объект `ThemeCategory` (`RangeCategory`, `IndividualValueCategory` или `MultiVarCategory`) имеет вход в легенду, содержащуюся в объекте `LegendText`.

Свойства объекта

- Свойство `Legend.BodyTextStyle` (Объект `Legend`)
- Свойство `Legend.Compact` (Объект `Legend`)
- Свойство `Legend.CompactTitle` (Объект `Legend`)
- Свойство `Legend.CompactTitleStyle` (Объект `Legend`)
- Свойство `Legend.CurrencyFormat` (Объект `Legend`)

- Свойство Legend.Height (Объект Legend)
- Свойство Legend.Left (Объект Legend)
- Свойство Legend.LegendTexts property (Объект Legend)
- Свойство Legend.PaperHeight (Объект Legend)
- Свойство Legend.PaperWidth (Объект Legend)
- Свойство Legend.ShowCount (Объект Legend)
- Свойство Legend.ShowEmptyRanges (Объект Legend)
- Свойство Legend.SubTitle (Объект Legend)
- Свойство Legend.SubTitleStyle (Объект Legend)
- Свойство Legend.Title (Объект Legend)
- Свойство Legend.TitleStyle (Объект Legend)
- Свойство Legend.Top (Объект Legend)
- Свойство Legend.Visible (Объект Legend)
- Свойство Legend.Width (Объект Legend)

Методы Объекта

- Метод Legend.ExportLegend (Объект Legend)
- Метод Legend.LegendDlg (Объект Legend)
- Метод Legend.PrintLegend (Объект Legend)

Свойство Legend.BodyTextStyle (Объект Legend)

Назначение

Это объект Style, который определяет стиль текста для содержимого легенды. Это объект Style и он может быть установлен для существующего объекта стиля, или Вы можете установить свойства стиля индивидуально.

Смотрите также

Объект Style

Свойство `Legend.Compact` (Объект `Legend`)

Назначение

Определяет, является ли легенда компактной (1 строка) или полноразмерной. Это логическая величина, по умолчанию устанавливается `Map.PreferCompactLegends`.

Смотрите также

Свойство `Map.PreferCompactLegends`

Свойство `Legend.CompactTitle` (объект `Legend`)

Назначение

Содержит текст заголовка компактной легенды. Он появляется, когда легенда является компактной. Это строковая величина и по умолчанию заголовок создается при создании тематической карты.

Свойство `Legend.CompactTitleStyle` (объект `Legend`)

Назначение

Это объект `Style` который определяет стиль текста для компактного заголовка. Это объект `Style` и он может быть установлен для существующего объекта стиля, или Вы можете установить свойства стиля индивидуально.

Смотрите также

Объект `Style`

Свойство Legend.CurrencyFormat (объект Legend)

Назначение

Определяет, отображаются ли числовые значения легенды в текущем формате или в стандартном числовом формате. Это логическая величина, по умолчанию равная False.

Метод Legend.ExportLegend (Объект Legend)

Назначение

Экспортирует изображение легенды в файл или в буфер.

Синтаксис

ОБЪЕКТ.ExportLegend (Destination, Format)

Часть Описание

ОБЪЕКТ Объект Legend.

Destination Местоположение выходного файла, например, "C:\Temp\Legend.bmp". Если укажите "CLIPBOARD" вместо пути, изображение попадет в буфер.

Format Выходной формат. Берется значение константы ExportFormatConstants. Внимание: Если указан miFormatGIF (для генерации файлов GIF), пожалуйста, прочтите Licensing Requirements for Users файлов GIF. Если Вы посылаете в CLIPBOARD, формат должен быть metafile или bitmap.

Смотрите также

Map.ExportMap

Свойство Legend.Height (объект Legend)

Назначение

Содержит значение высоты легенды в единицах экрана. Это короткое целое, и свойство открыто только для чтения.

Свойство Legend.Left (объект Legend)

Назначение

Содержит позицию левого угла легенды в единицах экрана, относительно относительно верхнего левого угла карты. Это короткая целая величина.

Внимание: Вы можете перетащить легенду за пределы контроля MapX в форме, и не сможете вернуть ее обратно.

Возможно запрограммировать переустановку свойств Left и Top объекта, что бы он размещался в пределах карты. Это также применимо к заголовку карты и аннотациям.

Метод Legend.LegendDlg (объект Legend)

Назначение

Представляет диалог, в котором пользователь может настроить легенду (изменять параметры свойства объекта легенды через пользовательский интерфейс). Если пользователь нажимает 'ОК', изменения в диалоге немедленно применяются к карте.

Синтаксис

```
[Boolean=]OBJECT.LegendDlg([ HelpFile ][ HelpID ])
```

Часть Описание

OBJECT	Объект Legend.
HelpFile	HelpFile это необязательный параметр - путь к .hlp файл содержащему описание для этого диалога.
HelpID	HelpID это необязательный параметр, который ссылается на индекс, специфического раздела электронной справки, в .hlp файле.

Примечание

Если оба параметра отсутствуют, то кнопка Справки в диалоге не появится.

Возвращаемое значение LegendDlg это True если пользователь нажал ОК и False если пользователь нажал Cancel.

Пример на Visual Basic

```
'This example displays the Legend Dialog
'for a theme when its legend is double-clicked
Private Sub Map1_ThemeModifyRequested(ByVal Theme As Object)

    Theme.Legend.LegendDlg

End Sub
```

Пример на C++

```
void OnThemeModifyRequested(LPDISPATCH ThemeDispatch) {
    CMapXTheme theme;

    theme.AttachDispatch(ThemeDispatch, FALSE); // Get theme object
    try {
        theme.GetLegend().LegendDlg();
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Свойство `Legend.LegendTexts` (объект `Legend`)

Назначение

Коллекция объектов `LegendText` для легенды. Диапазоны (`Ranged`), Индивидуальные значения (`Individual Value`), Круговые диаграммы (`Pie`), Столбчатые (`Bar`) и Плотность точек (`Dot Density`) имеют по одному свойству `LegendText`. Размерные символы имеют значение три.

Смотрите также

Объект `LegendText` и коллекция `LegendTexts`

Свойство `Legend.PaperHeight` (объект `Legend`)

Назначение

Содержит высоту легенды в единицах `Map.PaperUnit`. Это короткое целое и свойство открыто только для чтения.

Смотрите также

Свойство `Map.PaperUnit`

Свойство `Legend.PaperWidth` (объект `Legend`)

Назначение

Содержит ширину легенды в единицах `Map.PaperUnit`. Это короткое целое и свойство открыто только для чтения.

Смотрите также

Свойство `Map.PaperUnit`

Метод Legend.PrintLegend (объект Legend)

Назначение

Печатает легенду в указанном прямоугольнике и в указанном устройстве.

Синтаксис

ОБЪЕКТ.PrintLegend (hDC, x, y, w, h)

Часть	Описание
ОБЪЕКТ	Объект Legend.
hDC	Устройство принтера.
x	Верхний левый угол X в единицах HIMETRIC.
y	Верхний левый угол Y в единицах HIMETRIC.
w	Ширина в единицах HIMETRIC.
h	Высота в единицах HIMETRIC.

Примечание

Текущая легенда прорисовывается так, чтобы заполнить прямоугольник. Лучший результат достигается, если выдержана пропорция высота/ширина.

Пример на Visual Basic

```
Private Sub btnPrintLegend_Click()
    Dim Legend as MapXLib.Legend

    ' there is no Printer.StartDoc method, it seems it is done
    ' implicitly when you use one of the printer.print methods
    ' so we need to print something before we print our legend
    ' to start the page

    Printer.CurrentX = 0
    Printer.CurrentY = 0

    Printer.Print " "

    Set Legend = Map1.Datasets(1).Themes(1).Legend
```

```

' Print legend same width and height as is shown on the map
' coords must be in himetric
' Assumes a screen device with 96 pixels per inch (most are)
' There are 2540 himetric units per inch
' Legend.Width and Height are returned in Screen units
'(pixels)
' so conversion from pixels to himetric is
' pixels/pixelsperinch * himetricperinch
Legend.PrintLegend Printer.hDC, 0, 0, Legend.Width / _
96 * 2540,          Legend.Height / 96 * 2540
' This example prints a legend in the upper left that
' is 4 by 3 inches
' 4 by 3 inches (2540 himetric units per inch)
'Map1.Datasets(1).Themes(1).Legend.PrintLegend Printer.hDC,
'0,          '0, 10160, 7620
' VB4 needed an explicit NewPage before the EndDoc,
' but VB5 doesn't
'Printer.NewPage ' Send new page.
Printer.EndDoc ' Printing is finished.

End Sub

```

Пример на C++

```

void CSampleProjectView::OnPrintLegend(CDC* pDC, CPrintInfo* pInfo) {
    try {
        CMapXLegend Legend =
m_Map.GetDatasets().Item(1).GetThemes().Item(1).GetLegend();

```

```

// Print legend the same width and height as it is shown on the map
m_Map.SetPaperUnit(miUnitMillimeters);

// PrintLegend takes coordinates in HIMETRIC (100ths of a
Millimeter)

double pw = Legend.GetPaperWidth() * 100.0;
double ph = Legend.GetPaperHeight() * 100.0;

Legend.PrintLegend((long)pDC->m_hDC, 0, 0, (long)pw, (long)ph);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Свойство Legend.ShowCount (объект Legend)

Назначение

Определяет, показываются или нет количество объектов в диапазонах карты. Это логическое значение, по умолчанию равно True.

Пример на Visual Basic

```

Private Sub btnLegendShowCount_Click()

    Dim lgnd As MapXLib.Legend

    Set lgnd = Map1.Datasets(1).Themes(1).Legend

    ' This makes the counts, next to the legend's ranges,

```

```

        ' invisible

lgnd.ShowCount = False

End Sub

```

Пример на C++

```

void CSampleProjectView::LegendShowCount() {
    try {
        CMapXLegend lgnd =
m_Map.GetDatasets().Item(1).GetThemes().Item(1).GetLegend();

        // This makes the counts, next to the legend's ranges, invisible
        lgnd.SetShowCount(FALSE);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Свойство Legend.ShowEmptyRanges (объект Legend)

Назначение

Свойство доступное и для чтения и для записи; определяет, есть ли пустые диапазоны (диапазоны, не содержащие объектов) в легенде. По умолчанию равно False.

Свойство Legend.SubTitle (объект Legend)

Назначение

Это строковая величина, доступная для процедуры чтения/записи, которая содержит текст подписи подзаголовка. Он изображается на экране при нормальном размере легенды. По умолчанию подзаголовков создается при создании тематической карты.

Свойство Legend.SubTitleStyle (объект Legend)

Назначение

Это объект Style который определяет стиль текста для подзаголовка, открыт для чтения и записи. Этот объект Style может быть установлен по умолчанию, или Вы можете установить свойства стиля индивидуально.

Смотрите также

Объект Style

Свойство Legend.Title (объект Legend)

Назначение

Содержит текст заголовка легенды. Он показывается, когда легенда имеет нормальные размеры. Это строковая величина. По умолчанию заголовок создается при создании тематической карты.

Свойство Legend.TitleStyle (объект Legend)

Назначение

Это объект Style который определяет стиль текста для заголовка. Этот объект Style может быть установлен по умолчанию, или Вы можете установить свойства стиля индивидуально.

Смотрите также

Объект Style

Свойство Legend.Top (объект Legend)

Назначение

Содержит позицию верхнего угла легенды в единицах экрана, относительно верхнего левого угла карты. Это строковая величина.

Внимание: Вы можете перетащить легенду за пределы контроля MapX в форме, и не сможете вернуть ее обратно.

Возможно запрограммировать переустановку свойств Left и Top объекта, что бы он размещался в пределах карты. Это также применимо к заголовку карты и аннотациям.

Свойство Legend.Visible (объект Legend)

Назначение

Определяет будет видима или нет легенда. Это логическая величина, по умолчанию равная True.

Свойство Legend.Width (объект Legend)

Назначение

Содержит ширину легенды в экранных единицах. Это короткое целое, свойство открыто только для чтения

Пример на C++

```
// Legend.Visible Property
// Legend.Top Property
// Legend.Left Property

void CSampleProjectView::LegendMove(CMapXLegend& posLegend) {
```

```

// move the legend to the top left corner of the screen
try {
    posLegend.SetVisible(TRUE);
    posLegend.SetLeft(0);
    posLegend.SetTop(0);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Пример на Visual Basic

```

Private Sub LegendMove(posLegend As Legend)
    ' move the legend to the top left corner of the screen
    posLegend.Visible = True
    posLegend.Left = 0
    posLegend.Top = 0
End Sub

```

Объект и коллекция LegendText

Назначение

Каждый объект Theme имеет коллекцию объектов LegendText называемых LegendTexts. Каждый объект (каждая строка под заголовком) легенды имеет объект LegendText в этой коллекции, которыми можно манипулировать индивидуально.

- miThemeRanged имеет один LegendText для ранга (для RangeCategory).
- miThemeIndividualValue, miThemePieChart и miThemeBarChart имеют по одному LegendText для категории (для IndividualValueCategory или для MultiVarCategory).
- miGradSymbol имеет 3 LegendTexts - по одному для каждого примерного размера.
- miDotDensity имеет один LegendText.

Свойства объекта

- Свойство LegendTexts.AllOthersText (Коллекция LegendTexts)
- Свойство LegendText.Text (Объект LegendText)
- Свойство LegendText.Visible (Объект LegendText)

Свойства коллекции

- Свойство LegendTexts.AutoGenerate (Коллекция LegendTexts)
- Свойство LegendTexts.Count (Коллекция LegendTexts)
- Свойство LegendTexts.Item (Коллекция LegendTexts)

Смотрите также

Объект Theme

Свойство LegendTexts.AllOthersText (Коллекция LegendTexts)

Назначение

Возвращает объект LegendText из коллекции. Этот объект LegendText ассоциируется со свойством AllOthersCategory или коллекции RangeCategories или коллекции IndividualValueCategories. Вы можете установить настройки свойства AllOthersText как любого другого объекта LegendText.

Пример на Visual Basic

```

Dim ds As Dataset

Dim lyr as Layer

Dim thm As MapXLib.Theme

Dim AllOthersText As MapXLib.LegendText

Dim lgnd As MapXLib.Legend


Map1.Geoset = "Canada.gst"

Set lyr = Map1.Layers.Item("canada")

Set ds = Map1.Datasets.Add(miDataSetLayer, lyr)

ds.Themes.Add miThemeRanged, "POP_1994", "My Theme", False

Set thm = ds.Themes("My Theme")

Set lgnd = ds.Themes("My Theme").Legend


thm.DataMin = 10000000

thm.DataMax = 30000000

thm.Visible = True


Set AllOthersText = lgnd.LegendTexts.AllOthersText

AllOthersText.Text = "Test Range"

AllOthersText.Visible = True

```

Свойство LegendText.Text (объект LegendText)

Назначение

Содержит текст для диапазонов или категорий в легенде. Это строковая величина и по умолчанию устанавливается при создании тематической карты. Текст легенды автоматически

перестраивается, при перестройке объекта Theme (например, количество рангов изменяется на 6), хотя свойство LegendTexts.AutoGenerate установлено на False.

Пример на Visual Basic

```
' This example sets each LegendText Object in the Map
' to "XXXXXX//" when the legend is double clicked on.
Private Sub Map1_ThemeModifyRequested(ByVal Theme As Object)

    Dim legText as MapXLib.LegendText

    Dim iCount as Integer

    ' Prevent MapX from over-writing the custom LegendText
    ' objects if our theme is ever recomputed

    Theme.Legend.LegendTexts.AutoGenerate = False

    For iCount = 1 to Theme.Legend.LegendTexts.Count

        Theme.Legend.LegendTexts(iCount).Text = "XXXXXX" & _
            Str$(iCount)

    Next

End Sub
```

Пример на C++

```
// This example sets each LegendText Object in the Map to
// "XXXXXX#" when the legend is double clicked on
void CSampleProjectView::OnThemeModifyRequested(LPDISPATCH
ThemeDispatch) {

    CMapXTheme theme;

    long TextsCount,i;

    CString str;

    theme.AttachDispatch(ThemeDispatch,FALSE);

    try {
```

```

TextsCount = theme.GetLegend().GetLegendTexts().GetCount();

// Prevent MapX from over-writing the custom LegendText
// objects if the theme is ever recomputed
theme.GetLegend().GetLegendTexts().SetAutoGenerate(FALSE);

for(i=1;i<=TextsCount;i++) {
    str.Format("XXXXXX%d",i);
    theme.GetLegend().GetLegendTexts().Item(i).SetText(str);
}
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Смотрите также

Объект Theme

Свойство Theme.AutoRecompute

Свойство LegendText.Visible (объект LegendText)

Назначение

Определяет, являются ли диапазоны видимыми для легенды. Логическая величина, по умолчанию True.

Пример на Visual Basic

```
Private Sub btnLegendTextVisible_Click()
    Dim lgnd As MapXLib.Legend

    Set lgnd = Map1.Datasets(1).Themes(1).Legend
    ' Make all of the ranges in the legend invisible.
    nRanges = lgnd.LegendTexts.Count
    For iRange = 1 To nRanges
        lgnd.LegendTexts(iRange).Visible = False
    Next

End Sub
```

Пример на C++

```
void CSampleProjectView::MakeRangesInvisible() {
    CMapXLegend lgnd;
    long nRanges,i;
    try {
        lgnd = m_Map.GetDatasets().Item(1).GetThemes().Item(1).
        GetLegend();
        nRanges = lgnd.GetLegendTexts().GetCount();
        for(i=1;i<=nRanges;i++) {
            lgnd.GetLegendTexts().Item(i).SetVisible(FALSE);
        }
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
}
```



```

    } catch (CancellationToken *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Свойство **LegendTexts.AutoGenerate** (коллекция **LegendTexts**)

Назначение

Определяет, будет ли перевычисляться значение **LegendText** программой MapX когда заново вычисляется объект **Theme**. Это логическая величина и по умолчанию она **True**. Если Вы собираетесь изменять значения **LegendText**, установите **False** что бы MapX не смог учесть Ваши изменения.

Смотрите также

Theme.AutoRecompute

Свойство **LegendTexts.Count** (коллекция **LegendTexts**)

Назначение

Количество объектов **LegendText** в коллекции. Это свойство только для чтения.

Свойство **LegendTexts.Item** (коллекция **LegendTexts**)

Назначение

Возвращает обыкновенный объект **LegendText** из коллекции. Указывает текст.

Объект Map

Карты это базовые блоки для MapX. Каждая карта определяется объектами и коллекциями DataSet, Layer и Annotations.

Свойства объекта

- Свойство Map.Annotations (Объект Map)
- Свойство Map.AreaUnit (Объект Map)
- Свойство Map.AutoRedraw (Объект Map)
- Свойство Map.BackColor (Объект Map)
- Свойство Map.Bounds (Объект Map)
- Свойство Map.CenterX (Объект Map)
- Свойство Map.CenterY (Объект Map)
- Свойство Map.CurrentTool (Объект Map)
- Свойство Map.DataSet (Объект Map)
- Свойство Map.DataSetGeoField (Объект Map)
- Свойство Map.DataSets (Объект Map)
- Свойство Map.DataSetTheme (Объект Map)
- Свойство Map.DefaultConversionResolution (Объект Map)
- Свойство Map.DefaultStyle (Объект Map)
- Свойство Map.DisplayCoordSys (Объект Map)
- Свойство Map.ExportSelection (Объект Map)
- Свойство Map.FeatureFactory (Объект Map)
- Свойство Map.GeoDictionary (Объект Map)
- Свойство Map.GeoSet (Объект Map)
- Свойство Map.GeoSets (Объект Map)
- Свойство Map.GeoSetWidth (Объект Map)
- Свойство Map.hWnd (Объект Map)
- Свойство Map.InfotipSupport (Объект Map)
- Свойство Map.InfotipPopupDelay (Объект Map)
- Свойство Map.Layers (Объект Map)
- Свойство Map.MapUnit (Объект Map)
- Свойство Map.MapPaperHeight (Объект Map)
- Свойство Map.MapPaperWidth (Объект Map)

- Свойство Map.MapScreenHeight (Объект Map)
- Свойство Map.MapScreenWidth (Объект Map)
- Свойство Map.MatchNumericFields (Объект Map)
- Свойство Map.MatchThreshold (Объект Map) 231
- Свойство Map.MaxSearchTime (Объект Map)
- Свойство Map.MouseIcon (Объект Map)
- Свойство Map.MousePointer (Объект Map)
- Свойство Map.MouseWheelSupport (Объект Map)
- Свойство Map.NumericCoordSys (Объект Map)
- Свойство Map.PanAnimationLayer (Объект Map)
- Свойство Map.PaperUnit (Объект Map)
- Свойство Map.PreferCompactLegends (Объект Map)
- Свойство Map.RedrawInterval (Объект Map)
- Свойство Map.Rotation (Объект Map)
- Метод Map.SaveMapAsGeoset (Объект Map)
- Свойство Map.SearchPath (Объект Map)
- Свойство Map.SelectionStyle (Объект Map)
- Свойство Map.Title (Объект Map)
- Свойство Map.TitleText (Объект Map)
- Свойство Map.Version (Объект Map)
- Свойство Map.WaitCursorEnabled (Объект Map)
- Свойство Map.Zoom (Объект Map)

Методы объекта

- Метод Map.AboutBox (Объект Map)
- Метод Map.ClipLine (Объект Map)
- Метод Map.CliplineV (Объект Map)
- Метод Map.ConvertCoord (Объект Map)
- Метод Map.ConvertCoordV (Объект Map)
- Метод Map.CreateCustomTool (Объект Map)
- Метод Map.Distance (Объект Map)
- Метод Map.ExportMap (Объект Map)
- Метод Map.IsPointVisible (Объект Map)
- Метод Map.Pan (Объект Map)
- Метод Map.PrintMap (Объект Map)
- Метод Map.PropertyPage (Объект Map)

- Метод Map.Refresh (Объект Map)
- Метод Map.SetSize (Объект Map)
- Метод Map.ZoomTo (Объект Map)

Смотрите также

Объекты DataSet

Объект Layer

Объекты Annotations

Метод Map>AboutBox (Объект Map)

Назначение

Отображает About Box, который содержит информацию о версии MapX и сведения о разработчике.

Синтаксис

ОБЪЕКТ.AboutBox

Свойство Map.Annotations (Объект Map)

Назначение

Коллекция аннотаций для карты.

Смотрите также

Объект и коллекция Annotation

Свойство Map.AreaUnit (Объект Map)

Назначение

Единицы используемые в методах/свойствах Area. Берется значение AreaUnitConstants. По умолчанию это miUnitSquareMile.

Свойство Map.AutoRedraw (Объект Map)

Назначение

Включает или отключает обновление экрана. Логическая величина.

Примечание

Установите AutoRedraw в FALSE чтобы выключить обновление экрана, или в TRUE для нормального обновления.

Когда свойство или методы приводят к изменению карты (такое как Map.Zoom), карта перерисовывается. Если Вы хотите сделать несколько изменений карты, это свойство позволит отключить перерисовку, сделать несколько изменений, а затем включить прорисовку снова. Это предотвратит лишние прорисовки. Когда установлено TRUE, перерисуеться карта целиком.

Пример на C++

```
void CSampleProjectView::OnChangeMap() {
    try {
        // Temporarily disable redraws while we change the _
        map parameters
        // If this was not here, MapX might redraw the screen _
        after each change to the Map
        m_Map.SetAutoRedraw(FALSE);
        // Zoom in, offset, and rotate the Map
        // Note: the Map.ZoomTo method can also be used
```

```

    m_Map.SetZoom(m_Map.GetZoom()*0.5);
    m_Map.SetCenterX(m_Map.GetCenterX()+5.0);
    m_Map.SetCenterY(m_Map.GetCenterY()-5.0);
    m_Map.SetRotation(30);
    m_Map.SetAutoRedraw(TRUE);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Пример на Visual Basic

```

' Turn off map redraws.
Map1.AutoRedraw = False

' Using the map's PropertyPage dialog, add/remove some layers from the
map.
Map1.PropertyPage

' Turn map redraws back on and see the new layers.
Map1.AutoRedraw = True

```

Свойство Map.BackColor (Объект Map)

Назначение

Определяет цвет подложки карты “стирается” перед началом рисования карты. Указывает величину OLE_COLOR. Это может быть один из сплошных цветов или один из Windows System Colors такой как 'Window Background'. (Большинство контейнеров, таких как Visual Basic показывают специальный диалог для определения свойств страницы для свойства типа OLE_COLOR.)

Смотрите также

Значения OLE_COLOR

Свойство Map.Bounds (Объект Map)

Назначение

Свойство Map.Bounds (которое называлось Map.MBR в ранних версиях) позволяет Вым установить границы окна карты в координатах карты. Берется объект типа Rectangle.

Пример на Visual Basic

```
dim rect as new MapXLib.Rectangle
'correct way to set the map's bounding rectangle
rect.Set -124.82, 18.08, -52.14, 53.01
set map1.Bounds = rect
'incorrect way to set bounds7 - This will have no effect.
map1.Bounds.Set -124.82, 18.08, -52.14, 53.01
```

Пример на C++

```
void CSampleProjectView::SetBounds() {
    CMapXRectangle rect;
    if(!rect.CreateDispatch(rect.GetClsid())) {
```

```

        TRACE0("Could not Create rectangle object");
        return;
    }

    try {
        // The correct way to set the map's bounding rectangle
        rect.Set(-124.82,18.08,-52.14,53.01);
        m_Map.SetBounds(rect);

        // Incorrect way- this will have no effect
        m_Map.GetBounds().Set(-124.82,18.08,-52.14,53.01);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Смотрите также

Объект Rectangle

Свойство Map.CenterX (Объект Map)

Назначение

Определяет текущую координату X для центра карты. Это свойство имеет двойную точность, представляет долготу (отрицательные значения для западного полушария).

Обсуждение

Свойство CenterX устанавливается при загрузке нового набора данных Geoset.

Смотрите также

Свойство Map.CenterY

Метод Map.ZoomTo

Свойство Map.CenterY (Объект Map)

Назначение

Определяет текущую координату Y для центра карты. Это свойство имеет двойную точность.

Обсуждение

Свойство CenterY устанавливается, когда загружается новый Geoset.

Смотрите также

Свойство Map.CenterX

Метод Map.ZoomTo

Метод Map.ClipLine (Объект Map)

Назначение

Возвращает True если линия целиком или частично в видимой части окна карты, или False если линия полностью вне окна карты. В добавок, линия обрезается так, что если конечная точка линии вне окна карты, то далее конечной точкой станет та, которая пересекается краем окна карты.

Синтаксис

[Boolean=]ОБЪЕКТ.ClipLine (X1, Y1, X2, Y2)

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Map.
--------	-------------

X1	Двойная точность (by Reference). На входе: Долгота первой
----	---

	конечной точки необрезанной линии. На выходе: долгота первой конечной точки обрезанной линии.
Y1	Двойная точность (by Reference). На входе: широта первой конечной точки необрезанной линии. На выходе: широта первой конечной точки обрезанной линии.
X2	Двойная точность (by Reference). На входе: Долгота второй конечной точки необрезанной линии. На выходе: долгота второй конечной точки обрезанной линии.
Y2	Двойная точность (by Reference). На входе: широта второй конечной точки необрезанной линии. На выходе: широта второй конечной точки обрезанной линии.

Примечание

Все параметры посылаемые при ссылке должны быть в системе координат Map.NumericCoordSys.

Возвращаемое значение ClipLine это True если любая часть линии попадает внутрь видимой части окна карты и False в других случаях.

Если линия полностью располагается внутри окна Карты, ClipLine вернет True. Если некоторые части линии внутри окна Карты, координаты будут корректироваться, так что они примут форму той части линии, которая в пределах окна и ClipLine вернет True. Если линия полностью вне окна Карты, то ClipLine вернет False.

Смотрите также

Метод Map.ClipLineV

Метод Layers.AddUserDrawLayer

Событие DrawUserLayer

Свойство Map.NumericCoordSys

Метод Map.CliplineV (Объект Map)

Назначение

Метод обрезает линию и устанавливает новые значения x1, y1, x2, y2. Map.ClipLineV идентичен методу Map.ClipLine, но использует при вычислении величины двойной точности.

Синтаксис

[Boolean=] ОБЪЕКТ.ClipLineV(X1, Y1, X2, Y2)

Часть	Описание
ОБЪЕКТ	Объект Map.
X1	Переменная (by Reference). На входе: Долгота первой конечной точки необрезанной линии. На выходе: долгота первой конечной точки обрезанной линии.
Y1	Переменная (by Reference). На входе: широта первой конечной точки необрезанной линии. На выходе: широта первой конечной точки обрезанной линии
X2	Переменная (by Reference). На входе: Долгота второй конечной точки необрезанной линии. На выходе: долгота второй конечной точки обрезанной линии.
Y2	Переменная (by Reference). На входе: широта второй конечной точки необрезанной линии. На выходе: широта второй конечной точки обрезанной линии

Все параметры являются переменными. Они должны содержать числовые значения и иметь систему координат Map.NumericCoordSys. Если параметры не могут быть конвертированы в числа, то генерируется ошибка (код 1111)..

Возвращаемая величина ClipLineV это True если любая часть линии попадает внутрь окна и False в других случаях.

Если линия целиком в окне Карты, ClipLineV вернет True. Если часть линии внутри окна Карты, координаты будут изменены так что они сформируют часть линии целиком внутри окна карты и ClipLineV вернет True. Если линия полностью вне окна карты, ClipLineV будет возвращать False.

Смотрите также

Метод Map.ClipLine

Метод Map.NumericCoordSys

Метод Map.ConvertCoord (Объект Map)

Назначение

Конвертирует координаты карты в экранные или наоборот, в зависимости от параметра Dir существующего в этом методе. Экранные координаты задаются в пикселах.

Синтаксис

OBJECT.ConvertCoord (ScreenX, ScreenY, MapX, MapY, Dir)

Часть	Описание
OBJECT	Объект Map.
ScreenX	Экранная координата x в пикселах.Целая величина.
ScreenY	Экранная координата y в пикселах.Целая величина (Single value).
MapX	Координата карты x (долгота). Двойная точность.
MapY	Координата карты y (широта). Двойная точность
Dir	Направление конвертации координат, или от Карты к экрану или наоборот. Берется значение ConversionConstants.

Какие получатся в итоге координаты зависит от параметра Dir.

Пример на Visual Basic

```
`This sample shows how to use the Map.ConvertCoord method to
`convert from screen units (pixels) to Map X/Y
`coordinates in Degrees of Longitude/Latitude.
Private Sub Command1_Click()
    Dim ScreenX As Single
    Dim ScreenY As Single
```

```

Dim MapX As Double

Dim MapY As Double

`Get the center of the Map in screen coordinates:

ScreenX = Map1.MapScreenWidth / 2

ScreenY = Map1.MapScreenHeight / 2


`Convert the screen center to map coordinates
Map1.ConvertCoord ScreenX, ScreenY, MapX, MapY, miScreenToMap

`Display the converted coordinates and the map's centerX/Y
`properties: Note that these numbers may not be exactly
`the same due to rounding.

Debug.Print "Converted Center: " & MapX & ", " & MapY

Debug.Print "      Map Center: " & Map1.CenterX & ", " & _
Map1.CenterY

End Sub

```

Пример на C++

```

void CSampleProjectView::ConvertCenter() {

    float ScreenX,ScreenY;

    double MapX,MapY;

    CString message;

    try {

        // Get the center of the Map in screen coordinates

        ScreenX = (float)m_Map.GetMapScreenWidth() / 2;

        ScreenY = (float)m_Map.GetMapScreenHeight() / 2;
    }
}

```

```

// Convert the screen center to map coordinates
m_Map.ConvertCoord(&ScreenX,&ScreenY,&MapX,&MapY,
miScreenToMap);

// Display the converted coordinates and the map's centerX/Y
// properties: not that these numbers may not be exactly
// the same due to rounding.

message.Format("Converted Center: %f, %f\n      Map Center:
%f, %f",MapX,MapY,m_Map.GetCenterX(),m_Map.GetCenterY());
} catch (CleDbDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (CleDbException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Метод Map.ConvertCoordV (Объект Map)

Назначение

Конвертирует координаты карты в экранные или наоборот, в зависимости от параметра Dir существующего в этом методе. Это вариант метода ConvertCoord. Экранные координаты определяются в пикселах.

Синтаксис

ОБЪЕКТ.ConvertCoordV (ScreenX, ScreenY, MapX, MapY, Dir)

Часть

Описание

ОБЪЕКТ	Объект Map.
ScreenX	Экранная координата x в пикселах. Переменная величина.
ScreenY	Экранная координата y в пикселах. Переменная величина.
MapX	Координата x (долгота). Переменная величина.

MapY Координата y (широта). Переменная величина.
 Dir Направление конвертации координат, или от Карты к экрану или наоборот. Берется значение ConversionConstants.

Какие получатся в итоге координаты (экранные или Карты) зависит от параметра Dir.

Пример на Visual Basic

```
`This sample shows how to use the Map.ConvertCoordV method to
`convert from screen units (pixels) to Map X/Y
`coordinates in Degrees of Longitude/Latitude.
Private Sub Command1_Click()
    Dim ScreenX As Variant
    Dim ScreenY As Variant
    Dim MapX As Variant
    Dim MapY As Variant

    `Get the center of the Map in screen coordinates:
    ScreenX = Map1.MapScreenWidth / 2
    ScreenY = Map1.MapScreenHeight / 2

    `Convert the screen center to map coordinates
    Map1.ConvertCoordV ScreenX, ScreenY, MapX, MapY, miScreenToMap
    `Display the converted coordinates and the map's centerX/Y
    `properties: Note that these numbers may not be exactly the
    `same due to rounding.
    Debug.Print "Converted Center: " & MapX & ", " & MapY
    Debug.Print "      Map Center: " & Map1.CenterX & ", " & _
    Map1.CenterY
End Sub
```

Пример на C++

```

void CSampleProjectView::ConvertCenterV() {
    float ScreenX,ScreenY;
    double MapX,MapY;
    CString message;

    try {
        // Get the center of the Map in screen coordinates
        ScreenX = (float)m_Map.GetMapScreenWidth() / 2;
        ScreenY = (float)m_Map.GetMapScreenHeight() / 2;

        COleVariant ScreenXVar(ScreenX),ScreenYVar(ScreenY),
        MapXVar(MapX), MapYVar(MapY);

        // Convert the screen center to map coordinates
        m_Map.ConvertCoordV(&ScreenXVar,&ScreenYVar,&
        MapXVar,&MapYVar,miScreenToMap);

        // Display the converted coordinates and the map's centerX/Y
        // properties: not that these numbers may not be exactly
        // the same due to rounding.

        message.Format("Converted Center: %f, %f\n      Map Center:
        %f, %f", MapXVar.fltVal,MapYVar.fltVal,m_Map.GetCenterX(),
        m_Map.GetCenterY());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {

```



```

        e->ReportError();

        e->Delete();
    }
}

```

Метод Map.CreateCustomTool (Объект Map)

Назначение

Создает пользовательский инструмент, затем, когда он используется, посылает событие ToolUsed.

Внимание: Если Вы создаете Custom Tool типа окружность и в событии Tool_Used среды MapX определяется SelectByRadius с величиной, передаваемой в событие, результаты получаются не такими как если Вы выбираете объекты в Radius Select Tool. Метод SelectByRadius не будет точно выбирать, поскольку такой круг не учитывает проекцию карты. Выборки сделанные Radius Select Tool точно выбирают все объекты в заданной окружности.

Синтаксис

OBJECT.CreateCustomTool (ToolNumber, Type, Cursor, [ShiftCursor], [CtrlCursor], [InfoTips])

Часть	Описание
OBJECT	Объект Map.
ToolNumber	Количество инструментов используемое в дальнейшем. Это целое между 1 и 999.
Type	Тип описывает поведение инструмента. Берется значение ToolTypeConstants.
Cursor	Курсор в случае когда инструмент создан в CurrentTool и курсор в окне карты. Значение выбирается из коллекции CursorConstants.
ShiftCursor	Переменная: Значение CursorConstants, указывающее, что курсор должен появиться, пока нажата клавиша SHIFT. Не обязательный параметр. Если он пропущен, клавиша SHIFT не действует на курсор.
CtrlCursor	Переменная: Значение CursorConstants, указывает, что курсор должен появиться, пока нажата клавиша CTRL. Не обязательный параметр. Если он пропущен, клавиша CTRL не действует на курсор.

InfoTips

Логическое: Установите true если надо показать всплывающие подсказки InfoTips. По умолчанию равно false.

Примечание

Могут быть созданы пользовательские инструменты, имеющие особое поведение. Например, можно построить инструмент такой, который показывает информацию об объекте, когда на объекте щелкнули мышкой. Здесь мы определим инструмент типа `miToolTypePoint`, и в обработчике событий `ToolUsed`, напомним код для отображения информации при выделении объекта.

Пример на Visual Basic

```
Private Sub Command1_Click()
    Map1.CurrentTool = 99
End Sub

Private Sub Command2_Click()
    Map1.CurrentTool = miZoomInTool
End Sub

Private Sub Form_Load()
    Map1.CreateCustomTool 99, miToolType, miCrossCursor
End Sub

Private Sub Map1_ToolUsed(ByVal ToolNum As Integer, _
    ByVal X1 As Double, ByVal Y1 As Double, _
    ByVal X2 As Double, ByVal Y2 As Double, _
    ByVal Distance As Double, ByVal Shift As Boolean, _
    ByVal Ctrl As Boolean, EnableDefault As Boolean)
    Select Case ToolNum
        Case 99
            Debug.Print "Our custom tool was used at these coordinates:"
```

```

        Debug.Print "  (" & X1 & "," & Y1 & ") - (" & X2 & "," & Y2 & " _
        & ")"
    Case miZoomInTool
        Debug.Print "I will disallow the use of the ZoomIn tool."
        EnableDefault = False
    Case Else
        Debug.Print "Another tool was used: " & ToolNum
    End Select

    If Ctrl Then Debug.Print "  The Control Key was pressed"
    If Shift Then Debug.Print "  The Shift Key was pressed"
End Sub

```

Пример на C++

```

void CSampleProjectView::OnZoomInToolSelect() {
    try {
        m_Map.SetCurrentTool(miZoomInTool);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

void CSampleProjectView::OnCustomToolSelect() {
    try {

```

```

        m_Map.SetCurrentTool(99);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

void CSampleProjectView::CreateTool() {
    try {
        m_Map.CreateCustomTool(99,miToolTypeLine,miCrossCursor);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

void CSampleProjectView::OnToolUsed(short ToolNum, double X1, double Y1,
double X2, double Y2, double Distance, BOOL Shift, BOOL Ctrl, BOOL*
EnableDefault) {
    CString msg;

    switch(ToolNum) {

```

```

    case 99:
        msg.Format("Our custom tool was used at these coordinates: \n (%d,
%d) - (%d, %d)\n",X1,Y1,X2,Y2);

        break;

    case miZoomInTool:
        msg = "I will disallow the use of the ZoomIn tool\n";
        *EnableDefault = FALSE;

        break;

    default:
        msg.Format("Another tool was used: %d\n",ToolNum);

};

if(Ctrl)
    msg += "The Control Key was pressed\n";

if(Shift)
    msg += "The Shift Key was pressed\n";

AfxMessageBox(msg);

}

)

```

Смотрите также

Свойство Map.CurrentTool

Событие ToolUsed

Свойство Map.CurrentTool (Объект Map)

Назначение

Определяет текущий инструмент. Берется значение ToolConstants. По умолчанию это miArrowTool.

Примечание

Разные инструменты будут заставлять мышку по-разному реагировать на разные задачи. Например, если текущий инструмент установлен на `miLabelTool`, то когда Вы щелкаете мышкой, появится подпись рядом с объектом. Курсор мышки меняется в зависимости от того, какой инструмент используется.

Если Вы определили пользовательский инструмент, установите его в `CurrentTool` указав его номер `ToolNumber`.

Пример

```
`Make the select tool active:
Map1.CurrentTool = miSelectTool
```

Смотрите также

Метод `Map.CreateCustomTool`

Событие `ToolUsed`

Свойство `Map.DataSet` (Объект `Map`)

Назначение

Указывает объект `DataSet` который будет добавлен. Это работает только в режиме отладки (Design time) а не в работе (Run time). Это свойство используется только в Microsoft Visual Basic.

Примечание

Visual Basic показывает список всех Data Controls в текущей форме в списке свойств Visual Basic для `MapX`. Выбирая одну запись из списка Data Controls. Для свойств `DataSet` будет отдана команда `MapX` создать `DataSet` для Run time с данными, определенными в `DataControl`.

Это похоже на метод `DataSets.Add`. Параметр `GeoField` может быть определен используя свойство `Map.DataSetGeoField`. Вы можете также выбрать какие поля и методы объединения будут использоваться для каждого поля.

Смотрите также

Метод `DataSets.Add`,

Свойство `Map.DataSetGeoField`

Свойство `Map.DataSetGeoField` (Объект `Map`)**Назначение**

Определяет, какое поле `DataSet` указанное в свойстве `Map.DataSet` содержит географическую информацию. Это строковая величина, указывающая имя поля.

Примечание

Это похоже на параметр `GeoField` метода `DataSets.Add`, но `GeoField` для `DataSet` работает только в процессе отладки.

Смотрите также

Свойство `Map.DataSet`

Метод `DataSets.Add`

Свойство `Map.DataSets` (Объект `Map`)**Назначение**

Коллекция наборов данных `Datasets` для карты.

Смотрите также

объект и коллекция `DataSet`

Свойство `Map.DataSetTheme` (Объект `Map`)

Назначение

Указывает тип тематической карты для набора данных, указанного в свойстве `Map.DataSet`. Используются величины `ThemeTypeConstants`. По умолчанию это `miAutoThemeType`.

Обсуждение

Если Вы указываете `DataSet` во время отладки (свойство `Map.DataSet`), то Вы можете отобразить эти данные на тематической карте. Свойство определяет стиль тематики. Это похоже на метод `Themes.Add`, но работает только во время отладки. Вы можете выбрать, какое поле или поля являются основой для тематической карты в тематической таблице свойств страницы `MapX`.

Свойство `DataSetTheme` игнорируется если свойство `DataSet` не установлено.

Смотрите также

Метод `Themes.Add`

Свойство `Map.DataSet`

Свойство `Map.DefaultConversionResolution` (Объект `Map`)

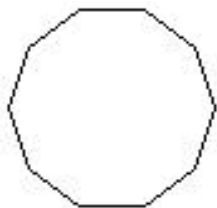
Назначение

Установка системы на свойство чтение/запись, которое определяет разрешение (количество узлов) используемое в специальных типах объектов `MapInfo` таких как дуги, полигоны или полилинии или при генерации окружностей, эллипсов, дуг или буферов. Это целая величина, принимающая значения до 32763.

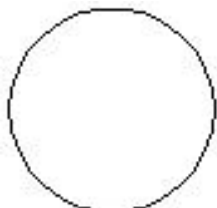
Примечание

Методы (перечисленные ниже) позволяют Вам ясно указать оптимальный параметр разрешения. Если Вы не укажете значение разрешения, `MapX` использует установку разрешения, которая указана в свойстве `DefaultConversionResolution`.

Это свойство действует при генерировании буферов, окружностей, эллипсов и дуг. Когда генерируются такие объекты Вы можете определять разрешение (или степень сглаживания) объектов. Смотрите рис.1 и рис.2.



окружность с разрешением 10 узлов



окружность с разрешением 30 узлов

Увеличение разрешения приводит к большему сглаживанию; однако более сглаженные объекты дольше отрисовываются и занимают больше места.

Это свойство также определяет разрешение, если используются таблицы MapInfo которые содержат такие объекты как дуги, окружности, прямоугольники, скругленные прямоугольники. MapX конвертирует такие объекты в полигоны или полилинии. Смотрите разделы Automatic Conversion of Special Features в электронной Справке.

Смотрите также

Метод `FeatureFactory.BufferFeatures`

Метод `FeatureFactory.CreateArc`

Метод `FeatureFactory.CreateCircularRegion`

Метод `FeatureFactory.CreateEllipticalRegion`

Метод `Layer.SearchWithinDistance`

Свойство `Map.DefaultStyle` (Объект `Map`)

Назначение

Это объект `Style` для символьных и для текстовых объектов, когда они создаются методами `Annotation.AddSymbol` или `Annotation.AddText`.

Обсуждение

По умолчанию стиль это серая звезда размером 12. Текст по умолчанию 10 Arial.

Пример на Visual Basic

```
Private Sub Command1_Click()
    With Map1.DefaultStyle
        .SymbolCharacter = 36
        .SymbolFontColor = RGB(255, 0, 0)
        .SymbolFont.Size = 24
    End With
End Sub
```

Пример на C++

```
void CSampleProjectView::SetDefaultStyle() {
    CY fontSize;
    fontSize.int64 = 24;
    try {
        m_Map.GetDefaultStyle().SetSymbolCharacter(36);
        m_Map.GetDefaultStyle().SetSymbolFontColor(255);
        m_Map.GetDefaultStyle().GetSymbolFont().SetSize(fontSize);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

}

Смотрите также

Стиль Style

Свойство Map.DisplayCoordSys (Объект Map)

Назначение

Возвращает объект CoordSys доступный для чтения и записи - это координатная система, определяющая в какой проекции будет отображаться карта.

Примечание

DisplayCoordSys представляет систему координат (или проекцию) в которой отображается карта. По умолчанию, координатная система карты определяется рабочим набором geoset. При изменении DisplayCoordSys (методом CoordSys.Set), можно показать карту в другой системе координат.

Свойство DisplayCoordSys действует только на отображение карты. Для изменения проекции карты в таблице, используйте свойство NumericCoordSys..

Метод Map.Distance (Объект Map)

Назначение

Вычисляет расстояние между двумя указанными точками на карте. Расстояние возвращается значением двойной точности, в единицах, определенных в свойстве MapUnit.

Синтаксис

[distance =]OBJECT.Distance(x1,y1,x2,y2)

Часть	Описание
-------	----------

OBJECT	Объект Map.
x1	x координата первой точки (долгота). Двойная точность.
y1	y координата первой точки (широта). Двойная точность.
x2	x координата второй точки (долгота). Двойная точность.

y2 у координата второй точки (широта). Двойная точность.

Примечание

Расстояние рассчитывается используя Great Circle caculation.

Пример на Visual Basic

```
Private Sub Command1_Click()
    Dim earthCircumference As Double
    Map1.MapUnit = miUnitMile
    earthCircumference = 2 * Map1.Distance(0, 0, 180, 0)
    Debug.Print "Earth's Circumference: " & earthCircumference
End Sub

Private Sub Form_Load()
    Map1.CurrentTool = miZoomInTool
End Sub
```

Пример на C++

```
void CSampleProjectView::EarthDistance() {
    double earthCircumference;
    CString msg;
    try {
        m_Map.SetMapUnit(miUnitMile);
        earthCircumference = 2 * m_Map.Distance(0,0,180,0);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
    }
```

```
        e->Delete();
    }

    msg.Format("Earth's Circumference: %d miles",earthCircumference);
    AfxMessageBox(msg);
}
```

Смотрите также

Свойство Map.MapUnit

Метод Map.ExportMap (Объект Map)

Назначение

Экспортирует окно карты.

MapX может экспортировать карту в следующие форматы:

- | | | |
|-----|-----|-----|
| JPG | TIF | GIF |
| PNG | WMF | PSD |
| BMP | | |

Синтаксис

ОБЪЕКТ.ExportMap (Location, Format, [Width], [He])

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Map.
Location	Спецификация файла - место куда он будет помещен. Если используется ключевое слово ‘CLIPBOARD’, изображение попадет в clipboard.
Format	Выходной формат. Берется значение ExportFormatConstants.
Width	Ширина выходного изображения. Двойная точность, ширина определяется в единицах бумаги (Map.PaperUnit). Это

Height	<p>необязательная величина, если она не указана, используется Map.MapPaperWidth.</p> <p>Высота выходного изображения. Двойная точность, определяется в единицах бумаги (Map.PaperUnit). Это необязательная величина, если она не указана, используется Map.MapPaperHeight.</p>
--------	--

Примечание

Если на карте есть выделенные объекты, которые Вы можете не захотеть экспортировать; смотрите свойство Map.ExportSelection.

Пример на Visual Basic

```

`This sample demonstrates the Map.ExportMap method. It uses
`the method to place a map in the clipboard as a BMP.
Private Sub Command1_Click()
    Map1.PaperUnit = miUnitCentimeter
    `Export a 12 cm by 9 cm map to the clipboard in BMP Format
    Map1.ExportMap "clipboard", miFormatBMP, 12, 9
    If Clipboard.GetFormat(vbCFBitmap) Then
        Debug.Print "Clipboard contains a BMP"
    Else
        Debug.Print "Clipboard does not contain a BMP"
    End If
End Sub

```

Пример на C++

```

void CSampleProjectView::ExportMapToClipboardAsBMP() {
    try {

```

```

    m_Map.SetPaperUnit(miUnitCentimeter);

    // Export a 12 cm by 9 c, map to the clipboard in BMP format
    m_Map.ExportMap("clipboard",miFormatBMP,12.0,9.0);
} catch (COleDispatchException *e) {
    e->ReportError();

    e->Delete();
} catch (COleException *e) {
    e->ReportError();

    e->Delete();
}
}

```

Смотрите также

Свойство Map.ExportSelection

Свойство Map.PaperUnit

Свойство Map.ExportSelection (Объект Map)

Назначение

Возвращает или устанавливает логическое свойство, контролирующее, является ли включена ли возможность экспорта выделенных объектов.

Если установлено True, то на выходе методов ExportMap и PrintMap в экспорт будут включаться все выделенные объекты, если установлено False, то на выходе выделенных объектов не будет.

Пример

```

'export a jpg image for display in a browser, and
'include the selection highlighting in the image
Map1.ExportSelection=True

```

```
Map1.ExportMap "c:\temp\map.jpg", miFormatJPEG
```

Смотрите также

Для контроля стиля в котором будет прорисована выборка, установите свойство Map.SelectionStyle.

Свойство Map.FeatureFactory (Объект Map)

Назначение

Возвращает объект FeatureFactory, который поддерживает различные методы для создания и изменения объектов карты.

Примечание

Это свойство возвращает объект FeatureFactory, который имеет методы, поддерживающие различные задачи редакции—создание объектов, генерирование буферов, комбинирование объектов, и определения есть ли пересечение объектов.

Свойство Map.GeoDictionary (Объект Map)

Назначение

Определяет путь к GeoDictionary. Строковая величина.

Примечание

GeoDictionary это файл (обычно названный geodict.dct) содержащий информацию в каких слоях какая информация. Это та же папка, в которой хранятся файлы GeoSet (*.gst).

По умолчанию путь к GeoSet path запрашивается из базы данных registry под ключом: HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\4.0\<Value of GeoDictionary Property>

Значение этого регистрационного ключа указывает на две вещи:

- Файл GeoDictionary, который содержит информацию о том, какие слои карты какую информацию содержат. (MapX может использоваться без GeoDictionary.)
- Директорию, где по умолчанию хранятся данные. Эта директория опрашивается на предмет GeoSet файлов (*.gst) которые показаны в Property Pages и коллекции Map.GeoSets.

Существуют три допустимых варианта для registry:

- Может быть указан полный путь для файла Geodictionary (типа "C:\Program Files\MapInfo MapX\Maps\geodict.dct"). Директория с данными устанавливается так, чтобы путь к файлу Geodictionary назывался по данному ключу (в нашем примере, "C:\Program Files\MapInfo MapX\Maps\"). Находящийся в памяти словарь инициализируется из файла Geodictionary и слои, содержащиеся в Geodictionary могут находиться автоматически. Слои добавленные к карте оказываются добавленными и в память словаря, так что их тоже можно будет находить автоматически.
- Может быть указан только путь без имени файла (типа "C:\Program Files\MapInfo MapX\Maps\"). Директория с данными разместится в этой директории. Словарь geodictionary, находящийся в памяти, инициализируется пустым. Слои, добавленные к карте, также добавляются к словарю geodictionary, находящемуся в памяти, так что их тоже можно будет находить автоматически.
- Может быть пустым. Словарь geodictionary, находящийся в памяти, инициализируется пустым. Слои, добавляемые к карте, также добавляются к словарю geodictionary, находящемуся в памяти, так что их можно будет искать автоматически.

По умолчанию значение Map.GeoDictionary это "GeoDictionary", значит, что MapX будет запрашивать ввод HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\4.0\GeoDictionary.

Смотрите также

Свойство Map.GeoSet

Свойство Map.GeoSet(Объект Map)

Назначение

Это свойство, открытое для чтения/записи. Устанавливает рабочий набор GeoSet. Строковая величина. Может использоваться спецификация файла GeoSet, или 'дружественное имя' ('дружественное имя' ссылается на имя рабочего набора geoset имеющегося в geodictionary).

Примечание

GeoSet это коллекция слоев карты и их настроек. GeoSet может быть определен в период отладки. Если GeoSet устанавливается в процессе работы программы, то GeoSet не имеет информации, затем формируется новый GeoSet с рабочим набором. Что бы просто удалить все

загруженные слои и наборы данных и не загружать новый geoset, укажите пустую строку. По умолчанию загружаемый GeoSet это US.GST.

Пример на C++

```
// Map.GeoSet Property
void CSampleProjectView::CanadaMap() {
    try {
        // Close the existing set of map layers and load the Canada map
        TRACE0("Old Geoset: " + m_Map.GetGeoSet());
        m_Map.SetGeoSet("Canada.gst");
        TRACE0("New Geoset: " + m_Map.GetGeoSet());
    } catch (CleDbDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (CleDbException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Пример на Visual Basic

```
Private Sub CanadaMap()
    ' Close the existing set of map layers and load the Canada map
    Debug.Print "Old Geoset: " & Map1.GeoSet
    Map1.GeoSet = "Canada.gst"
    Debug.Print "New Geoset: " & Map1.GeoSet
End Sub
```

Смотрите также

Метод `Layers.AddGeoSetLayers`

Свойство `Map.GeoSets` (Объект `Map`)**Назначение**

Это свойство, открытое только для чтения, возвращает коллекцию объектов `Geoset`. Эта коллекция состоит из всех наборов данных `geosets` в директории `Maps`.

Смотрите также

Объект и коллекция `Geoset`

Свойство `Map.GeoSetWidth` (Объект `Map`)**Назначение**

Свойство открытое только для чтения, представляющее ширину `GeoSet` (максимальная ширина всех слоев в карте). Двойная точность в единицах, указанных свойством `MapUnit`. Это свойство полезно, когда Вам надо установить масштаб такой чтобы был показан целиком самый протяженный слой карты.

Пример на C++

```
// Map.GeoSetWidth Property
void CSampleProjectView::ViewGeoSetExtents() {
    try {
        // The Zoom of a Map is defined as the width of the map control in
        Map Units

        // We can simply set the zoom to the GeoSet width to view the entire
        GeoSet

        m_Map.SetZoom(m_Map.GetGeoSetWidth());
    } catch (COleDispatchException *e) {
```

```

        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub CmdViewGeoSetExtents_Click()
    ' The Zoom of a Map is defined as the width of the map
    ' control in Map Units
    ' We can simply set the zoom to the GeoSet width to view
    ' the entire GeoSet
    Map1.Zoom = Map1.GeoSetWidth
End Sub

```

Свойство Map.hWnd (Объект Map)

Назначение

Обработчик контроля за окном MapX. Действует только во время работы программы. Свойство предназначенное только для чтения.

Свойство Map.InfotipSupport (Объект Map)

Назначение

Определяет, обработать ли MapX всплывающие подсказки InfoTips в окне карты. Это логическое свойство, и по умолчанию равно True.

Примечание

Когда InfoTipSupport равно True, InfoTips будет работать для следующих инструментов: Arrow Tool, Select Tool, Label Tool, и для любого пользовательского инструмента, в котором параметр ShowInfoTips для Map.CreateCustomTool будет True.

Смотрите также

Свойство Map.InfotipPopupDelay

Метод Map.CreateCustomTool

Свойство Map.InfotipPopupDelay (Объект Map)

Назначение

Это свойство определяет время с момента указанием курсором на объекте и до момента появления всплывающей подсказки. Это свойство применяется только когда Map.InfotipSupport имеет значение True. Это целая величина, указанная в миллисекундах. По умолчанию равна 500.

Смотрите также

Свойство Map.InfotipSupport

Метод Map.IsPointVisible (Объект Map)

Назначение

Возвращает True или False в зависимости от того, находится ли заданная точка (в координатах карты) внутри видимой части окна карты.

Синтаксис

```
[ bool= ] ОБЪЕКТ.IsPointVisible (X, Y )
```

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Map.
--------	-------------

X	Координаты карты (широта)
---	---------------------------

Y Координаты карты (долгота)

Свойство Map.Layers (Объект Map)

Назначение

Коллекция Layers для карты.

Смотрите также

Объект и коллекция Layer

Свойство Map.MapUnit (Объект Map)

Назначение

Устанавливает единицы для расчета расстояний. Берется величина MapUnitConstants, по умолчанию это miUnitMile.

Пример на C++

```
// Map.MapUnit Property
void CSampleProjectView::EarthCircumference() {
    double CircKm,CircMi;
    CString msg;

    try {
        // Save the old MapUnit so that we can change it back
        MapUnitConstants oldUnit = m_Map.GetMapUnit();

        // Get the Earth's circumference in Kilometers
        m_Map.SetMapUnit(miUnitKilometer);
```

```

    CircKm = 2.0 * m_Map.Distance(0,0,180,0);

    // Get the Earth's circumference in Miles
    m_Map.SetMapUnit(miUnitMile);
    CircMi = 2.0 * m_Map.Distance(0,0,180,0);

    msg.Format("The Earth has a circumference of %f miles (%f
kilometers)",CircMi,CircKm);
    AfxMessageBox(msg);
    m_Map.SetMapUnit(oldUnit);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Пример на Visual Basic

```

Private Sub EarthCircumference()
    Dim CircKm as Double
    Dim CircMi as Double

    Dim oldUnit as MapUnitConstants

    ' Save the old MapUnit so that we can change it back

```

```

oldUnit = Map1.MapUnit

' Get the Earth's circumference in Kilometers
Map1.MapUnit = miUnitKilometer
CircKm = Map1.Distance(0,0,180,0) * 2.0

' Get the Earth's circumference in Miles
Map1.MapUnit = miUnitMile
CircMi = Map1.Distance(0,0,180,0) * 2.0

Debug.Print "The Earth has a circumference of " & CircMi _
& " miles (" & CircKm & " kilometers)"

Map1.MapUnit = oldUnit

End Sub

```

Смотрите также

MapUnitConstants

Свойство Map.MapPaperHeight (Объект Map)

Назначение

Определяет высоту карты. Двойная точность, в терминах бумажных единиц (Map.PaperUnit.). Свойство только для чтения.

Свойство Map.MapPaperWidth (Объект Map)

Назначение

Определяет ширину карты. Двойная точность, в терминах бумажных единиц (Map.PaperUnit.). Свойство только для чтения.

Свойство Map.MapScreenHeight (Объект Map)

Назначение

Свойства MapScreenHeight и MapScreenWidth облегчают позиционирование легенд. Эти свойства содержат ширину и высоту окна карты в пикселах. Поскольку легенды тоже содержат их ширину и высоту, это облегчает позиционирование легенд.

Пример на C++

```
// Map.MapScreenWidth Property
// Map.MapScreenHeight Property
void CSampleProjectView::PostitionLegend(CMapXLegend& LegendMove) {
    try {
        // Place the legend at the bottom left corner of the screen
        LegendMove.SetTop(m_Map.GetMapScreenHeight() -
LegendMove.GetHeight());
        LegendMove.SetLeft(m_Map.GetMapScreenWidth() - LegendMove.GetWidth());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

```

    }
}

```

Пример на Visual Basic

```

Private Sub btnMapScreenHeightWidth_Click()

    Dim sHeight As Single

    Dim sWidth As Single

    sHeight = Map1.MapScreenHeight

    sWidth = Map1.MapScreenWidth

End Sub

```

Свойство Map.MapScreenWidth (Объект Map)

Назначение

Свойства MapScreenHeight и MapScreenWidth облегчают позиционирование легенд. Эти свойства содержат ширину и высоту окна карты в пикселах. Поскольку легенды тоже содержат их ширину и высоту, это облегчает позиционирование легенд.

Пример

```

Private Sub btnMapScreenHeightWidth_Click()

    Dim sHeight As Single

    Dim sWidth As Single

    sHeight = Map1.MapScreenHeight

    sWidth = Map1.MapScreenWidth

```

Свойство Map.MatchNumericFields (Объект Map)

Назначение

Это свойство определяет сравнение числовых полей. Это свойство появляется в Data (во время отладки) property page. Если MatchNumericFields это True, то числовые поля рассматриваются на предмет ключей (candidate keys) (по порядку номеров полей) где MapX производит автоматическое сравнение. Если это False, то числовые колонки не рассматриваются при автоматическом сравнении. По умолчанию это свойств - False.

Если Вы указываете числовое GeoField в Datasets.Add, MapX будет использовать это свойство не зависимо от того, MatchNumericFields TRUE или нет.

Свойство Map.MatchThreshold (Объект Map)

Назначение

Это минимально допустимый процент для сравнения слоя карты с исходными данными.

По умолчанию = 80.

Свойство Map.MaxSearchTime (Объект Map)

Назначение

Устанавливает время поиска для автоматического определения GeoColumn в GeoSet когда оно происходит в данных с использованием DataSets.Add. Это целая величина определяет время в секундах. По умолчанию время равно 5 секундам.

Свойство Map.MouseIcon (Объект Map)

Назначение

Если Вам надо использовать собственный курсор для мышки со свойством Map.MousePointer, используйте свойство Map.MouseIcon. Свойство Map.MouseIcon позволяет программисту указать стиль курсор для использования в качестве основного (по умолчанию) курсора в MapX.

Пример на Visual Basic

```
Map1.MousePointer = miCustomCursor

Map1.MouseIcon = "c:\windows\cursors\globe.ani"
```

Примечание

Строка, которая указывает имя файла или пользовательский курсор для мыши. Имя должно иметь тип файла .cur или .ani. Если указанный курсор не может быть загружен, то ничего не произойдет.

Смотрите также

Свойство Map.MousePointer

Свойство Map.MousePointer (Объект Map)

Назначение

Устанавливает форму для курсора мыши. Берется значение CursorConstants. По умолчанию это miDefaultCursor, которая означает, какой курсор в текущем инструменте будет использоваться.

Пример на C++

```
// Map.MousePointer Property

void CSampleProjectView::DatasetAdd() {
    try {
        m_Map.SetMousePointer(miHourglassCursor);

        AddMyODBCDataset(); // Embark on a lengthy operation...

        m_Map.SetMousePointer(miDefaultCursor);
    } catch (COleDispatchException *e) {
        e->ReportError();
    }
}
```

```

        e->Delete();
    } catch (CancellationToken *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub DatasetAdd_Click()
    Map1.MousePointer = miHourglassCursor
    ' Embark on a lengthy operation...
    AddMyODBCDataset
    Map1.MousePointer = miDefaultCursor
End Sub

```

Свойство Map.MouseWheelSupport (Объект Map)

Назначение

Определяет уровень поддержки Intellimouse: None, только Zoom/Scroll (без Autscroll) или Full. Это свойство устанавливается с MouseWheelSupportConstants.

Смотрите также

MouseWheelSupportConstants

Свойство Map.NumericCoordSys (Объект Map)

Назначение

Возвращает объект CoordSys доступный и для чтения и для записи (система координат) который определяет как MapX интерпретирует числовые координаты карты.

Примечание

NumericCoordSys представляет систему координат используемых для создания числовых координат карты. По умолчанию система координат это Широта/Долгота WGS-84. При изменении NumericCoordSys (методом CoordSys.Set), Вы можете работать с координатами карты в другой системе координат.

Свойство NumericCoordSys не действует на появление карты. Что бы установить систему координат (или "проекции") в которой отображается карта, используйте свойство DisplayCoordSys.

Метод Map.Pan (Объект Map)

Назначение

Этот метод перемещает карту на заданное число в экранных координатах(пикселях)

Синтаксис

ОБЪЕКТ.Pan (ScreenX, ScreenY)

Части	Описание
ОБЪЕКТ	Объект Map
ScreenX	Значение перемещения по горизонтали в экранных координатах.
ScreenY	Значение перемещения по вертикали в экранных координатах

Свойство Map.PanAnimationLayer (Объект Map)

Назначение

Это свойство определяет, является ли слой анимации (если он есть) прорисованным в резервной копии (backing store) когда используется инструмент перемещения, так что содержимое слоя захватывается вместе с остатком изображения карты. По умолчанию FALSE.

Пример

```
' Set up the pan tool so that the animation layer is seen when the user
pans the map

Map1.PanAnimationLayer = TRUE

Map1.CurrentTool = miPanTool
```

Свойство Map.PaperUnit (Объект Map)**Назначение**

Устанавливает единицы измерения, которые MapX будет использовать для отчета. Укажите единицы измерения используя константы PaperUnits. По умолчанию это miUnitInch.

Свойство Map.PreferCompactLegends (Объект Map)**Назначение**

Устанавливает отображается или нет компактная легенда. Логическое значение. По умолчанию это false.

Примечание

Когда легенды созданы, они могут быть компактны или полноразмерны. Когда создается тематическая карта, легенда всегда показывается. Эта установка определяет как будет показана легенда. Легенда всегда может быть изменена впоследствии использованием установок Theme.Legend.Compact или Theme.Legend.Visible, но начальные установки определяют расположение легенды.

Пример на Visual Basic

```
Private Sub Form_Load()

    Dim db As Database

    Dim rs As Recordset

    Dim ds As Dataset
```

```

Map1.PreferCompactLegends = True

Set db = DBEngine.Workspaces(0).OpenDatabase( _
"C:\Program Files\MapInfo MapX\Data\mapstats.mdb")

Set rs = db.OpenRecordset("USA")

Set ds = Map1.Datasets.Add(miDataSetDAO, rs, , , "USA")

ds.Themes.Add

End Sub

```

Пример на C++

```

void CSampleProjectView::AddTheme(CDaoRecordset& SourceRecordset) {
    CMapXDataset ds;
    COleVariant Recordset;
    COptionalVariant optVt;
    Recordset.vt = VT_DISPATCH;
    Recordset.pdispVal = SourceRecordset.m_pDAORecordset;

    try {
        m_Map.SetPreferCompactLegends(TRUE);

        ds = m_Map.GetDatasets().Add(miDataSetDAO, Recordset, optVt,
        optVt, optVt, "USA");

        ds.GetThemes().Add();
    } catch (COleDispatchException *e) {
        e->ReportError();

        e->Delete();
    } catch (COleException *e) {
        e->ReportError();

        e->Delete();
    }
}

```



```

    }
}

```

Метод Map.PrintMap (Объект Map)

Назначение

Печатает карту в указанном прямоугольнике.

Синтаксис

ОБЪЕКТ.PrintMap (hDC x, y, w, h)

Часть Описание

ОБЪЕКТ	Объект Map.
hDC	Устройство принтера.
x	Верхний левый угол X в единицах HIMETRIC.
y	Верхний левый угол Y в единицах HIMETRIC.
w	Ширина в единицах HIMETRIC.
h	Высота в единицах HIMETRIC.

Примечание

Текущая карта рисуется, заполняя данный прямоугольник. Лучший результат достигается, когда выдерживается пропорция ширина/высота.

Если объекты на карте выбраны, Вы можете захотеть печатать без выделенных объектов; смотрите свойство Map.ExportSelection.

Пример на Visual Basic

```

Private Sub Command4_Click()
    On Error GoTo ErrorHandler ` Set up error handler.
    ` coords must be in himetric
    ` print same size as on screen, in upper left of page
    ScaleMode = 6 `set mode to mm
    ` there is no Printer.StartDoc method, it seems it is done

```

```

` implicitly when you use one of the printer.print methods
` so we need to print something before we print our map
` to start the page
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Print " "
Map1.PrintMap Printer.hDC, 0, 0, Map1.Width * 100, _
Map1.Height * 100
Printer.NewPage ` Send new page.

Printer.EndDoc ` Printing is finished.
Exit Sub

```

Пример на C++

```

// Map.PaperUnit Property
// Map.PrintMap Method
void CSampleProjectView::OnPrintMap(CDC* pDC,CPrintInfo* pInfo) {
    try {
        // get paper width in mm and convert to HIMETRIC (100th of a mm)
        m_Map.SetPaperUnit(miUnitMillimeter);

        double pw = m_Map.GetMapPaperWidth() * 100;
        double ph = m_Map.GetMapPaperHeight()* 100;

        m_Map.PrintMap((long)pDC->m_hDC,
pInfo->m_rectDraw.left,pInfo->m_rectDraw.
top,(long)pw,(long)ph);
    }
}

```

```

    } catch (CleDbDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (CleDbException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Метод Map.PropertyPage (Объект Map)

Назначение

Метод PropertyPage позволяет пользователю контролировать или устанавливать различные свойства объектов во время работы программы. Это такое же свойство страницы (property page) которое возможно в режиме отладки, хотя некоторые свойства не могут быть установлены этим методом во время работы приложения.

Синтаксис

ОБЪЕКТ.PropertyPage

Свойство Map.RedrawInterval (Объект Map)

Назначение

Это свойство позволяет определять свойство прорисовки карты по растру. Это время, 1/100 часть секунды, между двумя прорисовками. Минимальное значение 10. Наибольшее значение (3000) приводит к тому, что экран не обновляется, пока карта полностью не появится на экране.

Примечание

Карта прорисовывается поверх выключенной прорисовки растра, затем прорисовывается, когда растр уже прорисован. Вы можете прорисовывать карту так часто, как надо, что бы показать пользователю, что произошло некоторое событие. (Это особенно полезно, когда прорисовываются улицы в 25 мильной зоне.)

Метод Map.Refresh (Объект Map)**Назначение**

Приводит к немедленному перерисовыванию карты.

Синтаксис

ОБЪЕКТ.Refresh

Примечание

Обычно используется, когда предпринято действие, которое приводит к перерисовке карты. MapX обычно не перерисовывает, пока полностью не получена информация, так что, есть возможность объединить несколько текущих изменений в одну прорисовку. Этот метод заставляет MapX прорисовывать немедленно, без ожидания следующего дополнения в изображении.

Пример на Visual Basic

```
Map1.BackColor = miColorYellow  
Map1.Refresh
```

Пример на C++

```
void CSampleProjectView::Recolor() {
```

```

try {
    m_Map.SetBackColor(miColorYellow);
    m_Map.Refresh();
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Смотрите также

Свойство Map.AutoRedraw

Свойство Map.Rotation (Объект Map)

Назначение

Устанавливает вращение карты. Величины вращения определяются в градусах, увеличиваясь по часовой стрелке, 0 показывает невращаемую карту. Применение вращения к карте не меняет масштаб или центральную точку.

Внимание: Карта с растровыми слоями вращаться не может.

Метод Map.SaveMapAsGeoset (Объект Map)

Назначение

Создает файл geoset представляющий текущий статус карты (информация о слоях, подписи, и др.).

Синтаксис

ОБЪЕКТ.SaveMapAsGeoset (name, filespec)

Часть	Описание
ОБЪЕКТ	Объект Map.
name	Строка, представляющая дружественное имя geoset's ("friendly" name) (описание, появляющееся в Geodictionary). Если Вы указываете пустой строинг, используется Title.
filespec	Строка, представляющая полный путь к файлу geoset. Если путь не указан, новый файл помещается в папку, где данные. Если расширение не указано, используется стандартное расширение .gst. Если указанный файл уже существует, он переписывается.

Примечание

Если карта содержит временные слои, то они не попадут в новый файл geoset, кроме того варианта, если они специально сохранены. Этот метод не действует, если нет открытых слоев.

Пример

```
Map1.SaveMapAsGeoset "My latest map", "C:\Data\Most Recent.GST"
```

Свойство Map.SearchPath (Объект Map)

Назначение

Это свойство позволяет пользователю динамически устанавливать SearchPath. Это свойство возможно только в режиме работы программы. SearchPath изначально считывается из registry.

Пример на Visual Basic

```
Private Sub btnSearchPath_Click()  
    Dim str As String  
  
    ' Get the default search path from the registry
```

```

    str = Map1.SearchPath
    '   Update the search path
    Map1.SearchPath = "d:\junk"

End Sub

```

Пример на C++

```

void CSampleProjectView::SetSearchPath() {
    CString oldPath,msg;

    try {
        // Get the default search path
        oldPath = m_Map.GetSearchPath();

        // Update the search path
        m_Map.SetSearchPath("d:\\junk");
    } catch (COleDispatchException *e) {
        e->ReportError();

        e->Delete();
    } catch (COleException *e) {
        e->ReportError();

        e->Delete();
    }

    msg.Format("Changed the search path from \n\"%s\"\\n to\n\"d:\\junk\\",oldPath);

    AfxMessageBox(msg);
}

```

Свойство Map.SelectionStyle (Объект Map)

Назначение

Возвращает объект Style, представляющий стиль, в котором нарисованы выбранные объекты.

Примечание

Это свойство возвращает объект Style, доступный для чтения и записи, который определяет внешний вид выбранных объектов во всех слоях.

Чтобы контролировать внешний вид выбранных полигонов, установите свойства полигонов этого объекта Style, такие как RegionColor. Свойства RegionColor и RegionBorderXXX используются для прорисовки прямоугольника вокруг выбранных символьных объектов. Свойство LineXXX используется для прорисовки выбранных линейных объектов. Обратите внимание, что свойства SymbolXXX и TextXXX не применяются и действия не оказывают.

Пример

```
'Change color of all selected features to blue
Map1.SelectionStyle.RegionColor = miColorBlue
Map1.SelectionStyle.LineColor = miColorBlue
```

Смотрите также

Что бы избежать печати выделенных объектов, установите свойство Map.ExportSelection в False.

Метод Map.SetSize (Объект Map)

Назначение

Устанавливает размеры карты, в пикселах.

Синтаксис

ОБЪЕКТ.SetSize (W, H)

Часть **Описание**

ОБЪЕКТ Объект Map.

W Желаемая ширина (width of the control), в пикселах.

H Желаемая высота (height of the control) в пикселах.

Примечание

В большинстве приложений Visual Basic, Вы можете изменить размер окна без использования метода `SetSize` — только используя стандартные свойства `Width` и `Height` поддерживаемые VB. Таким образом, если Вы создали объект `Map` вызывая функции `CreateObject()`, Вам необходимо использовать `SetSize` для контроля над размером карты.

Свойство `Map.Title` (Объект `Map`)

Назначение

Объект `Title` определяющий заголовок карты. Это может быть установлено на `NULL` что бы обойтись без заголовка.

Используйте объект `Title` для контроля текста заголовка, отображения стиля и позиции.

Замечание: Если Вы хотите установить текст заголовка, или Вы не следите за его стилем и позицией, то Вы можете установить свойство `Map.TitleText` во время разработки программы.

Свойство `Map.TitleText` (Объект `Map`)

Назначение

Возвращает или устанавливает строку, которая является заголовком карты.

Примечание

По умолчанию заголовок это имя `GeoSet`. Если `TitleText` равен имени `GeoSet`, а рабочий набор `Geoset` изменился, `TitleText` также меняется.

Пример

```
Map1.TitleText = "Sales Forecast"
```

Смотрите также

Свойство `Map.Title`

Свойство Map.Version (Объект Map)

Назначение

Указывает используемую версию MapX. Это строковая величина, доступная только для чтения.

Свойство Map.WaitCursorEnabled (Объект Map)

Назначение

Это свойство позволяет включать и выключать задержку реакции курсора вручную (Wait Cursor). По умолчанию True.

Свойство Map.Zoom (Объект Map)

Назначение

Содержит или устанавливает значение текущего масштаба карты. Значение масштаба определяется как ширина карты. Это величина двойной точности в единицах, указанных в свойстве MapUnit.

Пример

```
Dim dZom as Double
dZoom = Map.Zoom    'Reads the map's current zoom.
- или -
Map.Zoom = dZoom    ' Sets the map's current zoom to dZoom
```

Метод Map.ZoomTo (Объект Map)

Назначение

Определяет масштаб и положение центра карты.

Синтаксис

ОБЪЕКТ.ZoomTo (Zoom, x, y)

Часть Описание

ОБЪЕКТ Объект Map.

Zoom Значение масштаба. MapX определяет значение масштаба как ширину карты. Это величина двойной точности, определенная в терминах единиц карты(MapUnit).

X Координата x центра карты. Двойная точность, отображает долготу.

Y Координата y центра карты. Двойная точность, отображает широту.

Примечание

Используйте этот метод как простой путь для изменения масштаба и центра карты. Другими словами, будут установлены свойства Zoom, CenterX и CenterY объекта Map но что бы избежать троекратного перерисовывания экрана, используйте свойство Map.AutoRedraw.

Пример на Visual Basic

```
`This sample demonstrates the use of the Map.ZoomTo method
`It zooms in on New England on the US Map.
Private Sub Command1_Click()
    Debug.Print "CenterX = " & Map1.CenterX
    Debug.Print "CenterY = " & Map1.CenterY
    Debug.Print "    Zoom = " & Map1.Zoom
    Map1.ZoomTo 800, -70.26, 44.05 `Zoom in on New England
    Debug.Print "CenterX = " & Map1.CenterX
    Debug.Print "CenterY = " & Map1.CenterY
    Debug.Print "    Zoom = " & Map1.Zoom
End Sub
```

Пример на C++

```
void CSampleProjectView::Zoom() {
```

```

CString msg;
try {
    msg.Format("CenterX = %lf\nCenterY = %lf\n    Zoom = %lf",m_Map.GetCenterX(),m_Map.GetCenterY(),m_Map.GetZoom());
    AfxMessageBox(msg);
    m_Map.ZoomTo(800,-70.26,44.05); // Zoom in on New England
    msg.Format("CenterX = %lf\nCenterY = %lf\n    Zoom = %lf",m_Map.GetCenterX(),m_Map.GetCenterY(),m_Map.GetZoom());
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
AfxMessageBox(msg);
}

```

Объект MultivarCategory и коллекция MultivarCategories

Круговая или столбчатая тематическая карта связывается через OLE с коллекцией MultivarCategories, которая является коллекцией объектов MultivarCategory. Коллекция содержит один объект для каждого столбца в столбчатой диаграмме, или один объект для каждого сектора в круговой диаграмме.

Для получения коллекции MultivarCategories, обратитесь к свойству ThemeProperties.MultivarCategories.

Свойства объекта

- Свойство MultivarCategory.Style

Свойства коллекции

- Свойство MultivarCategories.Count
- Свойство MultivarCategories.Item

Пример на C++

```
// MultivarCategories Object
void CSampleProjectView::ChangeChart(CMapXTheme& InTheme) {
    try {
        switch(InTheme.GetType()) {
            case miThemeBarChart:
                // No break statement: will fall through
            case miThemePieChart:
                // The theme is a multivariate theme
                // If there is a pie or bar chart with two slices/bars,
                // change the
                // colors to Black and White
                CMapXMultivarCategories MCategories;
                MCategories =
                InTheme.GetThemeProperties().GetMultivarCategories();
                if(MCategories.GetCount()==2) {
                    MCategories.Item(1).GetStyle().SetRegionColor(miColorBlack);
                    MCategories.Item(2).GetStyle().SetRegionColor(miColorWhite);
                }
                break;
            case miThemeRanged:
                // ...
                break;
        }
    }
```

```

    } catch (ColeDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (ColeException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub ChangeChart(InTheme as MapXLib.Theme)
    Select Case InTheme.Type
        Case miThemeBarChart, miThemePieChart
            ' The theme is a multivariate theme
            ' If there is a pie or bar chart with two slices/bars,
            ' change the
            ' colors to Black and White
            If InTheme.ThemeProperties.MultivarCategories.Count = 2 Then
                InTheme.ThemeProperties.MultivarCategories(1).Style. _
                    RegionColor = miColorBlack
                InTheme.ThemeProperties.MultivarCategories(2). _
                    Style.RegionColor = miColorWhite
            End If
        Case miThemeRanged
            ' ...
    End Select
End Sub

```

Смотрите также

Свойство `ThemeProperties.MultivarCategories`

Объект `LegendText` и коллекция `LegendTexts`

Свойство `MultivarCategories.Count` (Коллекция `MultivarCategories`)**Назначение**

Целое значение, доступное только для чтения, указывающее на количество переменных, используемых в тематической карте— количество столбцов в столбчатой тематической карте или количество секторов в карте с круговыми диаграммами.

Свойство `MultivarCategories.Item` (Коллекция `MultivarCategories`)**Назначение**

Возвращает один объект `MultivarCategory` из коллекции; этот объект описывает настройки для одной из тематических переменных (например, один столбец для столбчатой диаграммы).

Свойство `MultivarCategory.Style` (Объект `MultivarCategory`)**Назначение**

Свойство определяет стиль для категории. Можно установить стиль существующего объекта по умолчанию, или свойства стиля могут быть установлены индивидуально. По умолчанию устанавливаются высококонтрастные стили оформления.

Смотрите также

Объект `Style`

Объект NotesQueryInfo

Вы можете осуществлять запросы, основываясь на наборе данных Notes dataset при создании объекта NotesQueryInfo и использовании набора данных 8-ого типа (dataset type 8) - это - (miDataSetNotesQuery).

Этот тип набора данных DataSet позволяет пользователю MapX запрашивать замечания (Notes) для создания запросов к базе данных во время работы программы. Для добавления набора данных этого типа к MapX, Вам надо создать объект NotesQueryInfo и передать его в DataSets.Add.

Свойства объекта

- Свойство NotesQueryInfo.BeginDate (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.Database (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.DefaultNumericValue (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.DefaultStringValue (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.EndDate (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.FullTextSearch (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.MaxNumDocs (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.Query (Объект NotesQueryInfo)
- Свойство NotesQueryInfo.Server (Объект NotesQueryInfo)

Свойство NotesQueryInfo.BeginDate (Объект NotesQueryInfo)

Назначение

Это строковая величина, которая определяет начало даты(времени) отключения или строковую величину, представляющую такую дату (добавочная). Замечания измененные перед этой датой, игнорируются во время поиска.

Свойство `NotesQueryInfo.Database` (Объект `NotesQueryInfo`)

Назначение

Это имя базы данных Notes на сервере, или путь к локальной базе данных Notes (.nsf).

Свойство `NotesQueryInfo.DefaultNumericValue` (Объект `NotesQueryInfo`)

Назначение

Потому что Notes Query может возвращать документы, которые не содержат все указанные поля, MapX вернет по умолчанию число двойной точности, если запрашиваемое числовое поле не найдено (необязательная величина). Эта величина по умолчанию равна 0.

Смотрите также

`DataSets.Add`

Свойство `NotesQueryInfo.DefaultStringValue` (Объект `NotesQueryInfo`)

Назначение

Поскольку Notes Query может вернуть документы, которые не содержат все указанные поля, MapX вернет по умолчанию строковую величину, если запрашиваемое строковое поле не найдено (не обязательная величина). Эта величина по умолчанию равна "".

Свойство `NotesQueryInfo.EndDate` (Объект `NotesQueryInfo`)

Назначение

Строковая величина, которая определяет время, такое, что примечания (Notes) измененные перед этим временем, игнорируются при поиске.

Свойство NotesQueryInfo.FullTextSearch (Объект NotesQueryInfo)

Назначение

Это логическая величина (TRUE или FALSE) которая определяет, какая база данных будет иметь полные текстовые индексы и что свойство Query это полнотекстовый запрос поиска.

Смотрите также

Свойство NotesQueryInfo.Query

Свойство NotesQueryInfo.MaxNumDocs (Объект NotesQueryInfo)

Назначение

Целая величина, которая определяет минимальное количество примечаний (Notes) позволенных в результате (необязательное значение). Если введен 0, то будут собраны все примечания, которые попадают под данный критерий поиска.

Свойство NotesQueryInfo.Query (Объект NotesQueryInfo)

Назначение

Если свойство FullTextSearch установлено на TRUE, это свойство является полнотекстовым запросом на поиск. Если свойство FullTextSearch это FALSE, то это свойство становится формулой поиска Notes. Смотрите электронную справочную систему, чтобы узнать как формировать полнотекстовый запрос на поиск или строку с формулой.

Смотрите также

Свойство NotesQueryInfo.FullTextSearch

Свойство `NotesQueryInfo.Server` (Объект `NotesQueryInfo`)

Назначение

Это строковая величина, определяющая какой сервер содержит базу данных Notes. Эта строковая величина оставляется пустой, если свойство `Database` ссылается на локальную базу данных.

Смотрите также

Свойство `NotesQueryInfo.Database`

Объект `NotesViewInfo`

Объект `NotesViewInfo` требуется для второго параметра к `Datasets.Add` для наборов данных типа `miDataSetNotesView`. Этот тип `DataSet` будет принимать определенный вид базы данных `Database`.

Свойства объекта

- Свойство `NotesViewInfo.Database` (Объект `NotesViewInfo`)
- `NotesViewInfo.Server` (Объект `NotesViewInfo`)
- `NotesViewInfo.View` (Объект `NotesViewInfo`)

Смотрите также

Метод `DataSets.Add`

Свойство `NotesViewInfo.Database` (Объект `NotesViewInfo`)

Назначение

Это свойство для чтения/записи. Определяет имя базы данных Notes на сервере, или путь к локальной базе (.nsf).

NotesViewInfo.Server (Объект NotesViewInfo)

Назначение

Это свойство для чтения/записи - строка, определяющая какой сервер Notes, располагающийся выше базы данных, является действующим. Строка остается пустой, если свойство Database ссылается на локальную базу данных.

NotesViewInfo.View (Объект NotesViewInfo)

Назначение

Это свойство для чтения/записи - строка, которая ссылается на определенный вид внутри базы данных из которой будут поддерживаться данные.

Объект ODBCQueryInfo

Объект ODBCQueryInfo передается в метод DataSets.Add как параметр источника данных, если тип набора данных DataSet является miDataSetODBC. Этот объект содержит информацию о том как связаться с источником данных ODBC и какие записи относятся к установкам записи. Внимание: Если Вы используете объект ODBCQueryInfo, Ваш проект должен ссылаться на MODBCDataset.DLL ("MapInfo ODBC Dataset Engine Library") и MMapXColumnInfo.dll.

В Visual Basic, выполните команду Tools > References (для VB 4) или Project > References (для VB 5). Нажмите кнопку Browse и добавьте ссылку к соответствующей библиотеке dll. Библиотеки dll установятся в директорию MapX Program.

Свойства объекта

- Свойство ODBCQueryInfo.ConnectionString (Объект ODBCQueryInfo)
- Свойство ODBCQueryInfo.DataSource (Объект ODBCQueryInfo)
- Свойство ODBCQueryInfo.SqlQuery (Объект ODBCQueryInfo)

Пример на Visual Basic

```
Private Sub Command2_Click()
    Dim ds As Dataset
    Dim parm As New ODBCQueryInfo
    parm.SqlQuery = "select * from usa"
    parm.DataSource = "usa    "' name of odbc datasource
    parm.ConnectionString = "ODBC;          "' can be left blank, or can
                                'include user= or pwd=
    Set ds = Map.Datasets.Add(miDataSetODBC, parm, "odbc Dataset")
    For Each fd In ds.Fields
        MsgBox fd.Name
    Next
    If Not ds Is Nothing Then ds.Themes.Add
End Sub
```

Пример на C++

```
void CSampleProjectView::ODBCDatasetAdd() {
    CString msg("Fields:\n\t");
    CMapXDataset ds;
    COleVariant parmVariant;
    CMapXODBCQueryInfo parm;
    long i,FieldCount;
    parm.CreateDispatch(parm.GetClsid());
    try {
        parm.SetSqlQuery("select * from usa");
        parm.SetDataSource("usa"); // name of odbc datasource
```

```
    parm.SetConnectionString("ODBC;"); // Can be left blank, or can include
    user= or pwd=
```

```
    parmVariant.vt = VT_DISPATCH;
    parmVariant.pdispVal = parm.m_lpDispatch;
```

```
    ds = m_Map.GetDatasets().Add(miDataSetODBC,parmVariant,"odbc
Dataset");
```

```
    FieldCount = ds.GetFields().GetCount();
```

```
    for(i=1;i<=FieldCount;i++) {
```

```
        msg += ds.GetFields().Item(i).GetName();
```

```
        msg += '\n\t';
```

```
    }
```

```
    AfxMessageBox(msg);
```

```
    if(FieldCount!=0)
```

```
        ds.GetThemes().Add();
```

```
} catch (COleDispatchException *e) {
```

```
    e->ReportError();
```

```
    e->Delete();
```

```
} catch (COleException *e) {
```

```
    e->ReportError();
```

```
    e->Delete();
```

```
}
```

```
}
```

```
// ODBCQueryInfo.ConnectionString Property (Page 150)
```

```
void CSampleProjectView::ODBCDatasetAdd() {
```

```

CString msg("Fields:\n\t");
CMapXDataset ds;
COleVariant parmVariant;
CMapXODBCQueryInfo parm;
long i,FieldCount;
parm.CreateDispatch(parm.GetClsid());
try {
    parm.SetSqlQuery("select * from usa");
    parm.SetDataSource("usa"); // name of odbc datasource
    parm.SetConnectionString("ODBC;"); // Can be left blank, or can include
user= or pwd=

    parmVariant.vt = VT_DISPATCH;
    parmVariant.pdispVal = parm.m_lpDispatch;

    ds = m_Map.GetDatasets().Add(miDataSetODBC,parmVariant,"odbc
Dataset");

    FieldCount = ds.GetFields().GetCount();
    for(i=1;i<=FieldCount;i++) {
        msg += ds.GetFields().Item(i).GetName();
        msg += '\n\t';
    }
    AfxMessageBox(msg);
    if(FieldCount!=0)
        ds.GetThemes().Add();
} catch (COleDispatchException *e) {
    e->ReportError();
}

```

```

        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Смотрите также

Метод DataSets.Add

Свойство ODBCQueryInfo.ConnectionString (Объект ODBCQueryInfo)

Назначение

Устанавливает строку подключения к источнику данных ODBC. Например, укажите “ODBC;”. Вы также можете включить “user=”, “pwd=” или DLG=. Если информация пропущена, а она необходима, появится соответствующая подсказка.

“DLG=” определяет появление диалога настройки коннекта: DLG=0 означает никогда не размещать диалог, DLG=1 означает всегда размещать диалог, DLG=2 означает показ диалога, только когда он необходим.

Пример на Visual Basic

```

Private Sub Command2_Click()
    Dim ds As Dataset
    Dim parm As New ODBCQueryInfo
    parm.SqlQuery = "select * from usa"
    parm.DataSource = "usa    '' name of odbc datasource
    ' uid= or pwd= or DLG=
    Set ds = Map.DataSets.Add(miDatasetODBC, parm, "odbc Dataset")

```



```

For Each fd In ds.Fields
    MsgBox fd.Name
Next
If Not ds Is Nothing Then ds.Themes.Add
End Sub

```

Пример на C++

```

// ODBCQueryInfo Object
// BindОбъект Layer
// Fields Object
void CSampleProjectView::CreateXYTable() {
{
    CMapXODBCQueryInfo QueryInfo;
    COleVariant BindLayerVt,FieldsVt,QueryInfoVt;
    CMapXFields Fields;
    CMapXBindLayer BindLayer;

    // Create new instances of the Fields, ODBCQueryInfo, and BindОбъект
Layers
    if(!Fields.CreateDispatch(Fields.GetClsid())) {
        TRACE0("Failed to Create Fields Object");
        return;
    }
    if(!QueryInfo.CreateDispatch(QueryInfo.GetClsid())) {
        TRACE0("Failed to Create ODBCQueryInfo Object!");
        return;
    }
}
}

```

```

if(!BindLayer.CreateDispatch(BindLayer.GetClsid())) {
    TRACE0("Failed to Create BindОбъект Layer!");
    return;
}

try {
    // You can specify a username and password in the Connect String
    QueryInfo.SetConnectionString("ODBC;");
    // Not specifying a datasource will pop up a datasource dialog
    // If the data source name is known in advance, you can avoid
    relying on
    // the user to find the Mapstats.mdb database
    QueryInfo.SetDataSource("");
    // The US_CUST recordset is in the Mapstats.mdb database
    // Note: It is possible to only select the fields you want
    // For example: QueryInfo.SetSqlQuery("select COMPANY,X,Y,ORDER_AMT
    from US_CUST");
    // Or other more complex SQL queries
    QueryInfo.SetSqlQuery("select * from US_CUST");

    QueryInfoVt.vt = VT_DISPATCH;
    QueryInfoVt.pdispVal = QueryInfo;

    // Create a permanent layer in the MapX directory (if the layer
    already exists,
    // the user will get an error in Datasets.Add)
    // This layer will have one column, which is the GeoField specified
    in Datasets.Add

```

```

// In this case, it is the company name
BindLayer.SetFilespec("C:\\Program Files\\MapInfo
MapX\\Maps\\CustLayer.tab");

BindLayer.SetLayerName("Customers");

BindLayer.SetLayerType(miBindLayerTypeXY);


// For layers of type miBindLayerTypeXY, RefColumn1 must be the
column that contains

// X coordinates, and RefColumn2 must be the column that contains Y
coordinates

BindLayer.SetRefColumn1("X");

BindLayer.SetRefColumn2("Y");


BindLayerVt.vt = VT_DISPATCH;

BindLayerVt.pdispVal = BindLayer;


// This specifies which fields from the source data to add and how
to add them

Fields.Add(ColeVariant("COMPANY"),ColeVariant("Company"),ColeVa
riant((long)miAggregationIndividual),ColeVariant((long)miTypeSt ring));

Fields.Add(ColeVariant("X"),ColeVariant("X"),ColeVariant((long)
miAggregationAverage),ColeVariant((long)miTypeNumeric));

Fields.Add(ColeVariant("Y"),ColeVariant("Y"),ColeVariant((long)
miAggregationAverage),ColeVariant((long)miTypeNumeric));

Fields.Add(ColeVariant("ORDER_AMT"),ColeVariant("Order_Amt"),CO
leVariant((long)miAggregationSum),ColeVariant((long)miTypeNumer ic));


FieldsVt.vt = VT_DISPATCH;

FieldsVt.pdispVal = Fields;

```

```

        // The Company Field is the used as the unique GeoField

        m_Map.GetDatasets().Add(miDataSetODBC,QueryInfoVt,ColeVariant("
Customers
Data"),ColeVariant("Company"),COptionalVariant(),BindLayerVt,Fi
eldsVt,COptionalVariant());

        } catch (COleDispatchException *e) {

            e->ReportError();

            e->Delete();

        } catch (COleException *e) {

            e->ReportError();

            e->Delete();

        }

    }
}

```

Свойство ODBCQueryInfo.DataSource (Объект ODBCQueryInfo)

Назначение

Устанавливает имя источника данных ODBC. Если это свойство оставляется пустым, ODBC запросит источник данных. Источник данных устанавливается используя администратор ODBC, который можно найти в Windows95 или Windows NT Control Panel.

Свойство ODBCQueryInfo.SqlQuery (Объект ODBCQueryInfo)

Назначение

Определяет строку SQL-запроса, который используется для выбора столбцов и строк при создании набора данных.

Например, укажите “select * from USA”.

Коллекция Parts

Объект Parts это коллекция объекта коллекции Points. Коллекция имеет методы Count, Item и Remove. Метод Add берет исходный объект Points и возвращает объект Points, который добавляется к коллекции Parts (то есть, новый объект Points добавляется к коллекции Parts и точки копируются из исходной коллекции Points).

Свойства коллекции

- Свойство Parts.Count
- Свойство Parts.Item

Методы коллекции

- Метод Parts.Add
- Метод Parts.Remove
- Метод Parts.RemoveAll

Пример для parts, points и point на Visual Basic

```
' create a polyline feature with 3 points
Dim newobj as new MapXLib.Feature      'standalone object
Dim obj as MapXLib.Feature             'to hold object added to layer
Dim pts as new Points
Dim pt as new Point
newobj.Type = miFeatureTypeLine
'Use the map's default symbol style
newobj.Style = map1.DefaultStyle
'set the lines 3 points
pt.Set -101.023, 45.0452
pts.Add pt
pt.Set -102.023, 49.0452
pts.Add pt
```

```

pt.Set -100.0, 34.2564
pts.Add pt
newobj.Parts.Add pts
'now add the object to a layer
' obj will be the newly added object
set obj = map1.Layers(1).AddFeature(newobj)

```

Пример на C++

```

void CSampleProjectView::CreatePolyFeature() {
    // Create a polyline feature with 3 points
    CMapXFeature obj,newobj;
    CMapXPoints pts;
    CMapXPoint pt;
    if(!newobj.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to create Feature object");
        return;
    }
    if(!pts.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to create Points collection");
        return;
    }
    if(!pt.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to create Point object");
        return;
    }

    try {

```

```

newobj.SetType(miFeatureTypeLine);
newobj.SetStyle(m_Map.GetDefaultStyle()); // Use the _
map's default symbol style

pt.Set(-101.023,45.0452);
pts.Add(pt);
pt.Set(-102.023,49.0452);
pts.Add(pt);
pt.Set(-100.0,34.2564);
pts.Add(pt);
newobj.GetParts().Add(pts);
obj = m_Map.GetLayers().Item(1).AddFeature(newobj); _
// obj will be the newly added object
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Метод **Parts.Add** (Коллекция **Parts**)

Назначение

Метод Add берет исходный объект Points и возвращает объект Points добавленный к коллекции Parts (то есть новый объект Points добавляется к коллекции Parts и точки копируются из исходной коллекции Points).

Синтаксис

ОБЪЕКТ.PartsAdd(Points)

Pointst это объект коллекции Points который Вам надо добавить.

Свойство **Parts.Count** (Коллекция **Parts**)

Назначение

Количество объектов в коллекции.

Свойство **Parts.Item** (Коллекция **Parts**)

Назначение

Возвращает объект коллекции Points из коллекции. Это переменная величина и Вы можете указать имя точки или индексный номер, начиная с 1. Это свойство по умолчанию коллекции Parts.

Метод **Parts.Remove** (Коллекция **Parts**)

Назначение

Удаляет объект коллекции Points из коллекции.

Синтаксис

ОБЪЕКТ.Remove (index)

Метод **Parts.RemoveAll** (Коллекция **Parts**)

Назначение

Удаляет все объекты **Parts** из коллекции.

Синтаксис

ОБЪЕКТ.RemoveAll

Пример На Visual Basic

```
Private Sub btnPartsRemoveAll_Click()
    Dim newobj As New MapXLib.Feature 'stand-alone object
    Dim pts As New Points
    Dim pt As New Point
    Dim nPnts As Integer
    Dim nParts As Integer
    newobj.Attach Map1
    newobj.Type = miFeatureTypeLine
    ' Use the map's default symbol style
    newobj.Style = Map1.DefaultStyle
    'set the lines 3 points
    'Also could use pts.AddXY here
    pt.Set -101.023, 45.0452
    pts.Add pt
    pt.Set -102.023, 49.0452
    pts.Add pt
    pt.Set -100.012, 34.2564
    pts.Add pt
```

```

' Add the points into the parts collection
newobj.Parts.Add pts
nParts = newobj.Parts.Count
' remove all of the parts
newobj.Parts.RemoveAll
nParts = newobj.Parts.Count
End Sub

```

Пример на C++

```

void CSampleProjectView::RemoveAllPoints() {
    CMapXFeature newobj;
    CMapXPoints pts;
    CMapXPoint pt;
    if(!newobj.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to create Feature object");
        return;
    }
    if(!pts.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to create Points collection");
        return;
    }
    if(!pt.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to create Point object");
        return;
    }

    try {

```

```

newobj.Attach(m_Map.GetDispatch());
newobj.SetType(miFeatureTypeLine);
newobj.SetStyle(m_Map.GetDefaultStyle());
pt.Set(-101.023,45.0452);
pts.Add(pt);
pt.Set(-102.023,49.0452);
pts.Add(pt);
pt.Set(-100.012,34.2564);
pts.Add(pt);
// Add the points into the parts collection
newobj.GetParts().Add(pts);

// now, remove all of the parts
newobj.GetParts().RemoveAll();
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Объект и коллекция Point

Объект Point представляет пару координат XY. Когда Point принадлежит Feature, координата X представляет долготу, а Y - широту.

Свойства объекта

- Свойство Point.X (Объект Point)
- Свойство Point.Y (Объект Point)

Методы объекта

- Метод Point.Offset (Объект Point)
- Метод Point.Set (Объект Point)

Свойства коллекции

- Свойство Points.Count (Коллекция Points)
- Свойство Points.Item (Коллекция Points)

Методы коллекции

- Метод Points.Add (Коллекция Points)
- Метод Points.AddXY (Коллекция Points)
- Метод Points.Remove (Коллекция Points)
- Метод Points.RemoveAll (Коллекция Points)

Пример на Visual Basic

```
'Print out all points in first object of selection
Dim obj as MapXLib.Feature 'to hold object added to layer
Dim pts as Points
Dim pt as Point
set obj = map1.layers(1).selection(1)
for each pts in obj.parts
    for each pt in pts
        debug.print pt.x, pt.y
    next
next
```

Пример на C++

```

void CSampleProjectView::PrintSelectedPoints() {
    CMapXFeature obj;
    CMapXPoints pts;
    CMapXPoint pt;
    CString ptmsg,msg;
    long i,j,PointsCount,PointCount;
    try {
        obj = m_Map.GetLayers().Item(1).GetSelection().Item(1);
        PointsCount= obj.GetParts().GetCount();
        for(i=1;i<=PointsCount;i++) {
            pts= obj.GetParts().Item(i);
            PointCount=pts.GetCount();
            for(j=1;j<=PointCount;j++) {
                pt=pts.Item(i);
                ptmsg.Format("%d, %d",pt.GetX(),pt.GetY());
                msg += ptmsg;
            }
        }
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

```
AfxMessageBox(msg);  
}
```

Метод **Point.Offset** (Объект **Point**)

Назначение

Меняет положение точки при изменении ее координат.

Синтаксис

ОБЪЕКТ.Offset(double deltaX, double deltaY)

Части Описание

ОБЪЕКТ Объект Point.

deltaX это приращение по координате x.

deltaY это приращение по координате y.

double DeltaX и deltaY это величины в числовой системе координат карты.

Смотрите также

Map.Numeric CoordSys

Метод **Point.Set** (Объект **Point**)

Назначение

Устанавливает координаты точки.

Синтаксис

ОБЪЕКТ.Set(double X, double Y)

Части Описание

ОБЪЕКТ Объект Point.

X X это x - координата точки.

Y Y это y - координата точки.
double DeltaX и deltaY это величины в числовой системе координат карты.

Свойство Point.X (Объект Point)

Назначение

Координата X точки. X и Y это величины в системе координат карты.

Смотрите также

Map.NumericCoordSys

Свойство Point.Y (Объект Point)

Назначение

Координата Y точки. X и Y это величины в системе координат карты.

Смотрите также

Map.NumericCoordSys

Метод Points.Add (Коллекция Points)

Назначение

Добавляет точку к коллекции Point.

Синтаксис

[Point=]ОБЪЕКТ.Add(Point, [Position])

Части Описание

Объект Points.

ОБЪЕКТ

Point Point это точечный объект, который Вы хотите добавить.

Position Position это положение точки. Параметр Position не обязательный. Если он пропущен или положение больше, чем Points.Count, точка присоединяется в конец коллекции. 1 это первая позиция в списке.

Пример на C++

```
// Point.Set Method
// Points.Add Method
// Points.AddXY Method
// Parts.Add Method

void CSampleProjectView::FtrCreate() {
    CMapXFeature Ftr;
    CMapXPoints pts1,pts2;
    CMapXPoint pt;

    if(!Ftr.CreateDispatch(Ftr.GetClsid()) ||
    !pts1.CreateDispatch(pts1.GetClsid()) ||
        !pts2.CreateDispatch(pts2.GetClsid()) ||
    !pt.CreateDispatch(pt.GetClsid())) {
        TRACE0("Failed to Create object in CSampleProjectView::FtrCreate");
        return;
    }

    try {
        Ftr.Attach(m_Map.GetDispatch());
        Ftr.SetType(miFeatureTypeRegion);
```



```

// Make a new region feature with two triangles
pt.Set(m_Map.GetCenterX()+10.0,m_Map.GetCenterY());
pts1.Add(pt);
pt.Set(m_Map.GetCenterX()+10.0,m_Map.GetCenterY() + 5.0);
pts1.Add(pt);
pt.Set(m_Map.GetCenterX()+15.0,m_Map.GetCenterY());
pts1.Add(pt);

// This performs the same operation (with different coordinates)
// using the AddXY method
pts2.AddXY(m_Map.GetCenterX()-10.0,m_Map.GetCenterY());
pts2.AddXY(m_Map.GetCenterX()-10.0,m_Map.GetCenterY()-5.0);
pts2.AddXY(m_Map.GetCenterX()-15.0,m_Map.GetCenterY());

// Add the two triangles to the new region
Ftr.GetParts().Add(pts1);
Ftr.GetParts().Add(pts2);

// Add the region to a temporary layer
Map1.GetLayers().Item("Temp Layer").AddFeature(Ftr);
} catch (CleDbDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (CleDbException *e) {
    e->ReportError();

```

```

        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub FtrCreate_Click()
    Dim Ftr As New Feature
    Dim pts1 As New Points, pts2 As New Points
    Dim pt As New Point

    Ftr.Attach Map1
    Ftr.Type = miFeatureTypeRegion

    ' Make a new region feature with two triangles
    pt.Set Map1.CenterX + 10, Map1.CenterY
    pts1.Add pt
    pt.Set Map1.CenterX + 10, Map1.CenterY + 5
    pts1.Add pt
    pt.Set Map1.CenterX + 15, Map1.CenterY
    pts1.Add pt

    ' This performs the same operation (with different
    ' coordinates)using the AddXY method
    pts2.AddXY Map1.CenterX - 10, Map1.CenterY
    pts2.AddXY Map1.CenterX - 10, Map1.CenterY - 5
    pts2.AddXY Map1.CenterX - 15, Map1.CenterY

```

```

' Add the two triangles to the new region

Ftr.Parts.Add pts1

Ftr.Parts.Add pts2


' Add the region to a temporary layer

Map1.Layers.Item("Temp Layer").AddFeature Ftr

End Sub

```

Метод **Points.AddXY** (Коллекция **Points**)

Назначение

Этот метод создает новую точку с данными координатами X и Y и добавляет ее в коллекцию Points.

Синтаксис

```
[ Point= ]ОБЪЕКТ.AddXY (x, y, [Position])
```

Часть	Описание
ОБЪЕКТ	Объект Points.
x	Двойная точность CoordSys.
y	Двойная точность CoordSys.
Position	Переменная, указывающая положение точки в списке коллекции от 1 до Points.Count. Без указания точка попадет в конец коллекции.

Свойство **Points.Count** (Коллекция **Points**)

Назначение

Количество объектов в коллекции.

Свойство **Points.Item** (Коллекция **Points**)

Назначение

Возвращает объект **Point** из коллекции **Points**. Это переменная и Вы можете указать имя точки или ее индекс, начиная с 1.

Метод **Points.Remove** (Коллекция **Points**)

Назначение

Удаляет объект **Point** из коллекции.

Синтаксис

ОБЪЕКТ.Remove (index)

Метод **Points.RemoveAll** (Коллекция **Points**)

Назначение

Удаляет все объекты **Point** из коллекции.

Синтаксис

ОБЪЕКТ.RemoveAll

Пример на **Visual Basic**

```
Private Sub btnPointsRemoveAll_Click()  
    Dim pts As New Points  
    Dim pt As New Point  
    Dim nPnts As Integer  
    ' add points to the points collection
```

```

pt.Set -101.023, 45.0452
pts.Add pt
pt.Set -102.023, 49.0452
pts.Add pt
pt.Set -100.012, 34.2564
pts.Add pt
nPnts = pts.Count
' remove all of the points
pts.RemoveAll
nPnts = pts.Count
End Sub

```

Пример на C++

```

void CSampleProjectView::PointsRemoveAll() {
    CMapXPoints pts;
    CMapXPoint pt;

    // Create the Points object
    if(!pts.CreateDispatch(pts.GetClsid())) {
        TRACE0("Failed to Create Points collection");
        return;
    }

    // Create a Point object
    if(!pt.CreateDispatch(pt.GetClsid())) {
        TRACE0("Failed to Create Point object");
        return;
    }
}

```

```

try {
    // add several points to the points collection
    pt.Set(-101.023,45.0452);
    pts.Add(pt);
    pt.Set(-102.023,49.0452);
    pts.Add(pt);
    pt.Set(-100.012,34.2564);
    pts.Add(pt);

    // now, remove all of the points
    pts.RemoveAll();
} catch (ColeDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (ColeException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Объект RangeCategory, Коллекция RangeCategories

Настройка диапазонов тематической карты осуществляется через коллекцию RangeCategories, которая является коллекцией объектов RangeCategory—один объект для каждого ранга,

хранящиеся в возрастающем порядке. Каждый объект в коллекции описывает один диапазон (его стиль, минимальное и максимальное значение и др.).

Свойства объекта

- Свойство RangeCategory.Max
- Свойство RangeCategory.Min
- Свойство RangeCategory.NumItems
- Свойство RangeCategory.Style

Свойства коллекции

- Свойство RangeCategories.AllOthersCategory
- Свойство RangeCategories.Count
- Свойство RangeCategories.Item

Пример на C++

```
// RangeCategories Object
void CSampleProjectView::ChangeChart(CMapXTheme& InTheme) {
    try {
        switch(InTheme.GetType()) {
            case miThemeBarChart:
            case miThemePieChart:
                //...
                break;
            case miThemeRanged:
                // Change the theme to have three ranges, colors white,
                // magenta, and red
                InTheme.GetThemeProperties().SetNumRanges(3);
                InTheme.GetThemeProperties().GetRangeCategories().Item(1)
                    .GetStyle().SetRegionColor(miColorWhite);
                InTheme.GetThemeProperties().GetRangeCategories().Item(2)
                    .GetStyle().SetRegionColor(miColorMagenta);
```

```

        InTheme.GetThemeProperties().GetRangeCategories().Item(3).GetSt
yle().SetRegionColor(miColorRed);

        break;

    }

} catch (COleDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (COleException *e) {

    e->ReportError();

    e->Delete();

}

}

```

Пример на Visual Basic

```

Private Sub ChangeChart(InTheme as MapXLib.Theme)

    Select Case Theme.Type

    Case miThemeBarChart, miThemePieChart

        ' ...

    Case miThemeRanged

        ' Change the theme to have three ranges, colors White,

        ' Magenta, and Red

        InTheme.ThemeProperties.NumRanges = 3

        InTheme.ThemeProperties.RangeCategories(1).Style. _

        RegionColor = miColorWhite

        InTheme.ThemeProperties.RangeCategories(2).Style. _

        RegionColor = miColorMagenta

        InTheme.ThemeProperties.RangeCategories(3).Style. _

```



```

        RegionColor = miColorRed
    End Select
End Sub

```

Свойство **RangeCategories.AllOthersCategory** (Коллекция **RangeCategories**)

Назначение

Возвращает один объект **RangeCategory** из коллекции; этот объект описывает весь диапазон в карте диапазонов. Объект **RangeCategory** возвращается с **AllOthersCategory**, имеет свойства **Min** и **Max**, если эти свойства не определены, их значение по умолчанию равно нулю. Таким образом, Вы можете установить свойство **Style** как показано в примере ниже. Текст, который появляется в легенде для всех диапазонов объекта **AllOthersCategory** может быть установлен со свойством **LegendTexts.AllOthersText**.

Пример

```

Dim ds As Dataset

    Dim lyr as Layer

    Dim thm As MapXLib.Theme

    Dim AllOthers As MapXLib.RangeCategory

    Dim styl As MapXLib.Style

    Set lyr = Map1.Layers.Add("states.tab", 1)

    Set ds = Map1.Datasets.Add(miDataSetLayer, Map1. _
        Layers("STATES"))

    ds.Themes.Add miThemeRanged, "POP_1990", "My Theme", False

    Set thm = ds.Themes("My Theme")

    thm.DataMin = 10000000

```

```

thm.DataMax = 30000000
thm.Visible = True

Set styl = Map1.DefaultStyle
styl.PickRegion

Set AllOthers = thm.ThemeProperties. _
RangeCategories.AllOthersCategory
AllOthers.Style = styl

```

Свойство **RangeCategories.Count** (Коллекция **RangeCategories**)

Назначение

Целое значение, открытое только для чтения, показывает количество диапазонов в тематической карте диапазонов.

Свойство **RangeCategories.Item property** (Коллекция **RangeCategories**)

Назначение

Возвращает один объект **RangeCategory** из коллекции; этот объект описывает диапазон в карте диапазонов.

Свойство **RangeCategory.Max** (Объект **RangeCategory**)

Назначение

Содержит или устанавливает максимальное значение для диапазона в карте диапазонов. Это величина двойной точности. Начальное значение устанавливается при создании тематического слоя (**Themes.Add**) в зависимости от метода распределения (**ThemeProperties.DistMethod**):

- Если **ThemeProperties.DistMethod** это **miEqualCountPerRange** или **miEqualRangeSize**, то новое значение будет устанавливаться каждый раз при новом вычислении объекта **Theme**.

- Если ThemeProperties.DistanceMethod это MiCustomRanges, MapX будет полагать, что Вы устанавливаете это значение самостоятельно, и будете использовать диапазоны, которые Вы определили при группировании значений данных. Генерируется ошибка, если диапазоны перекрываются при новом вычислении тематических значений.

Смотрите также

ThemeProperties.DistanceMethod

Свойство RangeCategory.Min (Объект RangeCategory)

Назначение

Содержит или устанавливает минимальное значение для диапазона в карте диапазонов. Это величина двойной точности. Начальное значение устанавливается при создании тематического слоя (Themes.Add) в зависимости от метода распределения (ThemeProperties.DistanceMethod):

- Если ThemeProperties.DistanceMethod это miEqualCountPerRange или miEqualRangeSize, то новое значение будет устанавливаться каждый раз при новом вычислении объекта Theme.
- Если ThemeProperties.DistanceMethod это MiCustomRanges, MapX будет полагать, что Вы устанавливаете это значение самостоятельно, и будете использовать диапазоны, которые Вы определили при группировании значений данных. Генерируется ошибка, если диапазоны перекрываются при новом вычислении тематических значений.

Смотрите также

ThemeProperties.DistanceMethod

Themes.Add

Theme.AutoRecompute

Свойство RangeCategory.NumItems (Объект RangeCategory)

Назначение

NumItems это количество объектов, которые попадают в диапазон, определенный свойствами Min и Max для этой категории диапазона. Это целая величина. Свойство доступно только для чтения.

Внимание: Это свойство действует, только когда MapX генерирует тематический слой (Theme.ComputeTheme = TRUE). Если Вы вычисляете тематику вручную (Theme.ComputeTheme = FALSE), то NumItems не выдаст количество объектов в диапазоне.

Свойство RangeCategory.Style (Объект RangeCategory)

Назначение

Содержит Style для диапазона тематической карты диапазонов. Оно может быть настроено на существующий объект стиля или Вы можете установить свойства Style индивидуально.

Объект Rectangle

Объекты Rectangle используются для указания протяженности объекта. Некоторые свойства, такие как Map.Bounds, Feature.Bounds и Layer.Bounds являются прямоугольниками объектов. Прямоугольник определяется минимальными и максимальными координатами X и минимальными и максимальными координатами Y. Для географических свойств, таких как Map.Bounds, эти координаты будут из определенной системы координат Map.NumericCoordSys. Для методов, относящихся к Windows (типа Style.DrawLineSample), свойства объекта Rectangle будут в пикселах.

Свойства объекта

- Свойство Rectangle.Height
- Свойство Rectangle.Width
- Свойство Rectangle.XMax
- Свойство Rectangle.XMin
- Свойство Rectangle.YMax
- Свойство Rectangle.YMin

Методы объекта

- Метод Rectangle.Offset
- Метод Rectangle.Set

Свойство **Rectangle.Height** (Объект **Rectangle**)

Назначение

Высота прямоугольника. Высота определяется как разность `Rectangle.YMax - Rectangle.YMin`. Свойство только для чтения, имеет двойную точность. Для изменения высоты прямоугольника, используйте метод `Rectangle.Set`.

Метод **Rectangle.Offset** (Объект **Rectangle**)

Назначение

Изменяет положение прямоугольника при изменении его координат. Сдвигается на `deltaX` по оси X и на `deltaY` по оси Y.

Синтаксис

`Object.Offset(deltaX, deltaY)`

Часть Описание

ОБЪЕКТ Объект `Rectangle`.

`deltaX` это изменение положения по координате x, двойная точность.

`deltaY` это изменение положения по координате y, двойная точность.

Метод **Rectangle.Set** (Объект **Rectangle**)

Назначение

Устанавливает координаты прямоугольника. `Rectangle.Set` будет сортировать параметры так, чтобы обеспечить условия `Rectangle.XMin <= Rectangle.XMax` и `Rectangle.YMin <= Rectangle.YMax`. Разрешено передать параметры, у которых `x1 > x2` или `y1 > y2`.

Синтаксис

`ОБЪЕКТ.Set (x1, y1, x2, y2)`

Часть Описание

ОБЪЕКТ Объект Rectangle.

x1, y1 Двойная точность, представляет координаты угла прямоугольника.

x2, y2 Двойная точность, представляет координаты противоположного угла
прямоугольника.

Свойство Rectangle.Width (Объект Rectangle)

Назначение

Ширина прямоугольника. Ширина определяется разницей `Rectangle.XMax - Rectangle.XMin`. Это свойство только для чтения, двойной точности. Чтобы изменить ширину прямоугольника, используйте метод `Rectangle.Set`.

Свойство Rectangle.XMax (Объект Rectangle)

Назначение

Координата левой стороны прямоугольника. Это свойство только для чтения, двойная точность. Для изменения положения прямоугольника, используйте метод `Rectangle.Set` или `Rectangle.Offset`.

Свойство Rectangle.XMin (Объект Rectangle)

Назначение

Координата правой стороны прямоугольника. Это свойство только для чтения, двойная точность. Для изменения положения прямоугольника, используйте метод `Rectangle.Set` или `Rectangle.Offset`.

Свойство `Rectangle.YMax` (Объект `Rectangle`)

Назначение

Координата верхней стороны прямоугольника. Это свойство только для чтения, двойная точность. Для изменения положения прямоугольника, используйте метод `Rectangle.Set` или `Rectangle.Offset`.

Свойство `Rectangle.YMin` (Объект `Rectangle`)

Назначение

Координата нижней стороны прямоугольника. Это свойство только для чтения, двойная точность. Для изменения положения прямоугольника, используйте метод `Rectangle.Set` или `Rectangle.Offset`.

Объект `ResolveObject` и коллекция `ResolveObjects`

Коллекция `ResolveObjects` действует как параметр для события `ResolveDataBindEx`. Это коллекция которая слоев может связываться при автоматической связи с данными.

Свойства объекта

- `ResolveObject.TableName` (Объект `ResolveObject`)
- `ResolveObject.SourceMatch` (Объект `ResolveObject`)
- `ResolveObject.TableMatch` (Объект `ResolveObject`)

Свойства коллекции

- Свойство `ResolveObjects.Count` (Коллекция `ResolveObjects`)
- Свойство `ResolveObjects.Item` (Коллекция `ResolveObjects`)

Смотрите также

Событие `ResolveDataBindEx`

Свойство ResolveObject.TableName (Объект ResolveObject)

Назначение

Это свойство используется для установки имени таблицы, которую разрешено связывать.

ResolveObject.SourceMatch (Объект ResolveObject)

Назначение

Процент строк из источника сравниваемой таблицы. **Пример:** Если исходные данные это 5 строк и 4 сравниваются, то это и будет = 80.

ResolveObject.TableMatch (Объект ResolveObject)

Назначение

Процент объектов карты, подлежащих сравнению. **Пример:** Если исходные данные имеют 5 строк и 4 сравниваются и карта имеет 50 объектов, то значение будет = 8.

Свойство ResolveObjects.Count (Коллекция ResolveObjects)

Назначение

Количество объектов в коллекции.

Свойство ResolveObjects.Item (Коллекция ResolveObjects)

Назначение

Возвращает объект ResolveObject из коллекции. Это переменная и Вы можете указать имя или индекс начиная с 1. Это свойство по умолчанию коллекции ResolveObjects.

Объект RowValue и коллекция RowValues

Объект RowValue представляет отдельное значение для поля в наборе данных. Коллекция объектов RowValue возвращается при применении Dataset.RowValues.

Коллекция RowValues это группа объектов RowValue которые можно создавать. Коллекция RowValues может быть передана в Layer.UpdateFeature, Layer.AddFeature и Feature.Update для указания значений данных для полей в наборе данных, которые связываются с обновляемым слоем.

Свойства объекта

- Свойство RowValue.ReadOnly (Объект RowValue)
- Свойство RowValue.Dataset (Объект RowValue)
- Свойство RowValue.Field (Объект RowValue)
- Свойство RowValue.Value (Объект RowValue)

Свойства коллекции

- Свойство RowValues.Count (Коллекция RowValues)
- Свойство RowValues.ReadOnly (Коллекция RowValues)
- Свойство RowValues.Item (Коллекция RowValues)

Collection Methods

- RowValues.Remove method (Коллекция RowValues)
- RowValues.Add method (Коллекция RowValues)
- RowValues.RemoveAll method (Коллекция RowValues)
- RowValues.Clone method (RowValues object)

Пример

```
' dump dataset to the debug window
```

```
dim fs as Features
```

```
dim F as Feature
```

```
dim rv as RowValue
```

```

dim rvs as RowValues
Dim s As String

Map1.Layers.Item("usa").BeginAccess miAccessRead

Set fs = Map1.Layers.Item("usa").AllFeatures

For Each F In fs
    Set rvs = Map1.Datasets.Item(1).RowValues(F)
    s = ""
    For Each rv In rvs
        s = s & rv.Value & ", " ' concat each column's values _
        together separated by commas
    Next
    s = Left(s, Len(s) - 2) ' strip off trailing comma
    Debug.Print s
Next

Map1.Layers.Item("usa").EndAccess miAccessEnd

```

Смотрите также

Метод Layer.UpdateFeature

Метод Layer.AddFeature

Метод Feature.Update

Свойство Dataset.RowValues

Свойство RowValue.ReadOnly (Объект RowValue)

Назначение

Это свойство определяет могут быть установлены свойства объекта или нет.

Свойство RowValue.Dataset (Объект RowValue)

Назначение

Это свойство используется для указания набора данных.

Свойство RowValue.Field (Объект RowValue)

Назначение

Это свойство используется для указания поля в наборе данных из которого берутся значения.

Свойство RowValue.Value (Объект RowValue)

Назначение

Это свойство используется для указания значения для поля.

Свойство RowValues.County (Коллекция RowValues)

Назначение

Это свойство только для чтения и указывает количество строк в коллекции RowValues.

Свойство RowValues.ReadOnly (Коллекция RowValues)

Назначение

Это свойство только для чтения, указывает могут быть изменены свойства коллекции или нет. Если "True", то коллекция RowValues открыта только для чтения, иначе коллекцию RowValues можно изменять.

Свойство RowValues.Item (Коллекция RowValues)

Назначение

Возвращает определенный объект RowValue из коллекции RowValues.

Синтаксис

ОБЪЕКТ.Item (index)

Часть **Описание**

ОБЪЕКТ Коллекция RowValues

index Переменная: Указывает какой объект из коллекции будет возвращен. Это может быть имя поля или числовой индекс.

Метод RowValues.Remove (Коллекция RowValues)

Назначение

Этот метод удаляет указанный объект RowValue из коллекции.

Внимание: Хотя объект удаляется из коллекции, набор данных останется прежним.

Синтаксис

ОБЪЕКТ.Remove (index)

Часть **Описание**

ОБЪЕКТ Коллекция RowValues.

index Переменная: Указывает объект из коллекции, который подлежит удалению. Это может быть имя поля или числовой индекс.

Метод RowValues.Add (Коллекция RowValues)

Назначение

Метод добавляет объект RowValue к указанной коллекции RowValues.

Синтаксис

```
[ RowValues= ] OBJECT.Add (RowValue)
```

Часть **Описание**

OBJECT Коллекция RowValues.

RowValue Это объект RowValue который будет добавлен к коллекции.

Метод RowValues.RemoveAll (Коллекция RowValues)

Назначение

Этот метод удаляет все объекты RowValue из коллекции RowValues.

Внимание: Хотя объекты удаляются из коллекции, они останутся в наборе данных.

Синтаксис

```
OBJECT.RemoveAll
```

Часть **Описание**

OBJECT Коллекция RowValues.

Метод RowValues.Clone (Объект RowValues)

Назначение

Этот метод клонирует указанную строку из набора данных и возвращает новый объект RowValues. Клонирование используется не только для чтения, но и для чтения/записи.

Синтаксис

[RowValues=]OBJECT.Clone

Части Описание

ОБЪЕКТ Коллекция RowValues.

Коллекция Selection

Это фундаментальная функция MapX для выбора объектов или записей. Selection это коллекция объектов Feature.

Объект Layer.Selection это специальная коллекция Features в которой объекты могут быть добавлены или удалены из коллекции при использовании доступных инструментов выбора. Вы можете получить снимок выборки со слоя используя метод Clone для возвращения объекта Features.

MapX дает Вам набор команд и инструментов для осуществления выборки, инструмент Select, Выбор-в-круге, Marquee Select и Boundary Select.

Для выбора записей с помощью инструментов, щелкните на ней мышкой или выберите на карте ассоциированные с записями объекты.

Используйте свойство и методы в коллекции Selection и объекты Feature для получения дополнительной информации или манипулирования объектами коллекции.

Методы коллекции

- Метод Selection.Add (Коллекция Selection)
- Метод Selection.ClearSelection (Коллекция Selection)

- Метод Selection.Clone (Коллекция Selection)
- Метод Selection.Common (Коллекция Selection)
- Метод Selection.Remove (Коллекция Selection)
- Метод Selection.Replace (Коллекция Selection)
- Метод Selection.SelectAll (Коллекция Selection)
- Метод Selection.SelectByID (Коллекция Selection)
- Метод Selection.SelectByPoint (Коллекция Selection)
- Метод Selection.SelectByRadius (Коллекция Selection)
- Метод Selection.SelectByRectangle (Коллекция Selection)
- Метод Selection.SelectByRegion (Коллекция Selection)

Пример на Visual Basic

```
'how to select the objects from a features collection
dim fs as Features
dim pt as new Point
pt.SetXY -96.731, 39.728
' assumes map unit is miles
set fs = map1.Layers(1).SelectWithinDistance(pt, 50, _
miSearchTypePartiallyWithin)
map1.Layers(1).Selection.Replace(fs)
```

Пример на C++

```
void CSampleProjectView::SelectObjects50Miles() {
    CMapXFeatures fs;
    CMapXPoint pt;
    if(!pt.CreateDispatch(pt.GetClsid())) {
        TRACE0("Failed to Create Point object");
        return;
    }
    try {
```

```

pt.Set(-96.731,39.728);

fs = m_Map.GetLayers().Item(1).
SearchWithinDistance(pt,50,miUnitMile,
miSearchTypePartiallyWithin);

m_Map.GetLayers().Item(1).GetSelection().Replace(fs);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Смотрите также

Объект Layer

Объект Feature

Метод Selection.Add (Коллекция Selection)

Назначение

Метод добавляет объект Feature или все объекты из объекта Selection в коллекцию. (Операция UNION).

Синтаксис

ОБЪЕКТ.Add (Source)

Части Описание

Объект Selection.

ОБЪЕКТ

Source Это добавляемый объект Feature

Пример на C++

```
// Selection.Add Method
void CSampleProjectView::SearchForCapital(CString Capital) {
    CMapXFindFeature FoundFeature;

    try {

        FoundFeature = m_Map.GetLayers().Item("US Capitals").
            GetFind().Search(Capital, "");

        if(FoundFeature.GetFindRC() % 10 == 1) {

            // Found exact match: add this to the current selection

            m_Map.GetLayers().Item("US Capitals"). GetSelection().
                Add(FoundFeature);

        }

    } catch (COleDispatchException *e) {

        e->ReportError();

        e->Delete();

    } catch (COleException *e) {

        e->ReportError();

        e->Delete();

    }

}
```

Пример на Visual Basic

```
Private Sub SearchForCapital(Capital As String)

    Dim FoundFeature as FindFeature
```

```

Set FoundFeature = Map1.Layers("US Capitals").
Find.Search(Capital)

If FoundFeature.FindRC Mod 10 = 1 Then
    ' Found exact match: add this to the current selection
    Map1.Layers("US Capitals").Selection.Add FoundFeature
End If
End Sub

```

Смотрите также

Объект Feature

Метод Selection.ClearSelection (Коллекция Selection)

Назначение

Метод отменяет выбор всех объектов на данном слое. Используйте Layers.ClearSelection для очистки выборки на всех слоях.

Синтаксис

ОБЪЕКТ.ClearSelection

Части Описание

ОБЪЕКТ Объект Selection.

Метод Selection.Clone (Коллекция Selection)

Назначение

Метод создает копию коллекции как объекта коллекции Features.

Синтаксис

ОБЪЕКТ.Clone

Части **Описание**

ОБЪЕКТ Объект Selection.

Метод Selection.Common (Коллекция Selection)**Назначение**

Метод комбинирует данную коллекцию и другой объект Selection так, что эта коллекция содержит только те объекты, которые есть и там и там. (Операция INTERSECT).

Синтаксис

ОБЪЕКТ.Common (Source)

Части **Описание**

ОБЪЕКТ Объект Selection.

Source Коллекция Selection или коллекция Features для комбинирования с ОБЪЕКТ. Будет удалять каждый объект из ОБЪЕКТ которого нет в коллекции Source.

Метод Selection.Remove (Коллекция Selection)**Назначение**

Метод удаляет объект Feature или все объекты из объекта Selection для данной коллекции. (Операция SUBTRACT).

Синтаксис

ОБЪЕКТ.Remove (Source)

Части **Описание**

Объект Selection.

ОБЪЕКТ

Source Объект Feature, Коллекция Selection или коллекция Features
 которые будут извлекаться из ОБЪЕКТ.

Смотрите также

Объект Feature

Метод Selection.Replace (Коллекция Selection)**Назначение**

Метод перемещает содержимое коллекции с объектом Feature или всеми объектами из объекта коллекции Selection.

Синтаксис

ОБЪЕКТ.Replace (Source)

Части Описание

 Объект Selection.

ОБЪЕКТ

Source Объект Feature, коллекция Features или коллекция Selection.
 ОБЪЕКТ будет перемещен в коллекцию со всеми атрибутами из
 параметра Source.

Смотрите также

Объект Feature

Метод Selection.SelectAll (Коллекция Selection)**Назначение**

Метод выбирает все объекты внутри слоя.

Синтаксис

Map.Selection.SelectAll SelectionTypeConstant

Смотрите также

SelectionTypeConstant

Пример на Visual Basic

```
Map.Layers("USA").Selection.SelectAll miSelectionNew
```

Метод Selection.SelectByID (Коллекция Selection)

Назначение

Метод выбирает объекты по FeatureID или FeatureKey.

Синтаксис

ОБЪЕКТ.SelectByID (FeatureID, Flag, FeatureKey)

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Selection.
--------	-------------------

FeatureID	FeatureID объекта, который будет выбран.
-----------	--

Flag	Определяет, будет ли выделенный объект добавлен, удален или перемещен. Берется значение SelectionTypeConstants.
------	---

FeatureKey	Строка, определяющая объект. Это значение, возвращаемое свойством Feature.FeatureKey.
------------	---

Метод Selection.SelectByPoint (Коллекция Selection)

Назначение

Метод выбирает объект с указанного места на слое.

Синтаксис

ОБЪЕКТ.SelectByPoint (X, Y, Flag)

Часть Описание

ОБЪЕКТ Объект Selection.

X X координата точки, в которой производится выборка. Двойная точность (Долгота).

Y Y координата точки, в которой производится выборка. Двойная точность (Широта).

Flag Параметр определяет, добавляется, удаляется или перемещается выбранный объект из текущей выборки. Берется значение SelectionTypeConstants.

Примечание

Если Вы выбираете полигоны, метод SelectByPoint будет выбирать полигон, содержащий данную точку. Если Вы выбираете точки или линии, метод выберет точки или линии, находящиеся на расстоянии 3 пикселей от заданной точки.

Пример на Visual Basic

```
Private Sub Command1_Click()
    ' Select NY
    Map1.Layers("USA").Selection.SelectByPoint -75.14, _
    42.9, miSelectionNew
    ' Select TX
    Map1.Layers("USA").Selection.SelectByPoint -98, _
    31.56, miSelectionAppend
    ' Select CA
    Map1.Layers("USA").Selection.SelectByPoint -118.92, 36.75, _
    miSelectionAppend
End Sub
```

Пример на C++

```

void CSampleProjectView::SelectStates() {
    try {
        // Select NY
        m_Map.GetLayers().Item("USA").GetSelection().
        SelectByPoint(-75.14,42.9,miSelectionNew);

        // Select TX
        m_Map.GetLayers().Item("USA").GetSelection().
        SelectByPoint(-98,31.56,miSelectionAppend);

        // Select CA
        m_Map.GetLayers().Item("USA").GetSelection().
        SelectByPoint(-118.92,36.75,miSelectionAppend);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Смотрите также

SelectionTypeConstants

Метод Selection.SelectByRadius (Коллекция Selection)

Назначение

Метод выбирает объекты из слоя в указанном радиусе поиска вокруг заданной точки. Флажок (SelectionTypeConstant) определяет добавляются, удаляются или перемещаются ли выбранные объекты текущей выборки.

Синтаксис

ОБЪЕКТ.SelectByRadius (X, Y, Radius, Flag)

Часть	Описание
ОБЪЕКТ	Объект Selection.
X	X координата центра круга. Двойная точность (Долгота).
Y	Y координата центра круга. Двойная точность (Широта).
Radius	Радиус поиска. Двойная точность, указывает расстояние от точки с координатами X и Y в единицах карты.
Flag	Определяет добавляются, удаляются или перемещаются ли выбранные объекты текущей выборки. Берется значение SelectionTypeConstants.

Примечание

Объект считается попавшим в круг поиска, только если туда попадает его центроид. Если Вам нужен усиленный контроль над критериями выборки, используйте метод SearchWithinDistance объекта Layer.

Смотрите также

SelectionTypeConstants

Метод Selection.SelectByRectangle (Коллекция Selection)

Назначение

Метод выбирает объекты внутри прямоугольника.

Синтаксис

ОБЪЕКТ.SelectByRectangle (X1, Y1, X2, Y2, Flag)

Часть Описание

ОБЪЕКТ	Объект Selection.
X1	Первая x координата прямоугольника. Двойная точность (Долгота).
Y1	Первая y координата прямоугольника. Двойная точность (Широта).
X2	Вторая x координата прямоугольника. Двойная точность (Долгота).
Y2	Вторая y координата прямоугольника. Двойная точность (Широта).
Flag	Определяет добавляются, удаляются или перемещаются ли выбранные объекты текущей выборки. Берется значение SelectionTypeConstants.

Примечание

Объект считается попавшим в прямоугольник поиска, только если туда попадает его центрост. Если Вам нужен усиленный контроль над критериями выборки, используйте метод SearchWithinRectangle объекта Layer.

Пример на Visual Basic

```
Private Sub Command1_Click()  
    Map1.Layers("USA").Selection.SelectByRectangle -98, 31.56, _  
    -75.14,42.9, miSelectionNew  
End Sub
```

Смотрите также

Метод Layer.SearchWithinRectangle

Метод Selection.SelectByRegion (Коллекция Selection)

Назначение

Выбирает объекты слоя попавшие в полигон поиска. Флажок (SelectionTypeConstant) определяет добавляются, удаляются или перемещаются ли выбранные объекты текущей выборки.

Синтаксис

ОБЪЕКТ.SelectByRegion (Layer, *FeatureKey, Flag)

Часть Описание

ОБЪЕКТ	Объект Selection.
Layer	Слой, на котором расположен полигон, по которому будет сделана выборка. Берется объект Layer.
FeatureKey	*FeatureKey (строка) полигона, внутри которого делается выбор. Это замена параметру FeatureID. FeatureID до сих пор работает, но рекомендуется использовать новый параметр FeatureKey.
Flag	Определяет добавляются, удаляются или перемещаются ли выбранные объекты текущей выборки. Берется значение SelectionTypeConstants.

Пример на C++

```
// Selection.SelectByRegion Method
void CSampleProjectView::SelectCities() {
    CMapXFindFeature NYState;

    CMapXLayer StateLayer;

    try {
        StateLayer = m_Map.GetLayers().Item("USA");

        // Get the feature that corresponds to New York State
```

```

NYState = StateLayer.GetFind().Search("NY","");
if(NYState.GetFindRC() % 10 == 1) { // Found NY

    // Unselect anything in this layer and select all the cities in NY
    m_Map.GetLayers().Item("US Major Cities").GetSelection().
        SelectByRegion(StateLayer,NYState.GetFeatureID(),
            miSelectionNew);

}
} catch (ColeDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (ColeException *e) {

    e->ReportError();

    e->Delete();

}
}
}

```

Пример на Visual Basic

```

Private Sub SelectCities()

Dim NYState As FindFeature

Dim StateLayer As Layer

Set StateLayer = Map1.Layers("USA")

' Get the feature that corresponds to New York State
Set NYState = StateLayer.Find.Search("NY")

If NYState.FindRC mod 10 = 1 Then ' Found NY

    'Unselect anything in this layer and select all the
    'cities in NY

    Map1.Layers("US Major Cities").Selection.SelectByRegion _
        (StateLayer,NYState.FeatureID,miSelectionNew)

End If

```

End Sub

Примечание

Объект рассматривается как попадающий в полигон выборки, если туда попадает его центроид. Если Вам нужен усиленный контроль над критериями выборки, используйте метод SearchWithinFeature.

Смотрите также

SelectionTypeConstants

Свойство Feature.FeatureKey

Объект и коллекция SourceRow

Свойство DataSet.SourceRows используется для получения коллекции строк (как объектов SourceRow) из исходного источника данных где они были соединены вместе для строки в наборе данных MapX DataSet.

Используйте этот объект для идентификации какая строка или строки в исходных данных представлена строкой в наборе данных MapX DataSet. Это часто полезно при работе с объектами Selections и Feature.

Свойства объекта

- Свойство SourceRow.Row

Свойства коллекции

- Свойство SourceRows.Count
- Свойство SourceRows.Item

Свойство `SourceRow.Row` (Объект `SourceRow`)

Назначение

Содержит количество строк в исходных данных. Это целая величина, доступная только для чтения.

Нумерация строк в исходных данных начинается с 1 и нарастает далее последовательно. Это свойство по умолчанию для объекта `SourceRow`.

Свойство `SourceRows.Count` (Свойство `SourceRows`)

Назначение

Содержит количество объектов `SourceRow` в коллекции. Это целое число, начиная с 1, и доступно только для чтения.

Свойство `SourceRows.Item` (Свойство `SourceRows`)

Назначение

Это свойство получает объект `SourceRow` из коллекции. Индекс используется для того, чтобы указать какой `SourceRow` надо получить. Индекс это целое число от 1 до `SourceRows.Count`. Это свойство по умолчанию коллекции `SourceRows`.

Синтаксис

```
[ SourceRow= ]OBJECT.Item (index)
```

Часть Описание

OBJECT Объект `SourceRows`.

index Строка в наборе данных для получения значения. Это переменная, и может быть номером строки или объектом `Feature`.

Свойство `DataSet.SourceRows` это коллекция `SourceRows`, содержащая объект `SourceRow` для каждой строки в исходном источнике данных, которая была объединена со строкой в наборе данных `MapX DataSet`.

Вы можете использовать объект Feature как параметр Row, с того момента, как объект однозначно идентифицирует одну строку в слое Layer, который связан с набором данных DataSet.

Пример на C++

```
// DataSet.SourceRows Property

// SourceRows.Count Property

void CSampleProjectView::QueryRowsTool(CMapXLayer&
QueryLayer,CMapXDataset& QueryDataset,double X,double Y) {

    CMapXPoint pt;

    CMapXFeatures FeaturesAtPoint;

    COleVariant FtrVariant;

    CMapXFeature ftr;

    CString msg;

    short RowCount;

    if(!pt.CreateDispatch(pt.GetClsid())) {

        TRACE0("Failed to Create Point object");

        return;

    }

    try {

        pt.Set(X,Y);

        FeaturesAtPoint = QueryLayer.SearchAtPoint(pt);

        for(long i=1;i<=FeaturesAtPoint.GetCount;i++) {

            CString rowMsg;

            ftr = FeaturesAtPoint.Item(i);

            FtrVariant.vt = VT_DISPATCH;

            FtrVariant.pdispVal = ftr;
```

```

        RowCount = QueryDataset.GetSourceRows(FtrVariant).
        GetCount();

        rowMsg.Format("%d Rows were aggregated from the
        source data to get data for\n%s\n", RowCount,
        (LPCSTR)ftr.GetName());

        msg += rowMsg;

        msg += '\n';
    }
} catch (COleDispatchException *e) {
    e->ReportError();

    e->Delete();
} catch (COleException *e) {
    e->ReportError();

    e->Delete();
}

AfxMessageBox(msg);
}

```

Пример на Visual Basic

```

Private Sub QueryRowsTool(QueryLayer As MapXLib.Layer, _
QueryDataset As MapXLib.Dataset, X As Double, Y As Double)

    Dim pt As New MapXLib.Point

    Dim FeaturesAtPoint As MapXLib.Features

    Dim RowCount as Integer

    pt.Set X,Y

    FeaturesAtPoint = QueryLayer.SearchAtPoint(pt)

```

```

For Each ftr in FeaturesAtPoint
    RowCount = QueryDataset.SourceRows(ftr).Count
    Debug.Print RowCount & " Rows were aggregated from _
    the source data to get data for "" & ftr.Name & """"
Next
End Sub

```

Объект Style

Объект Style содержит атрибуты для рисуемых объектов символов, линий, полигонов и текстов. Объект содержит атрибуты для всех типов объектов, даже если отдельный тип объекта использует только часть свойств. Имена свойств начинаются с ключевых слов, указывающих на тип изменяемого объекта.

Свойства объекта

- Свойство Style.LineColor
- Свойство Style.LineInterleaved
- Свойство Style.LineStyle
- Свойство Style.LineStyleCount
- Свойство Style.LineSupportsInterleave
- Свойство Style.LineWidth
- Свойство Style.LineWidthUnit
- Свойство Style.MaxVectorSymbolCharacter
- Свойство Style.MinVectorSymbolCharacter
- Свойство Style.RegionBackColor
- Свойство Style.RegionBorderColor
- Свойство Style.RegionBorderStyle
- Свойство Style.RegionBorderWidth
- Свойство Style.RegionBorderWidthUnit
- Свойство Style.RegionColor
- Свойство Style.RegionPattern

- Свойство Style.RegionTransparent
- Свойство Style.SupportsBitmapSymbols
- Свойство Style.SymbolBitmapColor
- Свойство Style.SymbolBitmapName
- Свойство Style.SymbolBitmapOverrideColor
- Свойство Style.SymbolBitmapSize
- Свойство Style.SymbolBitmapTransparent
- Свойство Style.SymbolCharacter
- Свойство Style.SymbolFont
- Свойство Style.SymbolFontBackColor
- Свойство Style.SymbolFontColor
- Свойство Style.SymbolFontHalo
- Свойство Style.SymbolFontOpaque
- Свойство Style.SymbolFontRotation
- Свойство Style.SymbolFontShadow
- Свойство Style.SymbolType
- Свойство Style.SymbolVectorColor
- Свойство Style.SymbolVectorSize
- Свойство Style.TextFont property
- Свойство Style.TextFontAllCaps
- Свойство Style.TextFontBackColor
- Свойство Style.TextFontColor
- Свойство Style.TextFontDblSpace
- Свойство Style.TextFontHalo
- Свойство Style.TextFontOpaque
- Свойство Style.TextFontRotation
- Свойство Style.TextFontShadow

Методы объекта

- Метод Style.Clone
- Метод Style.DrawLineSample
- Метод Style.DrawRegionSample
- Метод Style.DrawSymbolSample
- Метод Style.DrawTextSample
- Метод Style.ExportLineSample
- Метод Style.ExportRegionSample

- Метод `Style.ExportSymbolSample`
- Метод `Style.ExportTextSample`
- Метод `Style.PickLine`
- Метод `Style.PickRegion`
- Метод `Style.PickSymbol`
- Метод `Style.PickText`

Метод `Style.Clone` (Объект `Style`)

Назначение

Метод возвращает одиночный объект (не связанный со слоем) стиль которого является копией другого объекта `Style`.

Синтаксис

`ОБЪЕКТ.Clone`

Вместо `object` поместите объект `Style`.

Примечание

Когда Вы ссылаетесь на объект `Style` принадлежащий другому объекту (например, используя выражение типа `objFeature.Style`), то любые внесенные изменения в стиле неизбежно отразятся на “родительском” объекте.

Чтобы создать измененный стиль, не влияя на “родительский” объект, используйте метод `Clone` для создания копии стиля. Стиль, возвращаемый этим методом, является одиночным объектом; изменения, внесенные в этот объект `Style` не действуют автоматически на другой объект. Вы можете присвоить одиночный объект `Style` любому свойству объекта `Style`.

Объект `Style`, созданный посредством метода `Clone` поддерживает растровые символы, только если оригинальный объект `Style` также поддерживает растровые символы.

Пример на Visual Basic

`Working with copies of Style objects:`

`The following Visual Basic code example copies a Style _`
`object, allows the user to modify the style, and then applies _`

```

the modified Style

. Dim s As Style

' copy the style from layer 2 to use as starting point
Set s = Map1.Layers(2).Style.Clone

' let user modify style
s.PickSymbol

' now assign to layer 1

' note that layer 2's style is not affected
Set Map1.Layers(1).Style = s

' make sure that new style is used as override
' for all objects in layer
Map1.Layers(1).OverrideStyle = True

```

Пример На C++

```

void CSampleProjectView::ChangeStyle() {
    CMapXStyle s;
    try {
        // Copy the style from layer 2 to use as a starting point
        s = m_Map.GetLayers().Item(2).GetStyle().Clone();
        // Let the user modify the style
        s.PickSymbol();
        // Now assign to layer 1; layer 2 is not affected
        m_Map.GetLayers().Item(1).SetStyle(s);
        // Make sure that the new style is used as override
        // for all objects in layer
        m_Map.GetLayers().Item(1).SetOverrideStyle(TRUE);
    } catch (COleDispatchException *e) {

```

```

        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Метод **Style.DrawLineSample** (Объект **Style**)

Назначение

Метод используется, если пользователь хочет иметь образец стиля линии, нарисованной в HDC.

Синтаксис

OBJECT.DrawLineSample (HDC, Rectangle)

Части

Описание

HDC	это DC в котором нарисован образец линии.
Rectangle	Rectangle - объект MapXLib.Rectangle определяющий координаты (в пикселах) в DC так что образец будет нарисован в нем. Если образец не нуждается в этом, чтобы заполнить весь прямоугольник, то он будет помещен в его центр.

Пример на **Visual Basic**

```

Dim rect As New MapXLib.Rectangle

' picturebox's ScaleMode must be 'Pixel' for this code to work
rect.Set 0, 0, picturebox.ScaleWidth, picturebox.ScaleHeight

' To draw a line sample:
map1.DefaultStyle.DrawLineSample picturebox.hDC, rect

```

```

' To draw a region sample:
map1.DefaultStyle.DrawRegionSample pictureBox.hDC, rect
' To draw a symbol sample:
map1.DefaultStyle.DrawSymbolSample pictureBox.hDC, rect
' To draw a text sample:
map1.DefaultStyle.DrawTextSample pictureBox.hDC, rect, _
    "The Quick Brown Cow"

```

Пример на C++

```

void CSampleProjectView::DrawSample(CWnd& drawto) {
    CMapXRectangle rect;
    CRect WinRect;
    if(!rect.CreateDispatch(rect.GetClsid())) {
        TRACE0("Failed to Create Rectangle object");
        return;
    }
    drawto.GetWindowRect(&WinRect);
    try {
        rect.Set(WinRect.left,WinRect.top,WinRect.right,
            WinRect.bottom);
        // Draw a line sample
        m_Map.GetDefaultStyle().
            DrawLineSample((long)drawto.GetDC()->m_hDC,rect);
        // Draw a region sample
        m_Map.GetDefaultStyle().
            DrawRegionSample((long)drawto.GetDC()->m_hDC,rect);
        // Draw a symbol sample

```

```

    m_Map.GetDefaultStyle().
    DrawSymbolSample((long)drawto.GetDC()->m_hDC,rect);

    // Draw a text sample

    m_Map.GetDefaultStyle().
    DrawTextSample((long)drawto.GetDC()->m_hDC,rect,
    "The Quick Brown Cow");

} catch (CleDbDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (CleDbException *e) {

    e->ReportError();

    e->Delete();

}

}

```

Метод Style.DrawRegionSample (Объект Style)

Назначение

Метод используется если пользователь хочет иметь образец стиля региона, нарисованного в HDC.

Синтаксис

ОБЪЕКТ.DrawRegionSample (HDC, Rectangle)

Части Описание

ОБЪЕКТ	Объект Style.
HDC	это DC в котором нарисован образец.
Rectangle	это объект MapXLib.Rectangle определяющий координаты (в пикселах) в DC так что образец будет нарисован в нем. Если образец не нуждается в том, чтобы заполнить весь прямоугольник, то он будет помещен в его центр.

Пример на Visual Basic

```
Dim rect As New MapXLib.Rectangle

' picturebox's ScaleMode must be 'Pixel' for this code to work
rect.Set 0, 0, picturebox.ScaleWidth, picturebox.ScaleHeight

' To draw a line sample:
map1.DefaultStyle.DrawLineSample picturebox.hDC, rect

' To draw a region sample:
map1.DefaultStyle.DrawRegionSample picturebox.hDC, rect

' To draw a symbol sample:
map1.DefaultStyle.DrawSymbolSample picturebox.hDC, rect

' To draw a text sample:
map1.DefaultStyle.DrawTextSample picturebox.hDC, rect, _
"The Quick Brown Cow"
```

Пример на C++

```
void CSampleProjectView::DrawSample(CWnd& drawto) {
    CMapXRectangle rect;
    CRect WinRect;

    if(!rect.CreateDispatch(rect.GetClsid())) {
        TRACE0("Failed to create Rectangle object");
        return;
    }

    drawto.GetWindowRect(&WinRect);

    try {
```

```

rect.Set(WinRect.left,WinRect.top,WinRect.right,
WinRect.bottom);

// Draw a line sample

m_Map.GetDefaultStyle().
DrawLineSample((long)drawto.GetDC()->m_hDC,rect);

// Draw a region sample

m_Map.GetDefaultStyle().
DrawRegionSample((long)drawto.GetDC()->m_hDC,rect);

// Draw a symbol sample

m_Map.GetDefaultStyle().
DrawSymbolSample((long)drawto.GetDC()->m_hDC,rect);

// Draw a text sample

m_Map.GetDefaultStyle().
DrawTextSample((long)drawto.GetDC()->m_hDC,
rect,"The Quick Brown Cow");

} catch (ColeDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (ColeException *e) {

    e->ReportError();

    e->Delete();

}

}

```

Метод **Style.DrawSymbolSample** (Объект **Style**)

Назначение

Метод используется, если пользователю нужен образец стиля символа, нарисованного в HDC.

Синтаксис

OBJECT.DrawSymbolSample (HDC, Rectangle)

Часть **Описание**

OBJECT Объект Style.

HDC это DC в котором нарисован образец.

Rectangle это объект MapXLib.Rectangle определяющий координаты (в пикселах) в DC так что образец будет нарисован в нем. Если образец не нуждается в том, чтобы заполнить весь прямоугольник, то он будет помещен в его центр.

Пример на Visual Basic

```
Dim rect As New MapXLib.Rectangle

' picturebox's ScaleMode must be 'Pixel' for this code to work
rect.Set 0, 0, picturebox.ScaleWidth, picturebox.ScaleHeight

' To draw a line sample:
map1.DefaultStyle.DrawLineSample picturebox.hDC, rect

' To draw a region sample:
map1.DefaultStyle.DrawRegionSample picturebox.hDC, rect

' To draw a symbol sample:
map1.DefaultStyle.DrawSymbolSample picturebox.hDC, rect

' To draw a text sample:
map1.DefaultStyle.DrawTextSample picturebox.hDC, rect, _
"The Quick Brown Cow"
```

Пример на C++

```
void CSampleProjectView::DrawSample(CWnd& drawto) {
    CMapXRectangle rect;
    CRect WinRect;
```

```

if(!rect.CreateDispatch(rect.GetClsid())) {
    TRACE0("Failed to Create Rectangle object");
    return;
}

drawto.GetWindowRect(&WinRect);

try {
    rect.Set(WinRect.left,WinRect.top,WinRect.right,
WinRect.bottom);

    // Draw a line sample
    m_Map.GetDefaultStyle().
DrawLineSample(drawto.GetDC()->m_hDC,rect);

    // Draw a region sample
    m_Map.GetDefaultStyle().
DrawRegionSample(drawto.GetDC()->m_hDc, rect);

    // Draw a symbol sample
    m_Map.GetDefaultStyle().
DrawSymbolSample(drawto.GetDC()->m_hDC,rect);

    // Draw a text sample
    m_Map.GetDefaultStyle().
DrawTextSample(drawto.GetDC()->m_hDC,rect,
"The Quick Brown Cow");
} catch (COleDispatchException *e) {
    e->ReportError();

    e->Delete();
} catch (COleException *e) {
    e->ReportError();

    e->Delete();
}

```

```
}
```

Метод Style.DrawTextSample(Объект Style)

Назначение

Этот метод используется, если пользователь хочет иметь образец стиля текста, нарисованного в HDC.

Синтаксис

OBJECT.DrawTextSample (HDC, Rectangle, SampleText)

Части	Описание
OBJECT	Объект Style.
HDC	это DC в котором нарисован образец.
Rectangle	это объект MapXLib.Rectangle определяющий координаты (в пикселах) в DC так что образец будет нарисован в нем. Если образец не нуждается в том, чтобы заполнить весь прямоугольник, то он будет помещен в его центр.
SampleText	Пример текста, используемого для отображения образца TextStyle.

Пример на Visual Basic

```
Dim rect As New MapXLib.Rectangle

' picturebox's ScaleMode must be 'Pixel' for this code to work
rect.Set 0, 0, picturebox.ScaleWidth, picturebox.ScaleHeight

' To draw a line sample:
Map1.DefaultStyle.DrawLineSample picturebox.hDC, rect

' To draw a region sample:
Map1.DefaultStyle.DrawRegionSample picturebox.hDC, rect

' To draw a symbol sample:
```

```

Map1.DefaultStyle.DrawSymbolSample pictureBox.hDC, rect
' To draw a text sample:
Map1.DefaultStyle.DrawTextSample pictureBox.hDC, rect, _
"The Quick Brown Cow"

```

Пример на C++

```

void CSampleProjectView::DrawSample(CWnd& drawto) {
    CMapXRectangle rect;
    CRect WinRect;
    if(!rect.CreateDispatch(rect.GetClsid())) {
        TRACE0("Failed to create Rectangle object");
        return 0;
    }
    drawto.GetWindowRect(&WinRect);
    try {
        rect.Set(WinRect.left,WinRect.top,WinRect.right,
        WinRect.bottom);
        // Draw a line sample
        m_Map.GetDefaultStyle().
        DrawLineSample(drawto.GetDC()->m_hDC,rect);
        // Draw a region sample
        m_Map.GetDefaultStyle().
        DrawRegionSample(drawto.GetDC()->m_hDC,rect);
        // Draw a symbol sample
        m_Map.GetDefaultStyle().
        DrawSymbolSample(drawto.GetDC()->m_hDC,rect);
        // Draw a text sample

```

```
        m_Map.GetDefaultStyle().
        DrawTextSample(drawto.GetDC()->m_hDC,rect,
        "The Quick Brown Cow");
    } catch (COleDispatchException *e) {
        e->ReportError();

        e->Delete();
    } catch (COleException *e) {
        e->ReportError();

        e->Delete();
    }
}
```

Метод Style.ExportLineSample (Объект Style)

Назначение

Экспортирует образец стиля линии.

Синтаксис

ОБЪЕКТ.ExportLineSample (Destination, Format, Width, Height, [BackColor])

Часть	Описание
ОБЪЕКТ	Объект Style
Destination	Спецификация файла, куда будет экспорт, типа "C:\Temp\Symbol.bmp". Если Вы указываете "CLIPBOARD" вместо файла, изображение попадет в буфер.
Format	Выходной формат. Берется значение ExportFormatConstants. Внимание: Если Вы указываете miFormatGIF (для генерирования GIF файлов), прочтите лицензионное соглашение для Users of GIF Files.
Width	Ширина выхода. Величина двойной точности, в пикселах.
Height	Высота выхода. Величина двойной точности, в пикселах.
BackColor	Переменная. Определяет цвет фона. Определяет величину OLE_COLOR. Может быть сплошной цвет или один из Windows

System Colors такой как 'Window Background'.

Пример на Visual Basic

```
'This sample demonstrates the Style.ExportLineSample method.
'It uses
'the method to place a line in the clipboard as a BMP.
Private Sub Command1_Click()
'Export a 2 pixel by 9 pixel Line to the clipboard in BMP Format
    Map1. ExportLineSample "clipboard", miFormatBMP, 2, 9, _
        miColorWhite

    If Clipboard.GetFormat(vbCFBitmap) Then
        Debug.Print "Clipboard contains a BMP"
    Else
        Debug.Print "Clipboard does not contain a BMP"
    End If
End Sub
```

Метод Style.ExportRegionSample (Объект Style)

Назначение

Экспортирует образец стиля полигона.

Синтаксис

ОБЪЕКТ.ExportRegionSample (Location, Format, Width, Height, [BackColor])

Часть	Описание
ОБЪЕКТ	Объект Style
Location	Спецификация файла, куда будет экспорт, типа "C:\Temp\Symbol.bmp". Если Вы указываете "CLIPBOARD" вместо файла, изображение попадет в

	буфер.
Format	Выходной формат. Берется значение ExportFormatConstants. Внимание: Если Вы указываете miFormatGIF (для генерирования GIF файлов), прочтите лицензионное соглашение для Users of GIF Files.
Width	Ширина выхода. Величина двойной точности, в пикселах.
Height	Высота выхода. Величина двойной точности, в пикселах.
BackColor	Переменная. Определяет цвет фона. Определяет величину OLE_COLOR. Может быть сплошной цвет или один из Windows System Colors такой как 'Window Background'.

Пример на Visual Basic

```
'This sample demonstrates the Style.ExportRegionSample
method. It uses
'the method to place a region in the clipboard as a BMP.

Private Sub Command1_Click()
'Export a 120 pixel by 90 pixel region to the clipboard in BMP
'Format
    Map1.ExportRegionSample "clipboard", miFormatBMP, 120, _
    90, miColorWhite

    If Clipboard.GetFormat(vbCFBitmap) Then
        Debug.Print "Clipboard contains a BMP"
    Else
        Debug.Print "Clipboard does not contain a BMP"
    End If
End Sub
```

Метод Style.ExportSymbolSample (Объект Style)

Назначение

Метод экспортирует образец стиля символа.

Синтаксис

ОБЪЕКТ.ExportSymbolSample (Location, Format, Width, Height, [BackColor])

Часть	Описание
ОБЪЕКТ	Объект Style
Location	Спецификация файла, куда будет экспорт, типа "C:\Temp\Symbol.bmp". Если Вы указываете "CLIPBOARD" вместо файла, изображение попадет в буфер.
Format	Выходной формат. Берется значение ExportFormatConstants. Внимание: Если Вы указываете miFormatGIF (для генерирования GIF файлов), прочтите лицензионное соглашение для Users of GIF Files.
Width	Ширина выхода. Величина двойной точности, в пикселах.
Height	Высота выхода. Величина двойной точности, в пикселах.
BackColor	Переменная. Определяет цвет фона. Величину типа OLE_COLOR. Может быть сплошной цвет или один из Windows System Colors такой как 'Window Background'.

Пример на Visual Basic

```
'This sample demonstrates the Style.ExportSymbolSample
'method. It uses
'the method to place a symbol in the clipboard as a BMP.

Private Sub Command1_Click()
'Export a 12 pixel by 9 pixel symbol to the clipboard in BMP _
'Format
    Map1.ExportSymbolSample "clipboard", miFormatBMP,
    12, 9, miColorWhite
```



```

    If Clipboard.GetFormat(vbCFBitmap) Then
        Debug.Print "Clipboard contains a BMP"
    Else
        Debug.Print "Clipboard does not contain a BMP"
    End If
End Sub

```

Метод Style.ExportTextSample (Объект Style)

Назначение

Метод экспортирует образец стиля текста.

Синтаксис

ОБЪЕКТ.ExportTextSample (Location, Format, Width, Height, [BackColor])

Часть	Описание
ОБЪЕКТ	Объект Style
Location	Спецификация файла, куда будет экспорт, типа "C:\Temp\Symbol.bmp". Если Вы указываете "CLIPBOARD" вместо файла, изображение попадет в буфер.
Format	Выходной формат. Берется значение ExportFormatConstants. Внимание: Если Вы указываете miFormatGIF (для генерирования GIF файлов), прочтите лицензионное соглашение для Users of GIF Files.
Width	Ширина выхода. Величина двойной точности, в пикселах.
Height	Высота выхода. Величина двойной точности, в пикселах.
SampleText	Строка. Образец текста для экспорта.
BackColor	Переменная. Определяет цвет фона. Величина типа OLE_COLOR. Может быть сплошной цвет или один из Windows System Colors такой как 'Window Background'.

Пример на Visual Basic

```

'This sample demonstrates the Style.ExportTextSample method.

```

```

'It uses the method to place text in the clipboard as a BMP.

Private Sub Command1_Click()
'Export a 2 pixel by 9 pixel text to the clipboard in BMP Format
    Map1.ExportTextSample "clipboard", miFormatBMP, 2, 9, _
        "Hello World", miColorWhite

    If Clipboard.GetFormat(vbCFBitmap) Then
        Debug.Print "Clipboard contains a BMP"
    Else
        Debug.Print "Clipboard does not contain a BMP"
    End If
End Sub

```

Свойство Style.LineColor (Объект Style)

Назначение

Содержит цвет линии. Используется для линейных объектов. Это величина OLE_COLOR.

Свойство Style.LineInterleaved (Объект Style)

Назначение

Свойство только для чтения, логическое. Используется для пересекающегося типа линий со свойством LineSupportsInterleave. Пересекающийся стиль линий может использоваться при пересечении линий на карте.

Пример на Visual Basic

```
Private Sub btnTestInterleave_Click()
```

```

Dim lyr As MapXLib.Layer
Dim Style As MapXLib.Style
Dim sPat As Single

Set lyr = Map1.Layers.Add("us_hiway.tab", 1)
lyr.Style.LineStyle = 63

' If the pen style supports being interleaved, then
' turn interleaving on.
If (lyr.Style.LineSupportsInterleave) Then
    lyr.Style.LineInterleaved = True
End If

Set Style = lyr.Style
Style.LineWidthUnit = miStyleUnitTenthsOfPoint
' Set the line width to .5 points
Style.LineWidth = 5

Style.RegionBorderWidthUnit = miStyleUnitPixel
'Set the region border width to 3 pixels
Style.RegionBorderWidth = 3

End Sub

```

Пример на C++

```

void CSampleProjectView::TestInterleave() {
    CMapXLayer lyr;

```

```

CMapXStyle Style;

try {
    lyr = m_Map.GetLayers().Add("us_hiway.tab",1);
    lyr.GetStyle().SetLineStyle(63);

    // If the line style supports interleaving, turn
    interleaving on
    if(lyr.GetStyle().GetLineSupportsInterleave() == TRUE)
        lyr.GetStyle().SetLineInterleaved(TRUE);

    Style = lyr.GetStyle();
    Style.SetLineWidthUnit(miStyleUnitTenthsOfPoint);
    // Set the line width to .5 points
    Style.SetLineWidth(5);

    Style.SetRegionBorderWidthUnit(miStyleUnitPixel);
    // Set the region border width to 3 pixels
    Style.SetRegionBorderWidth(3);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}

```

}

Свойство Style.LineStyle (Объект Style)

Назначение

Возвращает или устанавливает номер стиля линии (сплошной, пунктир и др.); используется для изображения линейных объектов.

Примечание

Вы можете присвоить любое значение PenStyleConstants этому свойству. Таблица ниже показывает полный перечень номеров для LineStyle.

51		24			
50		23			
52		25			
54		21			
53		20		11	
55		40		10	
51		48		12	
50		41		14	
10		40		13	
18		42		15	
11		44		11	
10		43		10	
12		45		00	
14		41		08	
13		40		01	
15		30		00	
11		38		02	
10		31		04	
00		30		03	
08		32		01	
01		33		00	
00		35		20	
		31		28	
		30		21	
		50		20	
		58		22	

Обратите внимание, что таблица показывает стили линий используемых по умолчанию. Если пользователь имеет свой стиль в файле `mapx.rep` (файл, определяющий стили линий), то номера в этом файле могут не совпадать со стилями в этой таблице. Используйте свойство `LineStyleCount` для определения номеров стилей в `.rep` файле.

Смотрите также

`PenStyleConstants`

Свойство `Style.LineStyleCount` (Объект `Style`)

Назначение

Целое, доступное только для чтения, указывающее, какие стили линий доступны. Счетчик включает в себя стили линий, считанные из файла `mapx.rep`.

Свойство `Style.LineSupportsInterleave` (Объект `Style`)

Назначение

Логическое свойство, доступное только для чтения. Определяет, возможно ли пересечение для данного типа линий.

Смотрите также

`Style.LineInterleaved`

Свойство `Style.LineWidth` (Объект `Style`)

Назначение

Содержит ширину линии в пикселах. Используется для линейных объектов. Это целая величина.

Свойство **Style.LineWidthUnit** (Объект **Style**)

Назначение

Определяет единицы измерения ширины линий. Ширина может быть или в пикселах или в пунктах, это определяется параметром `StyleUnitConstants`.

Смотрите также

`Style.LineWidth`

Метод **Style.PickLine** (Объект **Style**)

Назначение

Отображает диалог `Line Style Picker`, позволяющий пользователю выбрать свойства стиля линии. Объект `Style` обновляется, с учетом новых заданных параметров, когда пользователь нажимает ОК в диалоге.

Синтаксис

`ОБЪЕКТ.PickLine`

Метод **Style.PickRegion** (Объект **Style**)

Назначение

Метод отображает диалог `Region Style Picker`, позволяющий пользователю выбрать свойства стиля полигона. Объект `Style` обновляется, с учетом новых заданных параметров, когда нажата кнопка ОК.

Синтаксис

`ОБЪЕКТ.PickRegion`

Метод **Style.PickSymbol** (Объект **Style**)

Назначение

Отображает диалог **Symbol Style Picker** позволяющий пользователю выбрать свойства стиля символа. Объект **Style** обновляется, с учетом новых заданных параметров, когда нажата кнопка **ОК**.

Синтаксис

ОБЪЕКТ.PickSymbol

Метод **Style.PickText** (Объект **Style**)

Назначение

Отображает диалог **Text Style Picker** позволяющий пользователю выбрать свойства стиля текста. Объект **Style** обновляется, с учетом новых заданных параметров, когда пользователь нажимает **ОК** в диалоге.

Синтаксис

ОБЪЕКТ.PickText

Свойство **Style.RegionBackColor** (Объект **Style**)

Назначение

Содержит цвет полигона (**background color**). Свойство используется, когда **RegionPattern** отличается от **miPatternSolid** для конкретного полигона. Это величина **OLE_COLOR**.

Смотрите также

FillPatternConstants

Свойство **Style.RegionBorderColor** (Объект **Style**)

Назначение

Содержит цвет рамки полигона. Используется для объектов типа полигон. Это величина `OLE_COLOR`.

Свойство **Style.RegionBorderStyle** (Объект **Style**)

Назначение

Содержит стиль линии для рамки полигона. Используется для объектов типа полигон. Берется значение `PenStyleConstants`.

Смотрите также

`PenStyleConstants`

Свойство **Style.RegionBorderWidth** (Объект **Style**)

Назначение

Содержит ширину рамки полигона. Используется для объектов типа полигон. Это целая величина и ширина указана в пикселах.

Свойство **Style.RegionBorderWidthUnit** (Объект **Style**)

Назначение

Определяет единицы измерения для ширины рамки(границы) полигона. Ширина может измеряться в пикселах или пунктах, как определено величиной `StyleUnitConstants`.

Смотрите также

`Style.RegionBorderWidth`

Свойство **Style.RegionColor** (Объект **Style**)

Назначение

Содержит цвет полигона. Используется для объектов типа полигон. Это величина типа `OLE_COLOR`.

Свойство **Style.RegionPattern** (Объект **Style**)

Назначение

Содержит шаблон стиля полигона. Используется для полигонов. Берется значение из `FillPatternConstants`.

Смотрите также

`FillPatternConstants`

Свойство **Style.RegionTransparent** (Объект **Style**)

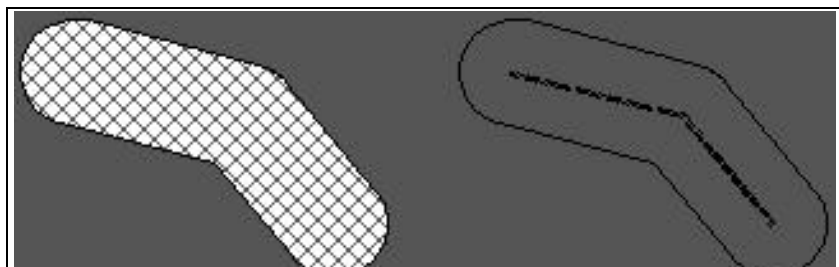
Назначение

Логическая величина, доступная только для чтения, указывающая, прозрачная ли заливка у полигона.

Примечание

Если полигон раскрашен по шаблону, Вы можете контролировать, будет ли его штриховка прозрачной. Используйте прозрачную штриховку, если полигон перекрывает объекты, которые должны быть видимыми на карте.

Следующий рисунок показывает два буферных полигона. Полигон слева непрозрачный; полигон справа - прозрачный, что позволяет Вам видеть оба объекта.



Свойство `RegionTransparent` не применяется к полигонам, раскрашенным сплошной заливкой.

Свойство `Style.SupportsBitmapSymbols` (Объект `Style`)

Назначение

Значение `True` показывает, что объект `Symbol` поддерживает растровые символы.

Только объекты `Feature`, `Layer` и `Theme` поддерживают растровые символы.

Примечание

Значение `True` указывает что объект `Symbol` поддерживает растровые символы.

Сейчас только `Feature`, `Layer` и `Theme` - объекты поддерживают растровые символы.

Смотрите также

Символы `Bitmap`

Свойство `Style.SymbolBitmapColor` (Объект `Style`)

Назначение

Значение, доступное только для чтения типа `OLE_COLOR`, используется для определения фона растрового символа. По умолчанию белый цвет.

Примечание

Если цвет фона оастрового символа отличен от белого цвета, тогда:

1. Установите свойство `SymbolBitmapColor` на нужный Вам цвет.

2. Установите свойство `SymbolBitmapOverrideColor` на `True`.

Когда Вы меняете символ фонта `TrueType` на растровый символ — другими словами, когда Вы меняете свойство `SymbolType` на `miSymbolTypeBitmap` (1) — MapX устанавливает свойство `SymbolBitmapColor` для совпадения со свойством `SymbolFontColor`.

Смотрите также

Символы Bitmap

Свойство `Style.SymbolBitmapName` (Объект `Style`)

Назначение

Строковая величина доступная для чтения–записи, представляющая имя растрового файла, используемого как растровый символ.

Примечание

Имя `Name` которое Вы указали не должно включать имя директории. Растровый файл должен находиться в пользовательской директории символов, определенной так:

```
<path>\CUSTSYMB
```

где `<path>` это путь к `mitmdl30.dll`.

Смотрите также

Символы Bitmap

Свойство `Style.SymbolBitmapOverrideColor` (Объект `Style`)

Назначение

Логическая величина, доступная для чтения–записи, указывающее перерисован ли цвет фона, принятый по умолчанию (белый), в цвет, определенный пользователем.

Примечание

По умолчанию False, это значит, цвет фона растрового символа будет белым. Если Вы хотите изменить цвет фона растрового символа на какой-то другой, указанный в свойстве `SymbolBitmapColor`, установите это свойство на True.

Смотрите также

Символы Bitmap

Свойство `Style.SymbolBitmapSize` (Объект `Style`)

Назначение

Целое, доступное для чтения–записи, указывающее размер символа, в пунктах. 72 points эквивалентно 1 дюйму.

Примечание

Когда Вы меняете символ фонта TrueType на растровый символ — другими словами, когда Вы меняете свойство `SymbolType` на `miSymbolTypeBitmap` (1) — MapX устанавливает свойство `SymbolBitmapSize` так, что оно совпадает со значением свойства `SymbolFont.Size`. Например, если Вы меняете 12–пунктный символ TrueType на растровый символ, то `SymbolBitmapSize` будет 12.

Смотрите также

Символы Bitmap

Свойство `Style.SymbolBitmapTransparent` (Объект `Style`)

Назначение

Логическая величина, доступная для чтения–записи. Определяет, является ли цвет фона растрового символа прозрачным.

Примечание

Если установлено True, то фон растрового символа будет прозрачным, иначе фон будет отображаться. По умолчанию False.

Смотрите также

Символы Bitmap

Свойство Style.SymbolCharacter (Объект Style)**Назначение**

Определяет, какой символ используется. Используется для символьных объектов. Берется целое значение от 0 до 255, если используются шрифты TrueType, то возвращается ASCII - код символа.

Смотрите также

Style.SymbolFont

Свойство Style.SymbolFont (Объект Style)**Назначение**

Возвращает стандартный объект Font (IFontDispatch, OLE_FONT) который позволяет Вам устанавливать свойства шрифта (Bold, Size и др.) для изменения вида символа.

Примечание

Символ может быть определен комбинацией шрифта TrueType и символом внутри этого шрифта (Style.SymbolCharacter). Таким образом, если Вам надо изменить вид символа, Вам надо получить и изменить свойство Style.SymbolFont.

Свойство SymbolFont доступно только для чтения. Таким образом, Вы можете установить возвращаемые свойства объекта Font.

Смотрите также

Свойство `Style.SymbolCharacter`

Установка атрибутов шрифтов

Пример

Следующий пример использует свойство `Style.SymbolFont` для возвращения объекта `Font`. Свойство `Size` объекта `Font` устанавливается на 36 pt. Далее используется новый стиль символа.

```
Map1.Layers(1).OverrideStyle = True
Map1.Layers(1).Style.SymbolFont.Size = 36
```

Свойство `Style.SymbolFontBackColor` (Объект `Style`)

Назначение

Содержит цвет background символа. Используется для символьных объектов. Это величина `OLE_COLOR`.

Свойство `Style.SymbolFontColor` (Объект `Style`)

Назначение

Содержит цвет символа. Используется для символьных объектов. Величина типа `OLE_COLOR`.

Свойство `Style.SymbolFontHalo` (Объект `Style`)

Назначение

Применяется только к `Layer Styles` и `Label Styles`. Определяет, есть ли кайма вокруг символа. Используется для символьных объектов. Кайма это ‘буфер’ вокруг символа - помогает создать контраст вокруг символа. Логическая величина.

Свойство **Style.SymbolFontOpaque** (Объект **Style**)

Назначение

Определяет, отображает ли символ цвет background. Используется для символьных объектов. Логическая величина. Только для символов TrueType.

Свойство **Style.SymbolFontRotation** (Объект **Style**)

Назначение

Целая величина, доступная для чтения–записи, показывающая на сколько градусов повернут символ TrueType.

Примечание

Целое от 0 до 360, показывающее угол под которым развернут символ. Вращение по часовой стрелке.

Это свойство применяется только к символам TrueType. Растровые символы не вращаются.

Объекты Feature, Layer и Theme поддерживают вращение символов. Если объект не поддерживает вращение символов, то это свойство игнорируется.

Свойство **Style.SymbolFontShadow** (Объект **Style**)

Назначение

Применяется только к Layer Styles и Label Styles. Определяет, нарисована ли тень под символом. Используется для символьных объектов. Это логическая величина.

Свойство **Style.MinVectorSymbolCharacter** (Объект **Style**)

Назначение

Это свойство применяется к объектам Symbol Style. Только для чтения, целая величина, содержащая минимальное значение Style.SymbolCharacter которое выводит векторный символ.

Для таких символов свойство `Style.SymbolType` должно быть установлено на `miSymbolTypeVector`.

Свойство `Style.MaxVectorSymbolCharacter` (Объект `Style`)

Назначение

Это свойство применяется к объектам `Symbol Style`. Только для чтения, целое, содержащее максимальное значение `Style.SymbolCharacter` которое выводит векторный символ. Для таких символов `Style.SymbolType` должно быть установлено на `miSymbolTypeVector`.

Свойство `Style.SymbolVectorColor` (Объект `Style`)

Назначение

Свойство применяется к векторным объектам `Style`. Оно считывает-записывает значение типа `OLE_COLOR`, которое указывает цвет векторного символа. По умолчанию это `miColorBlack`.

Свойство `Style.SymbolVectorSize` (Объект `Style`)

Назначение

Это свойство применяется к векторным объектам `Style`. Оно считывает-записывает целое значение которое определяет высоту векторного символа в пунктах. По умолчанию 12.

Свойство `Style.TextFont` (Объект `Style`)

Назначение

Свойство возвращает стандартный объект `Font` (`IFontDispatch`, `OLE_FONT`) который позволяет Вам устанавливать свойства шрифта (`Bold`, `Size`, и др.) для изменения изображения шрифта.

Примечание

Свойство `TextFont` только для чтения; таким образом, Вы можете установить свойства объекта `Font` который возвращается.

Свойство **Style.TextFontAllCaps** (Объект **Style**)

Назначение

Свойство применяется только к Layer Styles и Label Styles. Определяет, отображается ли текст заглавными буквами. Используется для текстового объектов. Логическая величина.

Свойство **Style.TextFontBackColor** (Объект **Style**)

Назначение

Содержит цвет background для текста. Используется для текстовых объектов. Величина типа OLE_COLOR.

Смотрите также

OLE_Color Values

Свойство **Style.TextFontColor** (Объект **Style**)

Назначение

Содержит цвет текста. Используется для текстовых объектов. Это величина OLE_COLOR.

Смотрите также

OLE_Color Values

Свойство **Style.TextFontDblSpace** (Объект **Style**)

Назначение

Применяется только к Layer Styles и Label Styles. Определяет, отображается ли текст с увеличенным расстоянием между буквами. Используется для текстовых объектов. Это логическая величина.

Свойство **Style.TextFontHalo** (Объект **Style**)

Назначение

Применяется только к Layer Styles и Label Styles. Определяет, есть ли кайма вокруг текста. Используется для текстовых объектов. Это логическая величина.

Свойство **Style.TextFontOpaque** (Объект **Style**)

Назначение

Определяет, отображается ли текст цветом фона. Используется для текстовых объектов. Это логическая величина.

Свойство **Style.TextFontShadow** (Объект **Style**)

Назначение

Применяется только к Layer Styles и Label Styles. Определяет, есть ли тень под текстом. Используется для текстовых объектов. Это логическая величина.

Свойство **Style.TextFontRotation** (Объект **Style**)

Назначение

Определяет угол поворота текста в градусах по часовой стрелке. Допустимы значения $0 \leq x \leq 360$. Сейчас свойство доступно только для объектов Feature. Для других типов объектов это свойство игнорируется.

Свойство **Style.SymbolType** (Объект **Style**)

Назначение

Короткая целая величина (short), показывает, как отображается точечный объект (символ TrueType или как растр). Это должна быть одна из констант SymbolTypeConstants.

Примечание

Что бы поменять символ TrueType на растровый, установите свойство SymbolType на miSymbolTypeBitmap (1).

Обратите внимание, что только некоторые объекты (Feature, Layer, Theme) поддерживают растровые символы. Попытка установить SymbolType на miSymbolTypeBitmap приведет к сбросу, если объект не поддерживает растровые символы.

Смотрите также

Символы Bitmap

Объект и коллекция Theme

Объекты Theme содержат атрибуты тематического слоя.

Каждый набор данных DataSet имеет коллекцию Themes (DataSets.Themes). Коллекция Theme имеет методы и свойства используемые для добавления и удаления объектов Theme из коллекции.

Свойства объекта

- Свойство Theme.AutoReCompute
- Свойство Theme.ComputeTheme
- Свойство Theme.DataMax
- Свойство Theme.DataMin
- Свойство Theme.Fields
- Свойство Theme.Layer
- Свойство Theme.Legend
- Свойство Theme.Name
- Свойство Theme.ThemeProperties
- Свойство Theme.Type
- Свойство Theme.Visible

Методы объекта

- Метод Theme.ThemeDlg

Свойства коллекции

- Themes.Count
- Themes.Item

Методы коллекции

- Метод Themes.Add
- Метод Themes.Remove
- Метод Themes.RemoveAll

Свойство Theme.AutoReCompute (Объект Theme)

Назначение

Определяет, будут ли автоматически пересчитываться диапазоны тематического слоя при изменении свойств тематического слоя. Логическая величина, по умолчанию равна TRUE.

Обсуждение

Когда свойства тематики изменяются (такие как ThemeProperties.NumRanges), объект Theme вычисляется заново. Процесс занимает определенное время и возобновляется при каждом изменении свойств тематики. Если Вам надо поменять только некоторые свойства, установите AutoReCompute на FALSE, измените эти свойства, а затем установите AutoRecompute на TRUE, и перевычисление будет одноразовое.

Свойство Theme.ComputeTheme (Объект Theme)

Назначение

Определяет, вычисляются ли тематические слои. Это логическое свойство и по умолчанию равно True. При значении True будут расчитываться тематические слои для данных. Если установлено False будут созданы невидимые тематические объекты с 20 диапазонами для карты индивидуальных значений и 5 диапазонами для карты диапазонов. Вы можете затем вручную установить минимальное и максимальное значение для настройки тематического слоя используя Theme.DataMin и Theme.DataMax. Свойства ThemeProperties.NumRanges и IndividualValueCategory.Value используются для изменения настроек.

Для карт диапазонов Вы можете вручную установить диапазоны или рассчитать диапазоны равного размера, используя значения минимума (Theme.DataMin) и максимума (Theme.DataMax).

Смотрите также

Свойство Theme.DataMax

Свойство Theme.DataMin

IndividualValueCategory.Value

Свойство Theme.DataMax (Объект Theme)

Назначение

Определяет максимальное значение для установки тематических диапазонов или рассчитывает диапазоны равного размера когда свойство Theme.ComputeTheme или параметр ComputeTheme метода Themes.Add установлены на FALSE.

Смотрите также

Theme.ComputeTheme

Theme.DataMin

Свойство Theme.DataMin (Объект Theme)

Назначение

Определяет минимальное значение для установки тематических диапазонов или рассчитывает диапазоны равного размера когда свойство Theme.ComputeTheme или параметр ComputeTheme метода Themes.Add установлены на FALSE.

Смотрите также

Theme.ComputeTheme

Theme.DataMax

Свойство Theme.Fields (Объект Theme)

Назначение

Возвращает доступную только для чтения коллекцию Fields, представляющую установку полей используемых в наборе данных, на которых базируется тематический слой.

Свойство Theme.Layer (Объект Theme)

Назначение

Свойство только для чтения, возвращающее Объект Layer, представляет слой, на котором базируется тематический слой.

Свойство Theme.Legend (Объект Theme)

Назначение

Каждый объект Theme имеет Объект Legend (Theme.Legend). Объект Legend содержит свойства для контроля изображения тематической легенды. Каждый объект ThemeCategory (RangeCategory, IndividualValueCategory или MultiVarCategory) имеет вход в легенду, содержащуюся в объекте LegendText.

Смотрите также

Объект Legend

Свойство Theme.Name (Объект Theme)

Назначение

Имя объекта Theme. Имя должно быть уникальным в коллекции Themes. Это свойство доступное для чтения/записи и определяется как параметр метода Themes.Add или генерируется программой MapX когда создается тематический слой.

Примечание

Свойство Name может использоваться вместе с свойством Themes.Item и методом Themes.Remove.

Смотрите также

Метод Themes.Add

Свойство Themes.Item

Метод Themes.Remove

Метод Theme.ThemeDlg (Объект Theme)

Назначение

Открывает диалог, позволяющий пользователю настраивать стили тематического слоя. Если пользователь нажимает 'ОК', внесенные в диалоге изменения настроек немедленно применяются к слою карты.

Синтаксис

[Boolean=]OBJECT.ThemeDlg ([HelpFile], [HelpID])

Части Описание

OBJECT Объект Theme.

HelpFile это дополнительный параметр, путь к файлу справки .hlp содержащему раздел, относящийся к этому диалогу.

HelpID это дополнительный параметр, который ссылается на индекс ID раздела электронной справки, данному в .hlp файле.

Внимание: Если дополнительные параметры не заданы, кнопка "help" в диалоге не появится.

Примечание

Если дополнительные параметры не заданы, то кнопка Справка не появится в диалоге.

Возвращаемое значение ThemeDlg это True если пользователь нажал ОК и False если пользователь нажал Отмена.

Пример на Visual Basic

```
` This example displays the Theme Dialog for a theme when its
` legend is double-clicked on.
```

```
Private Sub Map1_ThemeModifyRequested(ByVal Theme As Object)
    Theme.ThemeDlg
End Sub
```

Пример на C++

```
void CSampleProjectView::OnThemeModifyRequested(LPDISPATCH
ThemeDispatch) {
    // Display the Theme dialog for a theme when its legend
    // is double clicked on
    CMapXTheme theme;
    theme.AttachDispatch(ThemeDispatch,FALSE);
    try {
        theme.ThemeDlg();
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Свойство Theme.ThemeProperties (Объект Theme)

Назначение

Объект, содержащий свойства, относящиеся к тематическому слою. Устанавливаемые свойства, содержащиеся в объекте ThemeProperties определяются типом объекта Theme (тип тематической карты).

Смотрите также

Объект ThemeProperties

Свойство Theme.Type (Объект Theme)

Назначение

Тип тематической карты для объекта Theme. Это значение ThemeTypeConstants и свойство открыто только для чтения.

Смотрите также

ThemeTypeConstants

Свойство Theme.Visible (Объект Theme)

Назначение

Устанавливает, является ли видимым тематический слой. Логическая величина, по умолчанию равна TRUE.

Метод Themes.Add (Коллекция Themes)

Назначение

Создает тематический слой и добавляет его к коллекции Themes для обычного набора данных.

Примечание

Когда создается тематический слой, вычисляются диапазоны для слоев с большим количеством записей в таблице, на их обработку может уйти много времени. Параметр ComputeTheme позволяет Вам создавать невычисляемый тематический слой любого типа. Невычисляемый тематический слой дает Вам возможность создать тематику без автоматически вычисляемых диапазонов. Потом Вы можете создать диапазоны самостоятельно. Это более быстрый по времени путь.

Внимание: MapX будет создавать пустой тематический слой из пустого набора данных. Тематический объект создается и добавляется к коллекции тематических объектов, но поскольку тематический слой базируется на наборе данных без данных, тематика не будет видна.

Пустой набор данных может быть создан из-за ошибки в программировании.

Синтаксис

ОБЪЕКТ.Add ([Type], [Field], [Name], [ComputeTheme])

Часть	Описание
ОБЪЕКТ	Объект Themes.
Type	Определяет тип создаваемого тематического слоя. Берется значение ThemeTypeConstants. Это необязательный параметр и если он не указан (или указан как miThemeAuto), MapX будет пытаться выбрать по умолчанию тематику, базирующуюся не числе передаваемых полей. Если MapX не может по умолчанию выбрать тип тематической карты, сгенерируется ошибка.
Field(s)	Указывает поле или поля для тематической карты. Поле может быть указано по имени, индексу или по объекту Field. Если Вы создаете тематический слой используя несколько переменных (например круговые или столбчатые диаграммы), в коллекцию Fields передается массив имен полей, индексов или объектов Field. Это необязательный параметр и если он не указан, MapX использует первое числовое поле набора данных DataSet.
Name	Указывает имя тематической карты. Это строковый параметр. Это необязательный параметр, и если он не указан, MapX генерирует имя, такое как StatesBySales.
ComputeTheme	Логическая величина. По умолчанию равно True, что приводит к расчету

тематической карты из исходных данных.

Если значение установлено на `False` будут созданы невидимые тематические объекты 20 диапазонов для карты индивидуальных значений и 5 диапазонов для карт диапазонов. Вы можете затем вручную установить минимум и максимум для настройки тематической карты с `Theme.DataMin` и `Theme.DataMax`. Для карты диапазонов Вы можете вручную установить диапазоны или рассчитать равные диапазоны, учитывая значения (`Theme.DataMin`) и (`Theme.DataMax`).

Пример на Visual Basic

```
'This example assumes we have a dataset object "ds"
'containing the fields in the cols array.

'Add a ranged theme:
ds.Themes.Add miThemeRanged, "TOTPOPCUR", "My Ranged Theme"

'Add a Bar Chart theme:
dim cols(1 to 3) as string
cols(1) = "TOTPOPHIS"
cols(2) = "TOTPOPCUR"
cols(3) = "TOTPOPPRO"
ds.Themes.Add miThemeBarChart, cols, "My Bar Theme"
```

Пример на C++

```
void CSampleProjectView::ThemesAdd(CMapXDataset& ds) {
    try {
        // Add a ranged theme
        ds.GetThemes().Add(miThemeRanged, "TOTPOPCUR",
            "My Ranged Theme");
    }
```

```

m_Map.Refresh();

// Create the array of strings
SAFEARRAY FAR* psa=NULL;
SAFEARRAYBOUND rgsabound[1];

LONG i;

CString str1("TOTPOPHIS"), str2("TOTPOPCUR"),
str3("TOTPOPPRO");

ColeVariant Field;

rgsabound[0].lLbound = 1;
rgsabound[0].cElements = 3;

psa = SafeArrayCreate(VT_BSTR, 1, rgsabound);
if(psa == NULL){
    return;
}

i=1;
SafeArrayPutElement(psa,&i,str1.AllocSysString());
i=2;
SafeArrayPutElement(psa,&i,str2.AllocSysString());
i=3;
SafeArrayPutElement(psa,&i,str3.AllocSysString());

// put array in field param
Field.vt = VT_ARRAY | VT_BSTR;
Field.parray = psa;

```

```

        // Add a bar chart theme

        ds.GetThemes().Add(ColeVariant((long)miThemeBarChart),
        Field,ColeVariant("My Bar Theme"));

    } catch (ColeDispatchException *e) {

        e->ReportError();

        e->Delete();

    } catch (ColeException *e) {

        e->ReportError();

        e->Delete();

    }

}

```

Пример невычисляемых тематических объектов на Visual Basic

```

Dim ds As MapXLib.Dataset

Dim parm As New ODBCQueryInfo

parm.SqlQuery = "select * from usa"

parm.DataSource = "mapstats"      ' name of odbc datasource

parm.ConnectionString = "ODBC;"    ' can be left blank

Set ds = Map1.Datasets.Add(miDataSetODBC, parm, "odbc Dataset",
, , , , True)

ds.Themes.Add , , , False

ds.Themes(1).AutoRecompute = False

ds.Themes(1).ThemeProperties.DistMethod = miEqualRangeSize

ds.Themes(1).DataMin = 0

ds.Themes(1).DataMax = 32000000

ds.Themes(1).ThemeProperties.NumRanges = 10

ds.Themes(1).AutoRecompute = True

```

```
ds.Themes(1).Visible = True
```

Смотрите также

Объект Field

ThemeTypeConstants

Свойство Theme.DataMax

Свойство Theme.DataMin

Свойство Theme.Name

Свойство Theme.Type

Свойство Themes.Count (Коллекция Themes)

Назначение

Количество объектов Theme в коллекции Themes. Это свойство доступно только для чтения.

Пример на Visual Basic

```
Dim iThmCount as Integer

iThmCount = Map.Datasets(1).Themes.Count

Debug.Print "There are " & iThmCount & " Themes attached to _
dataset" & Map.Datasets(1).Name
```

Пример на C++

```
void CSampleProjectView::PrintNumThemes() {
    long thmCount;
    CString msg;
    try {
        thmCount = m_Map.GetDatasets().Item(1).GetThemes().
        GetCount();
    }
```



```

        msg.Format("There are %d Themes attached to dataset %s",
            thmCount, m_Map.GetDatasets().Item(1).GetName());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
    AfxMessageBox(msg);
}

```

Свойство Themes.Item (Коллекция Themes)

Назначение

Возвращает объект Theme из коллекции Themes. Берется параметр, указывающий тип тематического слоя. Параметр это переменная, представляющая индекс от 1, или имя тематического слоя. Это свойство по умолчанию для коллекции Themes.

Синтаксис

ОБЪЕКТ.Item (index)

Часть	Описание
-------	----------

ОБЪЕКТ	Объект Themes.
--------	----------------

index	Индексный номер тематического объекта, начиная с 1, или имя тематического слоя.
-------	---

Смотрите также

Свойство Theme.Name

Метод Themes.Remove (Коллекция Themes)

Назначение

Удаляет указанные объекты Theme из коллекции Themes.

Внимание: Если Вы удаляете объект, индексы коллекции перенумеровываются так, что бы заполнить пробел.

Синтаксис

ОБЪЕКТ.Remove (index)

Часть **Описание**

ОБЪЕКТ Объект Themes.

index Индексный номер объекта Theme или имя тематики.

Пример

```
Map.Datasets(1).Themes.Remove "My DD Theme"
```

- или -

```
Map.Datasets(1).Themes.Remove 1
```

Смотрите также

Свойство Theme.Name

Метод Themes.RemoveAll (Коллекция Themes)

Назначение

Удаляет все объекты Theme из коллекции.

Синтаксис

ОБЪЕКТ.RemoveAll

Пример на Visual Basic

```
Private Sub btnThemesRemoveAll_Click()
    Dim nThemes As Integer

    Dim ds As Dataset

    Set ds = Map1.Datasets.Add(miDataSetLayer, Map1.Layers(1))

    ' add a default theme
    ds.Themes.Add

    nThemes = ds.Themes.Count

    ' remove all of the themes
    ds.Themes.RemoveAll

    nThemes = ds.Themes.Count

End Sub
```

Пример на C++

```
void CSampleProjectView::RemoveAllThemes() {
    CMapXDataset ds;
    CMapXLayer Layer;
    COleVariant LayerVariant;
    try {
        Layer = m_Map.GetLayers().Item(1);
        LayerVariant.vt = VT_DISPATCH;
        LayerVariant.pdispVal = Layer.m_lpDispatch;
        LayerVariant.pdispVal->AddRef();
        ds=m_Map.GetDatasets().Add(miDataSetLayer,LayerVariant);
        // add a default theme
        ds.GetThemes().Add();
        ds.GetThemes().RemoveAll();
    }
```

```

    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Объект ThemeProperties

Объект ThemeProperties это свойство объекта Theme содержащее информацию, определяющую Theme.

Внимание: Свойство Theme.Properties до сих пор работает, но рекомендуется использовать объект ThemeProperties.

Свойства объекта

- Свойство ThemeProperties.AllowEmptyRanges
- Свойство ThemeProperties.ApplyAttribute
- Свойство ThemeProperties.BarFrame
- Свойство ThemeProperties.BarFramedStyle
- Свойство ThemeProperties.BarGraduatedStack
- Свойство ThemeProperties.BarIndependentScale
- Свойство ThemeProperties.BarStacked
- Свойство ThemeProperties.BarWidth
- Свойство ThemeProperties.BorderStyle
- Свойство ThemeProperties.ColorMethod
- Свойство ThemeProperties.DataValue
- Свойство ThemeProperties.DistMethod
- Свойство ThemeProperties.DotColor
- Свойство ThemeProperties.DotSize
- Свойство ThemeProperties.Graduated
- Свойство ThemeProperties.GraduateSizeBy

- Свойство ThemeProperties.Independent
- Свойство ThemeProperties.IndividualValueCategories
- Свойство ThemeProperties.InflectionColor
- Свойство ThemeProperties.InflectionRange
- Свойство ThemeProperties.InflectRanges
- Свойство ThemeProperties.MultivarCategories
- Свойство ThemeProperties.NegativeSymbolStyle
- Свойство ThemeProperties.NumRanges
- Свойство ThemeProperties.PieClockwise
- Свойство ThemeProperties.PieGraduated
- Свойство ThemeProperties.PieHalfPies
- Свойство ThemeProperties.PieStartAngle
- Свойство ThemeProperties.PositiveSymbolStyle
- Свойство ThemeProperties.RangeCategories
- Свойство ThemeProperties.RoundBy
- Свойство ThemeProperties.RoundRanges
- Свойство ThemeProperties.ShowNegativeValues
- Свойство ThemeProperties.Size
- Свойство ThemeProperties.SpreadBy
- Свойство ThemeProperties.SymbolStyle
- Свойство ThemeProperties.ValuePerDot
- Свойство ThemeProperties.Width

Свойство ThemeProperties.AllowEmptyRanges (Объект ThemeProperties)

Назначение

Свойство открытое для чтения/записи; определяет, допустимы ли пустые диапазоны в карте диапазонов.

Примечание

Если свойство ThemeProperties.DistanceMethod установлено на miCustomRanges (value: 0), пустые диапазоны допускаются, в зависимости от свойства AllowEmptyRanges.

Смотрите также

Legend.ShowEmptyRanges

Свойство ThemeProperties.DataValue (Объект ThemeProperties)

Назначение

Применяется к Размерным символам, Круговым и Столбчатым диаграммам. Работает совместно со свойством Size чтобы контролировать размер тематических объектов. Это величина двойной точности - величина контролирующая размер тематических диаграмм, определенный свойством Size. По умолчанию значение для этого свойства устанавливается на наибольшее значение в тематическом слое.

Смотрите также

Свойство ThemeProperties.Size

Свойство ThemeProperties.DistMethod (Объект ThemeProperties)

Назначение

Это свойство применяется к картам диапазонов. Оно определяет как создаются диапазоны (в коллекции RangeCategories), когда объект Theme вычисляется заново. Это свойство берет значение DistribMethodConstants и по умолчанию оно равно miEqualCountPerRange.

Допустимые типы распределений следующие:

Распределение	Описание
miEqualRangeSize	Берет максимальное и минимальное значение данных, и выбирает диапазоны, равные по интервалам (то есть 0-25; 25-50; 51-75; 76-100)
miEqualCountPerRange	Пытается распределить диапазоны так, что каждый диапазон содержит одинаковое количество.
miCustomRanges	Вам надо указать собственные диапазоны в коллекции RangeCategories (и MapX не будет вычислять диапазоны).
miNaturalBreak	Диапазоны делятся используя алгоритм, пытающийся минимизировать разницу между значениями данных и средним

	значением
miStandardDeviation	Средний диапазон соответствует среднему значению данных, а диапазоны выше и ниже занимают по 1 единице стандартного отклонения данных от среднего.

Смотрите также

Коллекция RangeCategories

Свойство ThemeProperties.DotSize (Объект ThemeProperties)

Назначение

Свойство применяется к картам Плотность точек. Оно определяет размер точек, используемых в тематической карте Плотность точек. Значение типа DotSizeConstants, по умолчанию равно miDotSizeSmall.

Свойство ThemeProperties.Graduated (Объект ThemeProperties)

Назначение

Это свойство применяется к картам Круговых диаграмм. Оно определяет, градуирован ли размер круговых диаграмм общим суммарным значением переменной. Логическая величина по умолчанию равная TRUE.

Смотрите также

ThemeProperties.PieGraduated

Свойство ThemeProperties.Independent (Объект ThemeProperties)

Назначение

Это свойство применяется к Столбчатым диаграммам. Оно определяет, обрабатываются ли данные для столбцов независимо, (несравнимые значения типа Населения и Налоги). Логическая величина и по умолчанию равна FALSE. Разработчик должен установить ее на true для многовариантных столбцов, где данные для каждого столбца, относящиеся к одному

графическому объекту, берутся из разных полей или максимальные значения для отдельных столбцов различаются очень сильно. Например, один столбец будет отражать население а другой - размер экспорта. Данные по населению измеряются в миллионах и их диапазон от 1 до 50. Если свойство установлено на True, столбец с максимальным населением по размерам будет равен столбцу с максимальным экспортом. Если свойство равно false, то такого выравнивания не будет.

Внимание: BarIndependentScale теперь является предпочтительным именем для этого свойства.

Смотрите также

Свойство ThemeProperties.BarIndependentScale

Свойство ThemeProperties.IndividualValueCategories (Объект ThemeProperties)

Назначение

Возвращает коллекцию IndividualValueCategories; используется с картами Индивидуальных значений.

Свойство ThemeProperties.MultivarCategories (Объект ThemeProperties)

Назначение

Возвращает коллекцию MultivarCategories; используется с тематическими картами, в которых есть множество переменных (столбчатые или круговые диаграммы).

Свойство ThemeProperties.NumRanges (Объект ThemeProperties)

Назначение

Это свойство применяется к картам диапазонов. Это целая величина, отражающая количество диапазонов. По умолчанию равно 5.

Свойство ThemeProperties.RangeCategories (Объект ThemeProperties)

Назначение

Возвращает коллекцию RangeCategories; используется в картах диапазонов.

Свойство ThemeProperties.Size (Объект ThemeProperties)

Назначение

Это свойство применяется к Столбчатым и Круговым диаграммам тематических карт. Оно действует в сочетании со свойством DataValue для контроля размера тематических графиков на карте. Оно указывает высоту тематических графиков в бумажных единицах (PaperUnit). Это величина двойной точности и по умолчанию равна 0.25.

Смотрите также

Свойство Map.PaperUnit

Свойство ThemeProperties.DataValue

Свойство ThemeProperties.SpreadBy (Объект ThemeProperties)

Назначение

Определяет как будут рассчитаны промежуточные диапазоны для тематических карт. MapX может вычислять промежуточные цвета между двумя заданными и размеры между двумя заданными размерами.

Примечание

Берется значение SpreadByConstants. Для контроля того, какие цвета или размеры используются для начального и конечного диапазонов, устанавливаются значения ThemeProperties.RangeCategories(x).Style, где $x = 1$ для первого диапазона (стартовое значение) и x это ThemeProperties.NumRanges для последнего диапазона (конечное значение).

Свойство ThemeProperties.SymbolStyle (Объект ThemeProperties)

Назначение

Это свойство применяется к картам Размерных символов. Оно определяет, какой символ используется, точнее его параметры, размер отображаемого символа в зависимости от величины тематического параметра, указанного в свойстве DataValue. Это объект Style и это свойство доступно только для чтения (хотя свойства SymbolStyle доступны для чтения/записи).

Внимание: PositiveSymbolStyle это теперь предпочтительное имя для этого свойства.

Смотрите также

Свойство ThemeProperties.PositiveSymbolStyle

Объект Style

Свойство ThemeProperties.DataValue

Свойство ThemeProperties.ValuePerDot (Объект ThemeProperties)

Назначение

Это свойство применяется к тематическим картам Плотность точек. Указывает, какую величину представляет одна точка. Например, если ValuePerDot была 2000, и регион имеет население 48000, то будут нарисованы 24 точки в регионе. Это величина двойной точности.

Свойство ThemeProperties.Width (Объект ThemeProperties)

Назначение

Это свойство применяется к Столбчатым диаграммам тематических карт. Оно указывает ширину каждого столбца в тематическом слое, в единицах карты (Map.PaperUnit). Это величина двойной точности, по умолчанию равная .25 дюйма.

Смотрите также

Свойство Map.PaperUnit

Свойство ThemeProperties.PieClockwise (Объект ThemeProperties)

Назначение

Это логическое свойство, по умолчанию true, применяемое к круговым диаграммам. Оно определяет направление отрисовки круговых диаграмм: если true, возрастание значений по часовой стрелке.

Смотрите также

Свойство ThemeProperties.PieStartAngle (Объект ThemeProperties)

Назначение

Свойство применяется к круговым диаграммам. Целая величина, определяющая, под каким углом будет отрисован первый сектор. Измеряется в градусах, возрастает против часовой стрелки, 0 градусов это положение справа. По умолчанию равно 180.

Смотрите также

Свойство ThemeProperties.PieClockwise

Свойство ThemeProperties.PieHalfPies (Объект ThemeProperties)

Назначение

Логическое свойство, применяющееся к круговым диаграммам. Если True, будет отрисовано половина круга а не целый. По умолчанию равно False.

Свойство ThemeProperties.PieGraduated (Объект ThemeProperties)

Назначение

Логическое свойство, применяющееся к круговым диаграммам. Определяет, меняется ли размер диаграммы. Если True, то общий размер диаграммы градуируется суммарным значением. Если False, то все диаграммы будут одного размера. По умолчанию True.

Внимание: Это свойство заменяет ThemeProperties.Graduated. PieGraduated это предпочтительное свойство.

Смотрите также

Свойство ThemeProperties.GraduateSizeBy

Свойство ThemeProperties.BarStacked (Объект ThemeProperties)

Назначение

Логическая величина, по умолчанию False, применяющаяся к столбчатым диаграммам. Если True, столбцы будут располагаться стопкой один над другим, если false, столбцы расположатся рядом друг с другом.

Смотрите также

Свойство ThemeProperties.BarGraduatedStack

Свойство ThemeProperties.BarGraduatedStack (Объект ThemeProperties)

Назначение

Это логическая величина, по умолчанию True, применяющаяся к столбчатым диаграммам , если ThemeProperties.BarStacked имеет значение True. Оно определяет, изменяется ли суммарный размер столбчатой диаграммы. Если true, размер столбчатой диаграммы будет градуироваться суммарным значением; если false, все столбцы будут иметь одинаковое суммарное значение.

Смотрите также

Свойство ThemeProperties.BarStacked

Свойство ThemeProperties.GraduateSizeBy

Свойство **ThemeProperties.BarIndependentScale** (Объект **ThemeProperties**)

Назначение

Свойство применяется к столбчатым диаграммам. Определяет, интерпретируются ли данные для столбцов независимо (несовместимые значения, например, Население и Налоги). Логическая величина, по умолчанию равна `False`. Разработчик должен установить `True` для многовариантных столбчатых диаграмм, где данные из разных полей для одного объекта несовместимы по размерности. Это свойство заменяет `ThemeProperties.Independent`.

Свойство **ThemeProperties.BarWidth** (Объект **ThemeProperties**)

Назначение

Величина двойной точности, свойство применяется к столбчатым диаграммам. Оно указывает ширину каждого столбца в “бумажных” единицах (`Map.PaperUnit`). Величина двойной точности, и по умолчанию равна 0.25 дюймов. Это свойство заменяет `ThemeProperties.Width`.

Смотрите также

Свойство `Map.PaperUnit`

Свойство **ThemeProperties.BarFramed** (Объект **ThemeProperties**)

Назначение

Логическое свойство устанавливается на `False`. Изображает столбцы окруженные рамкой.

Свойство **ThemeProperties.BarFrameStyle** (Объект **ThemeProperties**)

Назначение

Объект `Style` определяющий стиль рамки для столбцов.

Смотрите также

Объект Style

Свойство ThemeProperties.BorderStyle

Свойство ThemeProperties.BorderStyle (Объект ThemeProperties)**Назначение**

Объект Style действующий для столбчатых и круговых диаграмм. Он определяет стиль линии для рамки вокруг диаграмм.

Смотрите также

Объект Style

Свойство ThemeProperties.BarFrameStyle

Свойство ThemeProperties.ShowNegativeValues (Объект ThemeProperties)**Назначение**

Логическое свойство применяемое к Размерным символам. Если True, то отрицательные значения будут показаны в соответствии с их абсолютными значениями; символ, отражающий величину -4000 будет иметь тот же размер, что и символ для 4000. Если False, то отрицательные значения показаны не будут. По умолчанию False.

Смотрите также

Свойство ThemeProperties.PositiveSymbolStyle

Свойство ThemeProperties.NegativeSymbolStyle

Свойство `ThemeProperties.PositiveSymbolStyle` (Объект `ThemeProperties`)

Назначение

Это логическое свойство применяемое к Размерным символам. Оно определяет символ который используется для всех положительных значений, а размер символа будет определяться значением, указанным в свойстве `DataValue`. Это объект `Style`, и это свойство для чтения/записи (хотя свойства объекта `Style` тоже для чтения/записи).

Внимание: Это свойство замещает `ThemeProperties.SymbolStyle`.

Смотрите также

Объект `Style`

Свойство `ThemeProperties.ShowNegativeValues`

Свойство `ThemeProperties.NegativeSymbolStyle`

Свойство `ThemeProperties.DataValue`

Свойство `ThemeProperties.NegativeSymbolStyle` (Объект `ThemeProperties`)

Назначение

Применяется к Размерным символам, чье свойство `ThemeProperties.ShowNegativeValues` равно `True`. Это объект `Style` который определяет символ, использующийся для отрицательных значений. Это объект `Style`. Свойство для чтения/записи (так же как и свойства объекта `Style`).

Смотрите также

Объект `Style`

Свойство `ThemeProperties.ShowNegativeValues`

Свойство `ThemeProperties.PositiveSymbolStyle`

Свойство `ThemeProperties.DataValue`

Свойство ThemeProperties.GraduateSizeBy (Объект ThemeProperties)

Назначение

Это свойство применяется к Размерным символам, как и к Круговым и Столбчатым диаграммам, которые установлены в режиме градуировки. Это должна быть короткая целая величина, сравниваемая с одной из констант GraduationConstants. Она определяет отношение между значением в данных и размером соответствующего символа на диаграмме или карте. По умолчанию это `miGraduateBySquareRoot`.

Смотрите также

Свойство ThemeProperties.PieGraduated

Свойство ThemeProperties.BarGraduatedStack

Свойство ThemeProperties.DotColor (Объект ThemeProperties)

Назначение

Свойство применяется к плотности точек. Оно определяет цвет точек. Имеет значение типа `OLE_COLOR`.

Свойство ThemeProperties.RoundRanges (Объект ThemeProperties)

Назначение

Логическое свойство, применяющееся к тематическим картам Диапазонов. Оно определяет, округляются ли значения диапазонов. По умолчанию `True`. Величина, по которой производится округление, определяется свойством ThemeProperties.RoundBy.

Смотрите также

Свойство ThemeProperties.RoundBy

Свойство ThemeProperties.RoundBy (Объект ThemeProperties)

Назначение

Значение двойной точности, применяемое к Диапазнам. Свойство указывает интервал, подлежащий округлению. Применяется только если ThemeProperties.RoundRanges равно True.

Примечание

Если RoundBy равно 100, и вычисляемые диапазоны равны 420-560 и 560-930, то они будут округлены до 400-600 и 600-900 соответственно. По умолчанию величина RoundBy рассчитывается, нахождением степени 10 которая перекрывает разницу между минимумом и максимумом в данных. Например, если разница между минимумом и максимумом была 11554, то по умолчанию значение для свойства RoundBy будет 10000.

Смотрите также

Свойство ThemeProperties.RoundRanges

Свойство ThemeProperties.InflectRanges (Объект ThemeProperties)

Назначение

Логическое свойство применяемое к Диапазнам. В карте диапазонов точка перелома может использоваться для разделения данных в диапазонах на два раздела. Когда InflectRanges это true, цвета диапазона будут разбросаны от цвета верхнего диапазона до точки перелома цвета InflectionColor, затем от InflectionColor до цвета нижнего диапазона. По умолчанию равно False.

Примечание

Например, если InflectionColor это miColorWhite, и цвет верхнего диапазона синий, а нижнего красный, то теперь цвет будет изменяться от синего до белого и от белого до красного. Если InflectRanges было False, то цвет в середине будет средним между синим и красным.

Смотрите также

Свойство ThemeProperties.InflexionRange

Свойство ThemeProperties.InflexionColor

Свойство ThemeProperties.RangeCategories

Свойство ThemeProperties.InflexionRange (Объект ThemeProperties)**Назначение**

Короткое целое, применяемое к Диапазнам. Оно определяет, какой диапазон принимает цвет, определенный точкой перелома InflexionColor. Диапазоны выше и ниже точки перелома InflexionRange получают смешанные цвета.

Примечание

Это свойство применяется если ThemeProperties.InflexRanges это True. По умолчанию 2 (второй диапазон).

Смотрите также

Свойство ThemeProperties.InflexRanges

Свойство ThemeProperties.InflexionColor

Свойство ThemeProperties.InflexionColor (Объект ThemeProperties)**Назначение**

Значение типа OLE_COLOR применяемое к Диапазнам, для которых ThemeProperties.InflexRanges равно True. Оно указывает цвет переломного диапазона. Переломный диапазон указывается свойством ThemeProperties.InflexionRange. По умолчанию это miColorWhite.

Смотрите также

Свойство ThemeProperties.InflexRanges

Свойство ThemeProperties.InflectionColor

Свойство ThemeProperties.ColorMethod (Объект ThemeProperties)

Назначение

Короткое целое, применяющееся к Диапазнам. Это должна быть одна из констант ColorSpreadingMethodConstants. Она определяет метод, используемый для интерполяции между цветами верхнего и нижнего диапазонов для получения цветов промежуточных диапазонов. Свойство действует только когда ThemeProperties.SpreadBy установлено на miSpreadByColor.

Смотрите также

Свойство ThemeProperties.SpreadBy

Свойство ThemeProperties.ApplyAttribute (Объект ThemeProperties)

Назначение

Короткое целое, которое принимает одно из значений ApplyAttributeConstants. Свойство применяется к Диапазнам или Индивидуальным значениям. Оно определяет, какие атрибуты стиля диапазонов применяются к тематической карте. Свойство применяется для избирательного изменения или цвета или размера индивидуальных значений, без воздействия на другие атрибуты. По умолчанию это miApplyAttributeAll.

Смотрите также

Свойство ThemeProperties.RangeCategories

Коллекция RangeCategories

Свойство ThemeProperties.IndividualValueCategories

Коллекция IndividualValueCategories

Объект Title

Каждая карта содержит заголовок (Свойство Map.Title) в котором хранятся текст и стиль заголовка, а так же местоположение на карте.

Свойства объекта

- Свойство Title.Border
- Свойство Title.Caption
- Свойство Title.Editable
- Свойство Title.Position
- Свойство Title.TextStyle
- Свойство Title.Visible
- Свойство Title.X
- Свойство Title.Y

Пример на C++

```
// Title.Editable Property
// Title.Position Property
// Title.X Property
// Title.Y Property
void CSampleProjectView::PositionTitle() {
    try {
        CMapXTitle MapTitle = m_Map.GetTitle();

        // Place the title at the bottom center of the screen
        MapTitle.SetEditable(FALSE);
        MapTitle.SetX(m_Map.GetMapScreenWidth() / 2);
        MapTitle.SetY(m_Map.GetMapScreenHeight());

        // The title will be placed directly below the X,Y value
```

```

        MapTitle.SetPosition(miPositionBC);
    } catch (ColeDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (ColeException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Пример на Visual Basic

```

Private Sub PositionTitle()
    ' Place the title at the bottom center of the screen

    Map1.Title.Editable = False
    Map1.Title.X = Map1.MapScreenWidth / 2
    Map1.Title.Y = Map1.MapScreenHeight

    ' The title will be placed directly below the X,Y value
    Map1.Title.Position = miPositionBC
End Sub

```

Свойство Title.Border (Объект Title)

Назначение

Определяет, отображается ли рамка вокруг заголовка. Это логическая величина, по умолчанию равная True.

Свойство Title.Caption (Объект Title)

Назначение

Содержит текст заголовка Title. Это строковая величина.

Свойство Title.Editable (Объект Title)

Назначение

Определяет, является ли заголовок редактируемым для пользователя. Под этим подразумевается возможность щелкнуть мышкой на объекте заголовок, перемещать его, менять размер или менять текст. Логическая величина и по умолчанию она равна True.

Свойство Title.Position (Объект Title)

Назначение

Содержит значение, показывающее как должен быть расположен текст относительно координат X,Y. Берется значение PositionsConstants и по умолчанию оно равно miPositionCC.

Свойство Title.TextStyle (Объект Title)

Назначение

Объект Style содержащий стиль заголовка.

Свойство Title.Visible (Объект Title)

Назначение

Определяет, является ли заголовок видимым. Это логическая величина, по умолчанию равная True.

Свойство Title.X (Объект Title)

Назначение

Содержит X координату заголовка. Это величина типа Float, и представляет собой экранные координаты. Позиция заголовка определяется координатами X,Y совместно со свойством Position, которое определяет как заголовок расположен относительно координат X,Y.

Свойство Title.Y (Объект Title)

Назначение

Содержит Y координату заголовка. Это величина типа Float, и представляет собой экранные координаты. Позиция заголовка определяется координатами X,Y совместно со свойством Position, которое определяет как заголовок расположен относительно координат X,Y.

События MapX

Пользовательские события

MapX поддерживает следующие пользовательские события:

Событие	Dispatch ID	Вызывается...
Событие AddFeatureToolUsed	16	когда используется инструмент Add Feature.
Событие AnnotationAdded	7	когда добавляется аннотация.
Событие AnnotationChanged	8	когда аннотация добавляется, изменяется или удаляется.
Событие DataMismatch	5	во время связывания данных, если строки не сравниваются
Событие DrawUserLayer	10	когда окно карты нуждается в обновлении и приложение создает

		пользовательский (UserDraw) слой
Событие MapDraw	15	будет вызываться один раз перед рисованием и один раз после завершения рисования.
Событие MapInitialized	13	после завершения инициализации карты
Событие MapViewChanged	6	при изменении минимального описывающего прямоугольника (изменении масштаба или перемещении карты)
Событие MouseWheel	12	когда пользователь использует колесо манипулятора мышь
Событие PolyToolUsed	11	когда пользователь использует инструментарий отрисовки полилиний
Событие RequestData	4	при вызове запрашиваемых данных по одному значению при работе с несвязанными данными
Событие ResolveDataBind	2	во время автоматического связывания данных, если данные раздваиваются
Событие ResolveDataBindEx	14	имеет ту же функцию, что и событие ResolveDataBind, но посылает коллекцию ResolveObjects вместо массива строк.
Событие SelectionChanged	1	когда выборка изменяется
Событие ThemeModifyRequested	9	когда пользователь дважды щелкнет на легенде
Событие ToolUsed	3	когда пользователь использует инструментарий для работы с картой

События, связанные с обработкой команд клавиатуры и мыши

MapX поддерживает следующие события:

Событие	OLE Dispatch ID	Описание
Click	-600	Пользователь нажимает кнопку мышки.
DblClick	-601	Двойной щелчек.

KeyDown	-602	Нажатие клавиши.
KeyPress	-603	Пользователь нажимает клавишу(и не отпускает).
KeyUp	-604	Пользователь отпускает клавишу.
MouseDown	-605	Пользователь нажимает (и не отпускает) кнопку мышки.
MouseMove	-606	Пользователь перемещает мышку.
MouseUp	-607	Пользователь отпускает кнопку мышки.
Error	-608	Ошибка происходит во время асинхронной активизации, такой как перерисовка или когда пользователь использует инструментарий для работы с картой.

В зависимости от среды разработки, Вы можете использовать другие события. Например, Visual Basic обеспечивает обработку следующих событий: DragDrop, DragOver, GotFocus, LostFocus.

Событие AddFeatureToolUsed

Это событие вызывается когда пользователь использует один из стандартных инструментов создания объектов (miAddPointTool, miAddLineTool, miAddPolylineTool или miAddRegionTool). Поведение этого события похоже на событие PolyToolUsed.

Dispatch ID = 16

Синтаксис

AddFeatureToolUsed(ByVal ToolNum As Integer, ByVal Flags As Long, Feature As Object, ByVal Shift As Boolean, ByVal Ctrl As Boolean, EnableDefault As Boolean)

Часть	Описание
ToolNum	Указывает номер инструмента Add Feature.
Flags	Величина ToolFlagConstants; указывающая, стартует ли пользователь, находится ли в процессе или завершает использование инструмента.
Feature	Объект Feature который создал пользователь.
Shift, Ctrl	Логические величины, указывающие, нажимал ли пользователь клавиши (SHIFT, CTRL) в момент самого события; True означает,

что клавиша была нажата.

EnableDefault Логическое. Этот параметр определяет, позволяет ли MapX действовать инструменту стандартным образом. По умолчанию это True. Для избежания стандартного поведения инструмента, установите параметр на False. Этот параметр действует только когда пользователь завершает использование инструмента (miToolEnd).

Примечание

Это событие возникает многократно во время каждого использования инструментов создания объектов:

Один раз в начале использования инструмента, когда пользователь щелкает мышкой на месте первого узла объекта. Во время этого вызова, переменная Flags устанавливается на miToolBegin (0).

Для инструментов Add Polyline и Add Region, событие возникает каждый раз при добавлении новых узлов, и каждый раз когда пользователь удаляет узел нажатием клавиши BACKSPACE. Во время этого вызова переменная Flags имеет значение равное miToolInProgress (3).

Событие также возникает в конце использования инструмента, когда пользователь делает следующее:

- Завершает рисование объекта двойным щелчком мышки (инструменты полилинии и полигона) или отпускает кнопку мышки (инструменты линия и точка), в этом случае переменная Flags будет установлена на miToolEnd (1);
- Нажимает клавишу ESC для завершения работы инструмента добавления объекта, в этом случае переменная Flags будет установлена на miToolEndEscaped (2);
- Использует клавишу backspace для удаления всех узлов, в этом случае переменная Flags будет установлена на miToolEndEscaped (2).

Событие AnnotationAdded

Назначение

Это событие, которое вызывается после добавления аннотации. Оно позволяет программисту увидеть то что было добавлено.

Dispatch ID = 7

Синтаксис

AnnotationAdded

указатель объекта для объекта аннотации

Пример на Visual Basic

```
Private Sub Map1_AnnotationAdded(ByVal Annotation As Object)
    Select Case Annotation.Type
        Case miSymbolAnnotation
            Debug.Print "Symbol Annotation Added"
        Case miTextAnnotation
            Debug.Print "Text Annotation Added."
    End Select
    Debug.Print "    There are " & Map1.Annotations.Count & _
        " annotations."
End Sub
```

Пример на C++

```
void CSampleProjectView::OnAnnotationAdded(LPDISPATCH Annotation_Disp) {
    CMapXAnnotation annotation;
    // To attach an LPDISPATCH to a MapX object, use
    // the object.AttachDispatch(Dispatch,FALSE); Sintaxis
    annotation.AttachDispatch(Annotation_Disp,FALSE);
    try {
        switch(annotation.GetType()) {
            case miSymbolAnnotation:
                AfxMessageBox("Symbol annotation added.");
                break;
```

```

        case miTextAnnotation:
            AfxMessageBox("Text annotation added.");
            break;
    }
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Событие AnnotationChanged

Назначение

Это событие, которое вызывается когда аннотация добавляется, изменяется или удаляется. Правильнее вызывать это событие перед изменением аннотации, что дает программисту возможность предотвратить операцию.

Dispatch ID = 8

Внимание: Событие происходит только при изменении действий пользователя (типа выбора аннотации, перемещения аннотации, изменении ее размера и др.), NOT если происходят программные изменения в аннотациях. Например, в следующем случае не произойдет события AnnotationChanged: annotation.X = 100. Хотя аннотация перемещается.

Синтаксис

AnnotationChanged

int changetype { ADD, DEL, MOD }

object pointer to the annotation object

int EnableDefault

Пример на Visual Basic

```
Private Sub Map1_AnnotationChanged(ByVal ChangeType As Integer, _ ByVal
Annotation As Object, EnableDefault As Boolean)

Dim sType As String

    Select Case Annotation.Type

        Case miSymbolAnnotation

            sType = "Symbol"

        Case miTextAnnotation

            sType = "Text"

    End Select

Select Case ChangeType

    Case miAddAnnotation

        Debug.Print "About to add a " & sType & " annotation"

    Case miDeleteAnnotation

        Debug.Print "I will disallow the deletion of this " _
            & sType & " annotation"

        EnableDefault = False `Prevent annotation from being _
            changed

    Case miEditAnnotation

        Debug.Print "About to change a " & sType & " annotation"

    Case miSelectAnnotation

        Debug.Print "About to select a " & sType & " annotation"

    End Select

End Sub
```

Пример на C++

```

void CSampleProjectView::OnAnnotationChanged(short ChangeType,
LPDISPATCH Annotation_Disp, BOOL FAR* EnableDefault) {

    CMapXAnnotation annotation;

    CString sType, text;

    annotation.AttachDispatch(Annotation_Disp,FALSE);

    try {

        switch(annotation.GetType()) {

            case miSymbolAnnotation:

                sType = "Symbol"; // Symbol annotation added

                break;

            case miTextAnnotation:

                sType = "Text"; // Text annotation added

                break;

        }

        switch(ChangeType) {

            case miAddAnnotation:

                text = "About to add a "; text += sType; text += " annotation";

                break;

            case miDeleteAnnotation:

                text = "I will disallow the deletion of this"; text += sType; text
+= " annotation";

                (*EnableDefault) = FALSE; // Prevent the annotation from being
changed

                break;

            case miEditAnnotation:

```

```

        text = "About to change a "; text += sType; text += "
annotation";

        break;

    case miSelectAnnotation:

        text = "About to select a "; text += sType; text += " annotation";

        break;

    }

} catch (COleDispatchException *e) {

    e->ReportError();

    e->Delete();

} catch (COleException *e) {

    e->ReportError();

    e->Delete();

}

AfxMessageBox(text);

}

```

Событие DataMismatch

Назначение

Событие вызывается когда строка не совпадает с объектом на карте во время связи данных.

Поскольку MapX знает геоколонку для набора данных, он может затем привязать ее к карте. Таким образом, MapX может подсчитать строки, несовпадающие с объектом на карте. По умолчанию, это будет просто пропущено. Если обработчик события DataMismatch определен, то событие будет отработано.

Dispatch ID = 5

Синтаксис

DataMismatch

int rownum

string geocolvalue

Часть Описание

rownum Номер строки в источнике данных, которая не совпадает с объектом.

GeoColValue Значение поля, которое не совпадает.

Пример на Visual Basic

```

Private Sub Command1_Click()

    Dim i As Integer

    Dim flds As New MapXLib.Fields

    Dim ds As Dataset

    mapData(1, 1) = "ME"
    mapData(2, 1) = "NH"
    mapData(3, 1) = "VT"
    mapData(4, 1) = "MA"
    mapData(5, 1) = "RL"      `This will trigger the mismatch
    mapData(6, 1) = "CT"
    mapData(7, 1) = "NY"

    For i = 1 To 7
        mapData(i, 2) = i * 100
    Next

    flds.Add "State", "State", miAggregationIndividual, _
    miTypeString

    flds.Add "Value", "Value", miAggregationSum, miTypeNumeric

```



```

    Set ds = Map1.Datasets.Add(miDataSetUnbound, Nothing, _
    "My Dataset", "State", , "USA", flds)

    ds.Themes.Add

End Sub

Private Sub Map1_DataMismatch(ByVal DataSetName As String, _
ByVal Row As Long, GeoFieldValue As String)

    Debug.Print "Unmatched Value: " & GeoFieldValue

    Debug.Print "          In Row: " & Row

    Debug.Print "    will change to ""RI""

    GeoFieldValue = "RI"

End Sub

Private Sub Map1_RequestData(ByVal DataSetName As String, _
ByVal Row As Long, _
ByVal Field As Integer, _
Value As Variant, Done As Boolean)

    Done = False

    If DataSetName <> "My Dataset" Or Row > 7 Then

        Done = True

    Else

        Value = mapData(Row, Field)

    End If

End Sub

```

Пример на C++

```

struct StateData { char name[3]; int value; };

```

```

StateData MapData[7] = {
    {"ME",100},
    {"NH",200},
    {"VT",300},
    {"MA",400},
    {"RL",500}, // This will trigger the mismatch
    {"CT",600},
    {"NY",700}};

void CSampleProjectView::Start() {
    ColeVariant FieldsVariant;
    CMapXFields flds;
    CMapXDataset ds;

    if(!flds.CreateDispatch(flds.GetClsid())) {
        TRACE0("Failed to Create Fields object");
        return;
    }

    try {
        flds.Add(ColeVariant("State"),ColeVariant("State"),
            ColeVariant((long)miAggregationIndividual),
            ColeVariant((long)miTypeString));

        flds.Add(ColeVariant("Value"),ColeVariant("Value"),
            ColeVariant((long)miAggregationSum),
            ColeVariant((long)miTypeNumeric));
    }
}

```

```

FieldsVariant.vt = VT_DISPATCH;

FieldsVariant.pdispVal = flds.m_lpDispatch;


ds = m_Map.GetDatasets().Add(miDataSetUnbound,
COptionalVariant(),COleVariant("My Dataset"),
COleVariant("State"),COptionalVariant(),
COptionalVariant("USA"),FieldsVariant,
COptionalVariant());

ds.GetThemes().Add();
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

void CSampleProjectView::OnRequestData(LPCTSTR DataSetName, long Row,
short Field, VARIANT FAR* Value, BOOL FAR* Done) {
    *Done = FALSE;

    if(!strcmp(DataSetName,"My Dataset") || Row > 7)
        *Done = TRUE;
    else {
        if(Field == 1) {
            Value->vt = VT_BSTR;

            Value->bstrVal = CString(MapData[Row].name).AllocSysString();
        } else {

```

```

        Value->vt = VT_INT;

        Value->iVal = MapData[Row].value;
    }
}
}

void CSampleProjectView::OnDataMismatch(LPCTSTR DataSetName, long Row,
BSTR FAR* GeoFieldValue) {
    CString DebugString;

    DebugString.Format("Unmatched Value: %s\nIn row: %d\nWill change to
\\\"RI\\\"\"", *GeoFieldValue, Row);

    *GeoFieldValue= CString("RI").AllocSysString();

    AfxMessageBox(DebugString);
}

```

Событие DrawUserLayer

Назначение

Когда приложение создает пользовательский слой (UserDraw) с использованием метода AddUserDrawLayer Коллекции Layers, событие произойдет в тот момент, когда окно карты будет нуждаться в обновлении:

Dispatch ID = 10

Синтаксис

DrawUserLayer (Layer, hOutputDC, hAttributeDC, RectFull, RectInvalid)

Часть

Описание

Layer	Объект Layer: слой для рисования.
hOutputDC	OLE_HANDLE: устройство для рисования.
hAttributeDC	OLE_HANDLE: устройство для рисования; обычно тоже что и hOutputDC, кроме случая рисования в устройство metafile.
RectFull	Объект Rectangle: полное окно.

RectInvalid Объект Rectangle: недоступная область.

Прямоугольники это объекты типа Rectangle. Координаты в прямоугольниках являются экранными.

Примечание

Вы можете также добавлять UserDrawLayer через Layers.Add.

Пример на Visual Basic

```
' API DEFS should be declared in a separate module
Declare Function MoveToEx Lib "gdi32" Alias "MoveToEx" _
    (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, lpPoint _
    As POINTAPI) As Long
Declare Function LineTo Lib "gdi32" Alias "LineTo" (ByVal hdc _
    As Long, ByVal x As Long, ByVal y As Long) As Long
Declare Function SetMapMode Lib "gdi32" Alias "SetMapMode" _
    (ByVal hdc As Long, ByVal nMapMode As Long) As Long
Type POINTAPI
    x As Long
    y As Long
End Type
Public Const MM_TWIPS = 6
' this sets the UserDraw Layer to "My Layer"
Dim lyr as Layer
Set lyr = Map1.Layers.AddUserDrawLayer("My Layer", 1)
' this example draws a line between the corners of Wyoming
Private Sub Map1_DrawUserLayer(ByVal Layer As Object, ByVal _
    hdc As Long, ByVal RectFull As Object, ByVal RectInvalid As _
```

```

Object)

Dim pt As POINTAPI

SetMapMode hDC, MM_TWIPS

dim PX as single
dim PY as single

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

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

End Sub

```

Пример на C++

```

void CSampleProjectView::CreateDrawLayer() {
    try
        m_Map.GetLayers().AddUserDrawLayer("My Layer",1);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

```

    }
}

void CSampleProjectView::OnDrawUserLayer(LPDISPATCH Layer, long
hOutputDC, long hAttributeDC, LPDISPATCH RectFull, LPDISPATCH
RectInvalid)
{
    // This example draws a line between the corners of Wyoming
    CDC dc;

    CPen pen,*oldPen;

    float PX,PY;

    double X1,Y1,X2,Y2;

    pen.CreatePen(0,1,(COLORREF)0);

    dc.Attach((HDC)hOutputDC); // Attach the hDC to the MFC object dc
    dc.SetAttribDC((HDC)hAttributeDC);

    dc.SetMapMode(MM_TWIPS);

    //Set pen to black
    oldPen = dc.SelectObject(&pen);

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

```

```

try {
    m_Map.ConvertCoord(&PX,&PY,&X1,&Y1,miMapToScreen);
    dc.MoveTo((int)PX,(int)-PY);

    m_Map.ConvertCoord(&PX,&PY,&X2,&Y2,miMapToScreen);
    dc.LineTo((int)PX,(int)-PY);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}

dc.SelectObject(oldPen);

dc.ReleaseAttribDC(); // Release the hDC
dc.Detach();
}

```

Событие MapDraw

Это событие будет вызываться один раз перед рисованием, и один раз после завершения рисования. Это один параметр, определяющий состояние начала или конца рисования.

Disp ID = 15

Синтаксис

MapDraw (Flag)

Flag: Или miDrawBegin или miDrawEnd.

Смотрите также

MapDrawConstants

Событие MapInitialized**Назначение**

Событие вызывается немедленно после завершения инициализации карты. Это служит для извещения контейнера приложения, что наступило подходящее время для необходимых конфигураций запуска программы.

Dispatch ID = 13

Синтаксис

MapInitialized < no parameters >

Событие MapViewChanged**Назначение**

Это событие вызывается, когда меняется масштаб карты или перемещается центр карты.

Dispatch ID = 6

Синтаксис

MapViewChanged < no parameters >

Пример на Visual Basic

```
Private Sub Map1_MapViewChanged()
```

```

Debug.Print "The map view has changed."

    Debug.Print "  New center: " & Map1.CenterX & ", " _
    & Map1.CenterY

    Debug.Print "  New Zoom Level: " & Map1.Zoom

End Sub

```

Пример на C++

```

void CSampleProjectView::OnMapViewChanged() {
    CString msg;

    try {
        msg.Format("The map view has changed.\nNew Center: %lf, %lf\nNew
Zoom Level: %lf",m_Map.GetCenterX(),m_Map.GetCenterY(),m_Map.GetZoom());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }

    AfxMessageBox(msg);
}

```

Событие MouseWheel

Назначение

Intellimouse снабжен третьим колесом. В добавление к сообщениям об обычных кнопках мыши, колесо посылает сообщение MOUSEWHEEL когда оно вращается.

Mousewheel изменяет масштаб карты, когда крутится, прокручивает карту вверх и вниз, когда колесо крутится при нажатой клавише control, и обеспечивает режим автопрокрутки

”AutoScroll” когда средняя кнопка нажата и движется в сторону от точки, в которой оно было нажато. Скорость прокрутки пропорциональна расстоянию курсора от того места, где была нажата третья кнопка. Событие MouseWheel позволяет Вам изменять эти аспекты поведения Mousewheel.

Обратите внимание, что в Visual Basic под Windows 95 нет поддержки для Mousewheel, но технология ActiveX позволит Вам использовать возможности настроек Mousewheel.

Dispatch ID = 12

Синтаксис

MouseWheel(ByVal Flags As Long, ByVal zDelta As Integer, X As Single, Y As Single, EnableDefault As Boolean)

Часть	Описание												
Flags	Определяет, нажаты ли некоторые клавиши. Этот параметр может быть любой комбинацией следующих величин: <table><tr><th>Величина</th><th>Описание</th></tr><tr><td>MK_CONTROL</td><td>Устанавливается, если клавиша CTRL нажата.</td></tr><tr><td>MK_LBUTTON</td><td>Устанавливается, если нажата левая кнопка мыши.</td></tr><tr><td>MK_MBUTTON</td><td>Устанавливается, если нажата средняя кнопка мыши.</td></tr><tr><td>MK_RBUTTON</td><td>Устанавливается, если нажата правая кнопка мыши.</td></tr><tr><td>MK_SHIFT</td><td>Устанавливается, если нажата клавиша SHIFT.</td></tr></table>	Величина	Описание	MK_CONTROL	Устанавливается, если клавиша CTRL нажата.	MK_LBUTTON	Устанавливается, если нажата левая кнопка мыши.	MK_MBUTTON	Устанавливается, если нажата средняя кнопка мыши.	MK_RBUTTON	Устанавливается, если нажата правая кнопка мыши.	MK_SHIFT	Устанавливается, если нажата клавиша SHIFT.
Величина	Описание												
MK_CONTROL	Устанавливается, если клавиша CTRL нажата.												
MK_LBUTTON	Устанавливается, если нажата левая кнопка мыши.												
MK_MBUTTON	Устанавливается, если нажата средняя кнопка мыши.												
MK_RBUTTON	Устанавливается, если нажата правая кнопка мыши.												
MK_SHIFT	Устанавливается, если нажата клавиша SHIFT.												
zDelta	Значение слова старшего разряда wParam. Указывает дистанцию, на которую повернуто колесо, выражено увеличении или уменьшении WHEEL_DELTA, которое равно 120. Положительное значение указывает, что колесо вращается вперед, от пользователя; отрицательное значение указывает, что колесо крутится назад, к пользователю.												
X	Определяет x-координату указчика (pointer), относительно верхнего левого угла экрана.												
Y	Определяет y-координату указчика (pointer), относительно верхнего левого угла экрана.												
EnableDefault	Логическое. Этот параметр определяет, позволяет ли MapX стандартное поведение mousewheel. По умолчанию это True. Если												

программист устанавливает EnableDefault на FALSE, тогда настройки mousewheel не работают.

Событие PolyToolUsed

Событие вызывается когда пользователь рисует полилинию или полигон, используя или стандартный инструмент miPolygonSelectTool (1010) или пользовательский инструмент типа miToolTypePoly (4).

Dispatch ID = 11

Синтаксис

PolyToolUsed(ToolNum As Integer, Flags As Long, Pts As Object,
Shift As Boolean, Ctrl As Boolean, EnableDefault As Boolean)

Часть	Описание
ToolNum	Указывает номер пользовательского инструмента.
Flags	Величина PolyToolFlagConstants; указывающая в начале или конце процедуры использования инструмента находится пользователь.
Pts	Коллекция точек (такогоже вида как и в Feature.Parts.Add).
Shift, Ctrl	Логические величины, указывающие нажаты ли клавиши (SHIFT, CTRL) в момент возникновения события; True указывает, что клавиша нажата. Для стандартного инструмента miPolygonSelectTool, MapX использует состояние этих клавиш только когда пользователь завершил использование инструмента (т.е. когда параметр Flag это miPolyToolEnd).
EnableDefault	Логическое. Если это событие произошло при стандартном miPolygonSelectTool, то этот параметр определяет позволяет ли MapX использовать стандартное поведение инструмента. По умолчанию это True. Для избегания стандартного поведения инструмента (т.е. отмены действий пользователя), установите этот параметр на False. Этот параметр действует только если пользователь завершил использование инструмента (miPolyToolEnd).

Примечание

Это событие возникает многократно во время каждого использования инструментов рисования полилинии или полигона.

Один раз в начале использования инструмента, когда пользователь щелкает мышкой на месте первого узла объекта. Во время этого вызова, переменная `Flags` устанавливается на `miPolyToolBegin (0)`.

Для инструментов `Add Polyline` и `Add Region`, событие возникает каждый раз при добавлении новых узлов, и каждый раз когда пользователь удаляет узел нажатием клавиши `BACKSPACE`. Во время этого вызова переменная `Flags` имеет значение равное `miToolInProgress (3)`.

Событие также возникает в конце использования инструмента, когда пользователь делает следующее:

- Завершает рисование объекта двойным щелчком мышки (инструменты полилинии и полигона) или отпускает кнопку мышки (инструменты линия и точка), в этом случае переменная `Flags` будет установлена на `miToolEnd (1)`;
- Нажимает клавишу `ESC` для завершения работы инструмента добавления объекта, в этом случае переменная `Flags` будет установлена на `miToolEndEscaped (2)`;
- Использует клавишу `backspace` для удаления всех узлов, в этом случае переменная `Flags` будет установлена на `miToolEndEscaped (2)`.

Использование пользовательских инструментов рисования полилиний и полигонов (Пример на Visual Basic)

Если Вы использовали метод `CreateCustomTool` создания инструментов рисования полилиний и полигонов, следующий пример показывает как обработать событие `PolyToolUsed`. Этот пример позволяет пользователю рисовать объекты полигон или линия на слое `Layer 1`.

```
Private Sub map1_PolyToolUsed(ByVal ToolNum As Integer, _
    ByVal Flags As Long,
    ByVal pts As Object, ByVal Shift As Boolean, ByVal Ctrl As _
    Boolean, EnableDef As Boolean
    If Flags = miPolyToolBegin Then
```

```

    'Someone's beginning the use of a PolyTool..
.    ElseIf Flags = miPolyToolEnd Then
    ' The user finished using a PolyTool by double clicking
If ToolNum = MY_SUPER_POLYGON_TOOL Then
    ' They used MY_POLYGON_TOOL! Make a new
    ' region feature and add it to the first layer
Dim f As New Feature
f.Type = miFeatureTypeRegion
    f.Style.RegionPattern = miSolid
    f.Style.RegionColor = 255
    f.Style.RegionBorderColor = 0
    f.Style.RegionBorderWidth = 2
    f.Style.RegionBorderStyle = 1
    'Use the points that were given to me by the event.
    'MapX automatically closes open polygons, so I don't
    'have to worry about that!
f.Parts.Add pts
map1.Layers(1).AddFeature f
    ElseIf ToolNum = MY_SUPER_POLYLINE_TOOL Then
    ' They used MY_POLYLINE_TOOL! Make a new
    ' line feature and add it to the first layer!
    Dim f As New Feature
    f.Type = miFeatureTypeLine
    f.Style.LineStyle = 1
    f.Style.LineColor = 255
    f.Style.LineWidth = 2

```

```

        'Use the points that were given to me by the event
    .       f.Parts.Add pts

        map1.Layers(1).AddFeature f

    End If

    ElseIf Flags = miPolyToolEndEscaped Then
        ' The users hit 'Esc' or backspaced all the nodes
        ' away... don't add anything in
    .     End If

End Sub

```

Пример на C++

```

const short CUSTOM_POLYGON_TOOL = 42;
const short CUSTOM_POLYLINE_TOOL = 43;

void CSampleProjectView::OnPolyToolUsed(short ToolNum, long Flags,
LPDISPATCH Points, BOOL bShift, BOOL bCtrl, BOOL FAR* EnableDefault) {

    if(Flags==miPolyToolEnd) {

        // The user double clicked to end the PolyTool

        if(ToolNum==CUSTOM_POLYGON_TOOL) {

            // Make a new region feature and add it to the first layer

            CMapXFeature f;

            CMapXPoints pts;

            pts.AttachDispatch(Points,FALSE);

            if(!f.CreateDispatch(f.GetClsid())) {

                TRACE0("Failed to Create Point object");

                return;
            }
        }
    }
}

```

```

    }

    try {

        f.Attach(m_Map.GetDispatch());

        // Setup the style of the polygon
        f.SetType(miFeatureTypeRegion);
        f.GetStyle().SetRegionPattern(miPatternSolid);
        f.GetStyle().SetRegionColor(255);
        f.GetStyle().SetRegionBorderColor(0);
        f.GetStyle().SetRegionBorderWidth(2);
        f.GetStyle().SetRegionBorderStyle(1);

        // Now add the points that were given by the event
        f.GetParts().Add(pts);
        m_Map.GetLayers().Item(1).AddFeature(f);
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
} else if (ToolNum==CUSTOM_POLYLINE_TOOL) {
    CMapXFeature f;
    CMapXPoints pts;
    pts.AttachDispatch(Points,FALSE);

```



```

if(!f.CreateDispatch(f.GetClsid())) {
    TRACE0("Failed to Create Feature object");
    return;
}

try {
    f.Attach(m_Map.GetDispatch());

    // Set up the style of the polyline
    f.SetType(miFeatureTypeLine);
    f.GetStyle().SetLineStyle(1);
    f.GetStyle().SetLineColor(255);
    f.GetStyle().SetLineWidth(2);

    // Now add the points that were given by the event
    f.GetParts().Add(pts);
    m_Map.GetLayers().Item(1).AddFeature(f);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}

}

} // Other possible Flag values:

// miPolyToolBegin          The user began the use of the tool

```

```

        // miPolyToolEndEscaped    The user cancelled the tool

        // miPolyToolInProgress The user added a new node to the polyline or
        deleted a node using BACKSPACE
    }

```

Событие RequestData

Назначение

Это событие вызывается для запроса данных по одному значению при работе с несвязанными данными.

Dispatch ID = 4

Синтаксис

RequestData

int row

int column

variant value

int done

Установите флаг “сделано” если больше нет данных для передачи в контейнер.

Пример на Visual Basic

```

` This sample demonstrates the use of Datasets.Add and
` the RequestData
` event supported by MapX to build an unbound dataset, which
` allows MapX to access data whose format is known only to the
` programmer. In this example, the data is in the form of a
` two-dimensional array in VB.

Private Const kNumberOfRows = 3

```

```

Dim theData(1 To 3, 1 To 2) As Variant

Private Sub Form_Load()

    theData(1, 1) = "ME"           `Fill in the data to
    theData(2, 1) = "NH"           `be used by RequestData
    theData(3, 1) = "VT"

    theData(1, 2) = 100
    theData(2, 2) = 200
    theData(3, 2) = 300

    Map1.ZoomTo 800, -70.26, 44.05    `Zoom in on New England

End Sub

Private Sub Command1_Click()

    Dim flds As New MapXLib.Fields
    Dim ds As Dataset

    `Describe the structure of the unbound dataset:
    flds.Add "State", "State", miAggregationIndividual,
    miTypeString
    flds.Add "Sales", "Sales", miAggregationSum, miTypeNumeric

    `Create the unbound dataset. The "RequestData" event will be
    `triggered to get the data to be used.
    Set ds = Map1.Datasets.Add(miDatasetUnbound, Nothing, _
        "My Dataset", "State", , "USA", flds)

    `Create a theme based on the "Sales" column in the unbound dataset
    ds.Themes.Add miThemeGradSymbol, "Sales", "My Theme"

End Sub

```

```

Private Sub Map1_RequestData(ByVal DataSetName As String, _
ByVal Row As Long, ByVal Field As Integer, _
Value As Variant, Done As Boolean)

    Done = False

    If DataSetName <> "My Dataset" Or Row > kNumberOfRows Then

        Done = True

    Else

        Value = theData(Row, Field)

    End If

End Sub

```

Пример на C++

```

struct StateData { char name[3]; int value; };

StateData MapData[3] = {

    {"ME",100},

    {"NH",200},

    {"VT",300}};

void CSampleProjectView::Start() {

    ColeVariant FieldsVariant;

    CMapXFields flds;

    CMapXDataset ds;

    // Create a new, empty fields collection

    if(!flds.CreateDispatch(flds.GetClsid())) {

        TRACE0("Failed to Create Fields collection");

        return;

    }

```

```

    }

    try {

        flds.Add(ColeVariant("State"),ColeVariant("State"),
        ColeVariant((long)miAggregationIndividual),
        ColeVariant((long)miTypeString));

        flds.Add(ColeVariant("Value"),ColeVariant("Value"),
        ColeVariant((long)miAggregationSum),
        ColeVariant((long)miTypeNumeric));

        FieldsVariant.vt = VT_DISPATCH;

        FieldsVariant.pdispVal = flds.m_lpDispatch;

        ds = m_Map.GetDatasets().Add(miDataSetUnbound,
        COptionalVariant(),ColeVariant("My Dataset"),
        ColeVariant("State"),COptionalVariant(),
        ColeVariant("USA"),FieldsVariant,COptionalVariant());

        ds.GetThemes().Add(miThemeGradSymbol,"Sales","My Theme");
    } catch (ColeDispatchException *e) {

        e->ReportError();

        e->Delete();
    } catch (ColeException *e) {

        e->ReportError();

        e->Delete();
    }
}

void CSampleProjectView::OnRequestData(LPCTSTR DataSetName, long Row,
short Field, VARIANT FAR* Value, BOOL FAR* Done) {

```

```

*Done = FALSE;

if(strcmp(DataSetName,"My Dataset") || Row > 3)

    *Done = TRUE;

else {

    if(Field == 1) {

        Value->vt = VT_BSTR;

        Value->bstrVal = CString(MapData[Row].name).AllocSysString();

    } else {

        Value->vt = VT_INT;

        Value->iVal = MapData[Row].value;

    }

}

}

```

Событие ResolveDataBind

Назначение

Automatic Data Binding может подсчитывать двусмысленные ситуации. Такие ситуации включают повторяющиеся геоколонки в данных пользователя, множественные слои, совпадающие с геоколонками, множественные наборы данных для одного слоя. В этих случаях, MapX будет брать первый попавшийся, кроме случая, когда определено последующее событие ResolveDataBind. В этом случае, MapX вызовет это событие и позволит обработчику выбрать один из возможных путей для выбора нужного набора данных, геоколонок или слоев.

Dispatch ID = 2

Синтаксис

ResolveDataBind(flag, nummatches, matches, choice, cancel)

Часть

Описание

flag

Определяет причину по какой было вызвано событие:
множественные геоколонки (0), множественные слои (1),

	множественные наборы данных (2).
nummatches	Определяет сколько вхождений в массиве совпадений. Массив строковых совпадений содержит список совпадений..
matches	Переменная: Это массив строк.
choice	Возвращаемая переменная. 1 базовое значение. 0 означает автоматический выбор и позволяет MapX сделать лучший выбор).
cancel	Позволяет пользователю отменять связь с данными.

Примечание

ResolveDataBind может вызываться три раза.

Первый раз событие выбирает колонку в исходной таблице, затем для данной колонки выбирает таблицу если значение > 1 , и наконец, для таблицы выбирает рабочие наборы > 1 .

The geodict list geosets for each table.0

Событие ResolveDataBindEx

Назначение

Событие имеет ту же функцию, что и событие ResolveDataBind, но посылает коллекцию ResolveObjects вместо массива строк. Так для ситуации когда обнаружено несколько геоколонок для данных, множественные слои, совпавшие для геоколонок или множественные наборы данных допустимы для совпадающего слоя карты, MapX будет брать первый из них, если не определены обработчики событий ResolveDataBind или ResolveDataBindEx. Если определены обработчики, MapX позволит обработчику выбрать один из них.

Dispatch ID = 14

Синтаксис

ResolveDataBindEx (flag, nummatches, matches, choice, cancel)

Часть	Описание
flag	Указывает событие которое будет вызвано в случае нескольких геоколонок. Этот флаг всегда устанавливается на ноль.
nummatches	Указывает сколько встречено совпадений в коллекции совпадений.
matches	Переменная: Представляет коллекцию объектов ResolveObject.
choice	Возвращаемая переменная. 1 базовое значение. 0 означает автовыборку и позволяет MapX выбрать наилучший вариант.
cancel	Отмена связывания данных.

Примечание

ResolvedatabindEx действует только для множественных геоколонок и только если есть установка на ноль во время события ResolveDataBind.

Событие SelectionChanged

Назначение

Событие вызывается если произошли изменения в выборке и стало возможным воздействие контейнера на выборку сделанную с карты. Выборка может изменяться в результате применения пользователем инструмента выбора или при использовании одного из методов Selection объекта Layer.

Dispatch ID = 1

Синтаксис

SelectionChanged

Пример

```

Private Sub Map1_SelectionChanged()
    Dim l As Layer
    Dim f As Feature
    Dim parNode As Node
    TreeView1.Nodes.Clear
    For Each l In Map1.Layers
        If l.Selection.Count > 0 Then
            Set parNode = TreeView1.Nodes.Add(, , l.Name, l.Name)
            parNode.Expanded = True
            For Each f In l.Selection
                TreeView1.Nodes.Add parNode, tvwChild, parNode.Key & _
                    f.Name, f.Name
            Next
        End If
    Next
End Sub

```

Событие ThemeModifyRequested

Назначение

Событие вызывается, когда пользователь дважды щелкает на тематической легенде. Параметр это объект Theme который «владеет» легендой. Назначение этого события в способности писать код, позволяющий пользователю изменять тематику или легенду, когда пользователь дважды щелкнет на легенде.

Dispatch ID = 9

Синтаксис

ThemeModifyRequested

Theme as object

Пример на Visual Basic

```
Private Sub Map1_ThemeModifyRequested(ByVal Theme As Object)
    Debug.Print "User Double-Clicked the legend for the theme:"
    Debug.Print "    "" & Theme.Name & """"
End Sub
```

Пример на C++

```
void CSampleProjectView::OnThemeModifyRequested(LPDISPATCH
ThemeDispatch) {
    CMapXTheme theme;
    CString msg;
    theme.AttachDispatch(ThemeDispatch,FALSE);
    try {
        msg.Format("User Double-Clicked the legend for the
        theme:\n\"%s\"", theme.GetName());
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
    AfxMessageBox(msg);
}
```

Событие ToolUsed

Событие ToolUsed вызывается когда пользователь использует собственные инструменты работы с картой. Эта процедура позволяет Вам определить как используется инструмент; например, он сообщает текущие координаты курсора при щелканье клавишей мыши. Внутри процедуры ToolUsed, Вы пишете код, для выполнения нужных Вам функций таким инструментом.

Это событие также вызывается при использовании стандартных инструментов, в этом случае событие вызывается после действий пользователя, но перед действием, выполняемым стандартным инструментом.

Dispatch ID = 3

Синтаксис

ToolUsed(ToolNum, x1, y1, x2, y2, distance, shift, control, *EnableDefault)

Часть	Описание
ToolNum	Целое: Определяет номер инструмента. Обратите внимание, что если вы создаете много инструментов, это событие вызывается, когда используются любой из этих инструментов; таким образом Вам надо проверять параметр ToolNum для определения того, какой инструмент используется.
x1, y1	Двойная точность: Координаты того места на карте где был щелчек мышью.
x2, y2	Двойная точность: Координаты того места, где кончилось использование инструмента; не применяется к инструменту Point.
distance	Двойная точность: Расстояние в единицах карты, между начальным и конечным положением.
shift	Логическое: Определяет, нажата ли клавиша SHIFT.
control	Логическое: Определяет, нажата ли клавиша CTRL.
*EnableDefault	Логическое: В случае использования стандартного инструмента, этот параметр определяет, позволяет ли MapX иметь инструменту его стандартное поведение. По умолчанию равно True. Для избежания стандартного поведения(например отмена действий пользователя), установите этот параметр на False.

Использование пользовательских инструментов (Visual Basic пример)

Этот пример демонстрирует:

Использование метода Map.CreateCustomTool для создания пользовательского инструмента "line" (инструмент, позволяющий выбирать и перемещать линию).

Использование события ToolUsed для выполнения действия, когда пользователь использует собственный инструмент.

```
' The following code in the Form_Load event of a form with
' a Объект Map (Map1) creates a Custom Tool that is a
' "line-type" tool (it lets the user click and drag).
Private Sub Form_Load()
    Map1.CreateCustomTool 1, miToolTypeLine, miIconCursor
End Sub

' The following code is called whenever any tool is used on the map.
' If the Current Tool is #1 (created above) then the code within the
' If-Then statement will be true, and we will set the caption of
' a label "lblDistance" equal to the Distance passed in with the
' "line-type" tool.
Private Sub Map1_ToolUsed(ByVal ToolNum As Integer, ByVal X1 _
    As Double, ByVal Y1 As Double, ByVal X2 As Double, ByVal Y2 _
    As Double, ByVal Distance As Double, ByVal Shift As Boolean, _
    ByVal Ctrl As Boolean, EnableDefault As Boolean)
    If ToolNum = 1 then
        lblDistance.Caption = Str$(Distance)
    End If
End Sub
```

Использование пользовательских инструментов (Пример на C++)

Этот пример показывает как приложение:

- Изменяет стиль объекта карты там, где находится курсор
- Помещает новый символ там где пользователь щелкнет мышкой.

```
void CMapxSampleView::OnToolUsed(short ToolNum, double X1,
    double Y1, double X2, double Y2, double Distance,
    BOOL Shift, BOOL Ctrl, BOOL* EnableDefault)
{
    CString str;
    CMapXPoint pnt;

    str.Format("Tool=%d, [%f,%f] [%f, %f], dist=%f, %s %s\n",
        ToolNum, X1,Y1,X2,Y2,Distance, (Shift)?"Shift":"", (Ctrl)?"Ctrl":"");
    TRACE(str);

    // change the style of the feature under the cursor
    if (ToolNum == MAP_TOOL_CHANGESTYLE) {
        try {
            // Need the dispatch to use the point
            if (pnt.CreateDispatch(pnt.GetClsid())) {
                pnt.Set(X1, Y1);
            }

            else {
                // something went wrong, can't use the point...
                AfxThrowOleException(CO_E_CLASS_CREATE_FAILED);
            }

            CMapXLayers layers = m_ctrlMapX.GetLayers();
```

```

        // Get the USA feature under the cursor
        CMapXFeatures    ftrs =
layers.Item("USA").SearchAtPoint(LPDISPATCH(pnt));

        // work on only the first feature
        CMapXFeature    ftr = ftrs.Item(1);
        // get the style object from the feature

        CMapXStyle    style = ftr.GetStyle();
        style.SetRegionBackColor(255);
        // update the feature in the layer
        ftr.Update();
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

// place a new symbol at the point clicked on
else if (ToolNum == MAP_TOOL_NEWPOINT) {
    try {
        CMapXLayers    layers = m_ctrlMapX.GetLayers();
        CMapXFeature    ftr;
    }
}

```

```

// Need the dispatch id to use the feature
if (ftr.CreateDispatch(ftr.GetClsid())) {
    // Symbol feature
    ftr.SetType(miFeatureTypeSymbol);
    // Get the point object from the feature
    // and call the Set method
    ftr.GetPoint().Set(X1, Y1);
    // Add it to the layer
    layers.Item("USA").AddFeature(ftr);
}
else {
    AfxThrowOleException(CO_E_CLASS_CREATE_FAILED);
}
}
catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
}
catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}
}

```

Приложение А: Коды ошибок MapX

Это таблица ошибок MapX встречающихся при написании приложения MapX. Для информации об обработке ошибок на Visual Basic, Delphi и Powerbuilder, смотрите соответствующие справочные руководства. Для программистов на C++, смотрите Working With Visual C++.

Error	Описание
1000	Превышение размеров памяти.
1001	Свойства или метод не встроены.
1002	Свойство не поддерживает тематику такого типа.
1003	Object is no longer valid.
1004	По указанному Вами индексу объектов не обнаружено.
1005	Неправильно определен тип DataSet.
1006	Указано не уникальное имя. (имя уже используется другим объектом).
1007	Неправильно определено GeoField. Имя не найдено или индекс за пределами ранга.
1008	Неправильно определено вторичное геополе. Имя не найдено, или индекс за пределами ранга.
1009	Невозможно сравнить автоматически GeoField или BindLayer automatically.
1010	Неправильный параметр Fields.
1011	Неожидаемая ошибка в MapX.
1012	Неправильно определен тип Theme.
1013	Неожидаемая ошибка.
1014	Не найден ключ колонки в указанном наборе данных. Невозможно создать тематический слой.
1015	The DataBinding process was cancelled by setting the cancel property in the ResolveDataBind event.
1016	Invalid aggregation function specified.
1017	No source field specified.
1018	Invalid field type specified.
1019	Unable to refresh dataset.

1020	Cannot find the BindLayer for this dataset. Unable to create theme.
1021	Unable to create theme.
1022	Invalid ranges.
1023	Invalid number of ranges.
1024	Theme categories cannot be removed.
1025	Invalid distribution method.
1026	Invalid fill pattern.
1027	Invalid pen style.
1028	Invalid Fields Parameter - Name or index of a field not found.
1029	Could not create a default theme.
1030	This property is available at design time only.
1031	Layer specified is not matchable.
1032	Matched or specified Layer is not in any GeoSets.
1033	Invalid Name parameter specified.
1034	The color you specified is invalid.
1035	Invalid Spread by value.
1036	Invalid pen width.
1037	The Coordinate specified was not within the bounds of the map's Numeric CoordSys.
1038	Invalid Zoom specified. Zoom must be > 0.
1039	Invalid Tool specified. The tool number is invalid or doesn't exist, or the tool type or cursor is invalid.
1040	Invalid paper unit specified.
1041	Invalid mouse pointer specified.
1042	Invalid export format specified.
1043	Unable to create specified file.
1044	An ODBC error has occurred: <error>.
1045	Invalid ODBCQueryInfo object specified. Object not of correct type or SqlQuery not specified.
1046	Property not supported for annotations of this type.
1047	Invalid character code. Value must be between 0 and 255.
1048	Cannot create a theme of this type based on an attribute data column that has been aggregated by individual value.

1049	Invalid Row Parameter: Row out of range, or fetch past end of table.
1050	Invalid Column Parameter.
1051	Unable to find or load MIODBC.DLL
1052	Error initializing MIODBC.DLL Check DLL version.
1053	MIODBC.DLL called with an invalid handle.
1054	Dataset has no fields. No fields were successfully added to the dataset from the source data.
1055	Unable to obtain a temporary directory.
1056	Unable to create a filename for the new layer.
1057	Key column not specified, cannot create new layer.
1058	The specified reference layer is not installed in the geodictionary.
1059	Unable to create new layer. Layer filename may already exist.
1060	Invalid bind layer type specified.
1061	Invalid bind layer specified.
1062	Unable to complete this type of bind with the reference column(s) specified.
1063	Unknown error creating map.
1064	Invalid Search Time specified. Time must be between 1 and 3600 seconds.
1065	Display unit specified was not valid .
1066	Inalid Conversion Constant specified.
1067	Missing or Invalid Source, or Source doesn't match Type specified.
1068	The specified reference layer is not a point reference layer.
1069	Invalid dot size.
1070	Invalid position.
1071	The maximum size for a point theme object is 48.
1072	Unable to create geotable.
1073	Unable to create map layer.
1074	Unable to set or retrieve layer information.
1075	Unable to set or retrieve label information.
1076	Unable to set or retrieve layer style information.
1077	Unable to set or retrieve label style information.

1078	Feature object is no longer valid.
1079	Unable to find or load Helper Dialog DLL.
1080	Error initializing Helper Dialog DLL. Check DLL version.
1081	For the geofield specified or auto detected a secondary geofield is required, and one was not specified or auto detected.
1082	Fields Add or Remove not allowed after dataset has been created.
1083	Property or Method not supported by layers of this type.
1084	A Lotus Notes error has occurred: <error>.
1085	Invalid NotesViewInfo object specified. Object not of correct type or Database or View not specified.
1086	A numeric overflow or underflow has occurred during a coordinate conversion. Under Windows95, all coordinates are limited to the range of -32767 to +32767. The actual screen or map coordinates may be in range, but the conversion overflowed. See IsPointVisible or ClipLine.
1087	Attempt to set the Animation Layer to an object which is not a valid Объект Layer.
1088	Invalid Filespec parameter.
1089	Invalid DataSet specified.
1090	Feature Type Invalid or not applicable.
1091	Operation not allowed on a standalone feature object.
1092	Error updating feature object.
1093	Error adding feature object.
1094	Error deleting feature object.
1095	Feature property or method not valid for feature type.
1096	Error reading Feature from layer.
1097	A line must have at least 2 points. A region must have at least 3 points.
1098	Неправильно определен прямоугольник.
1099	Cannot find a specified Field in the DataSet.
1100	Неправильный объект Features.
1101	Неправильно определен тип Search.
1102	Неправильно определена точность Search.
1103	Неправильный объект Points.

1104	Неправильный объект Point.
1105	Неправильный объект Style.
1106	Попытка использовать новый объект Символ или Текст без предварительной установки свойства Point.
1107	Неправильно определены единицы измерения площади.
1108	Каждый объект Feature в коллекции Features должен быть с того же слоя.
1109	Ключевое поле на слое не найдено.
1110	The Layer that the Feature or Features object is from is no longer valid.
1111	Переменная не может быть конвертирована в подходящий тип.
1112	The reference layer specified has no matchable columns, and therefore cannot be used as a reference layer.
1113	Creating bitmap image failed on map export.
1114	Invalid Circle Type specified.
1115	Resolution must be ≥ 3
1116	Invalid Datum object.
1117	Invalid CoordSys object.
1118	Invalid units specified.
1119	Invalid origin longitude specified.
1120	Invalid origin latitude specified.
1121	Invalid standard parallel #1 specified.
1122	Invalid standard parallel #2 specified.
1123	Invalid azimuth specified.
1124	Invalid scale factor specified.
1125	Invalid false easting specified.
1126	Invalid false northing specified.
1127	Invalid range specified.
1128	Invalid bounds specified.
1129	Invalid AffineInfo object.
1130	Datum has not been specified.
1131	Units have not been specified.
1132	Origin longitude has not been specified.

1133	Origin latitude has not been specified.
1134	Standard parallel #1 has not been specified.
1135	Standard parallel #2 has not been specified.
1136	Azimuth has not been specified.
1137	Scale factor has not been specified.
1138	False easting has not been specified.
1139	False northing has not been specified.
1140	Range has not been specified.
1141	Bounds have not been specified.
1142	Invalid CoordSys type specified.
1143	Invalid AffineTransform object.
1144	Degenerate AffineTransform specified.
1145	Custom dataset error.
1146	The map contains no layers. The Geoset was not saved.
1147	The map contains temporary or userdraw layers. Those layers were not included in the saved Geoset.
1148	Dataset type does not support dynamic column binding.
1149	Style object does not support bitmap symbols.
1150	Invalid symbol type specified.
1151	Attempt to restore a dataset by name for which there was no saved dataset.
1152	Custom dataset error. Method failed: IMMapXColumnInfoContainer::GetColumnInfoEnumerator.
1153	Custom dataset error. Method failed: IMEnumMapXColumnInfo::Reset.
1154	Custom dataset error. Required interface not supported: IMMapXColumnInfoContainer.
1155	Custom dataset error. Method failed: IMMapXColumnInfo::GetColumnName.
1156	Custom dataset error. Method failed: IMMapXColumnInfo::GetColumnNumber.
1157	Custom dataset error. Method failed: IMMapXColumnInfo::GetDataType.
1158	Custom dataset error. A column contains an unsupported type.

1159	Custom dataset error. Method failed: IMMapXDataset::GetSample.
1160	Custom dataset error. Method failed: IMMapXStaticDataset::BeginFetch.
1161	Custom dataset error. Method failed: IMMapXStaticDataset::FetchData.
1162	Custom dataset error. Method failed: IMMapXStaticDataset::EndFetch.
1163	Custom dataset error. Method failed: IMMapXDynamicDataset::FetchData.
1164	Custom dataset error. Method failed: IMMapXDataset::Init.
1165	Custom dataset error. Could not find a CLSID for the type of custom dataset specified. Check that the custom dataset is properly registered.
1166	Custom dataset error. Could not create the custom dataset object. Check that the custom dataset is properly registered.
1167	Operation not allowed due to license restrictions.
1168	Unable to open one of the Drilldown component tables.
1169	Unable to reset Drilldown layer to specified level.
1170	The Feature is too complex. There are too many parts or points in the Feature object.
1171	A Line or Region Feature must have at least 1 Part.
1172	The stand-alone Feature object is not attached to the map. Use Feature.Attach() before other Feature methods or properties.
1173	Unable to attach Feature to the map. Make sure Объект Map is valid.
1174	Error creating buffer.
1175	Operation not allowed on an empty Features collection.
1176	Invalid parameter combination.
1177	Invalid Caption parameter specified.
1178	Invalid Intersection Point flag specified.
1179	Invalid Intersection Test flag specified.
1180	Error Combining Features
1181	Invalid Arc Distance specified.

1182	Invalid Angle specified.
1183	Missing or Invalid RefineBoundary specified in Search method.
1184	XY and PointRef datasets are not allowed with dynamic binding.
1185	Custom dataset error. Required interface not supported.
1186	Custom dataset error. GetSample must return an array.
1187	Custom dataset error. Unsupported type returned.
1188	Error saving geoset.
1189	Unable to find file <file> needed by the coordsys dialog.
1190	Layer.UpdateFeature(Target, Source) - 'Target' must be a feature from 'Layer'.
1191	Range and individual value themes are not available on SpatialWare layers.
1192	The find method is not available on SpatialWare layers.
1193	Operation not allowed on a map with no layers.
1194	Empty Text Annotations not allowed.
1195	Failed adding key <key> from drilldown level <level>; possibly others failed. Key Field not indexed?
1196	Failed removing key <key> from drilldown level <level>; possibly others failed.
1197	Invalid minimum zoom value specified. Zoom values must be greater than zero.
1198	Invalid maximum zoom value specified. Zoom values must be greater than zero.
1199	Setting Value not allowed unless ComputeTheme is false.
1200	Invalid style unit value specified.
1201	Cannot add a theme of type miThemeNone.
1202	The ShowCount property is only supported for ranged and individual value themes.
1203	The maps display coordinate system cannot be modified while the map contains visible raster layers.
1204	This line style does not support interleaved styles.
1205	No feature was found using the name you specified.
1206	The AllOthersText property is only supported for ranged and individual value themes.

1207	Invalid Match Threshold specified. Percentage must be between 1 and 100.
1208	The ShowCount property cannot be set to true, when ComputeTheme is false.
1209	The text font size property cannot be changed, when overriding a layer style.
1210	Invalid cache option.
1211	Cannot load specified cursor.
1212	Error occurred while searching within a layer.
1213	Error occurred while updating or drawing the selection.
1214	Invalid feature ID.
1215	Unable to retrieve or set the theme style information.
1216	Error while accessing thematic object.
1217	Error occurred while creating the theme.
1218	Error occurred while accessing the thematic modifier.
1219	Error occurred while recalculating the theme.
1220	The theme ranges are invalid. Please verify that the ranges do not overlap. You may want to set AutoRecompute to FALSE when setting custom theme ranges.
1221	This property or method is only supported for themes with custom ranges.
1222	Invalid theme category index specified.
1223	This property or method is not supported for the all others category.
1224	Invalid Layer MBRSearch option specified.
1225	Invalid predominant object type.
1226	Operation not allowed on this datasets collection.
1227	Dataset created without SourceRow information. See Datasets.BuildSourceRows.
1228	This property or method is only supported when ComputeTheme is false.
1229	Error occurred while setting the number of categories for the theme.
1230	A file named <filename> already exists.

1231	Unable to open file <filename>. Verify that the file exists and that you have sufficient access rights to open it.
1232	Unable to open file <filename>. Verify that the file exists in either the geodictionary directory or on the list of search paths. Also make sure that you have sufficient access rights to open it.
1233	Error writing file during export.
1234	Invalid BeginAccessType or Invalid EndAccessType.
1235	Theme value must be positive.
1236	This property is not supported for stacked bar charts.
1237	This property is only supported for stacked bar charts.
1238	The GraduateSizeBy property is only supported for graduated stacked bar charts and for bar charts which are not stacked or independently scaled.
1239	Invalid LayerInfo parameter specified.
1240	Error adding LayerInfo parameter.
1241	Expected LayerInfo parameter <parameter name> is either missing or invalid.
1242	Couldn't open <table name>. Table not found in the geodictionary.
1243	This property is not supported for the current symbol type. The symbol type must match the type of symbol for which you are changing properties.
1244	Theme deserialization error occurred.
1245	Theme serialization error occurred.
1246	Theme recreation error occurred.
1247	Cannot modify layer. Table may be readonly.
1248	Invalid Expression Parameter specified.
1249	Error occurred while creating the legend.
1250	Legend deserialization error occurred.
1251	Legend serialization error occurred.
1252	Legend recreation error occurred.
1253	Table is not mappable. Make sure table has an object column. If table is a server table, make sure that there is an entry in the mapinfo mapcatalog for it.

1254	Invalid closematch max value specified. Closematch values must be greater than or equal to zero.
1255	Invalid rounding value specified.
1256	This property can only be set when the RoundRanges property is set to true.
1257	This property can only be set when the SpreadBy property is set to miSpreadByColor.
1258	This property can only be set when the InflectRanges property is set to true.
1259	An invalid inflection range index was specified.
1260	Not enough ranges to use inflection. There must be at least three ranges to use inflection.
1261	An invalid apply attribute was specified.
1262	Cannot apply size attributes on a layer which predominantly contains bounded objects or text objects.
1263	Cannot spread by size on a layer which predominantly contains bounded objects or text objects.
1264	Cannot spread by color when applying size attributes.
1265	Cannot spread by size when applying color attributes.
1266	Cannot apply size attributes on an individual value theme.
1267	Invalid label max value specified. Label max values must be between 0 and 32767.
1268	Attempt to set the Insertion Layer to an object which is not a valid Объект Layer.
1269	Invalid RowValue object specified.
1270	Invalid RowValues object specified.
1271	Invalid UpdateFeature parameter specified.
1272	Error occurred while setting the current tool to a standard tool.
1273	The insertion layer is not set. The insertion layer must be set before the object creation tools can be used.
1274	Error occurred while setting the current tool to a custom tool.
1275	Error occurred while creating a standard tool.
1276	Error occurred while creating a custom tool.
1277	Cannot create a theme automatically on Seamless table.

1278	Invalid Key Specified.
1279	The layer that was specified is read-only. A read-only layer cannot be editable.
1280	The layer that was specified is not editable. The insertion layer can only be set to a layer that is editable.
1281	The layer specified does not contain any area objects. Centroids can only be shown for layers that contain bounded area objects.
1282	The layer specified does not contain any line objects. Line direction can only be shown for layers that contain line objects.
1283	The layer specified does not contain any point objects or area objects. Nodes can only be shown for layers that contain point objects or bounded area objects.
1284	Invalid layer specified. Cannot set the insertion layer to a drilldown, raster, seamless, or user draw layer.
1285	Invalid Field specified. Name not found, or index out of range.
1286	Error accessing the insertion tools for the map.
1287	The editable property for the insertion layer cannot be set to false.

Приложение В: Константы

Здесь рассмотрены следующие константы MapX:

AggregationFunctionConstants	AnnotationChangedTypeConstants
AnnotationTypeConstants	ApplyAttributeConstants
AreaUnitConstants	BindLayerTypeConstants
CircleTypeConstants	ColorConstants
ColorSpreadingMethodConstants	ConversionConstants
CoordSysTypeConstants	CursorConstants
DatasetTypeConstants	DistribMethodConstants
DotSizeConstants	ExportFormatConstants
FeatureTypeConstants	FieldTypeConstants
FillPatternConstants	Graduation Constants
IntersectionPointConstants	IntersectionTestConstants
LineTypeConstants	LayerBeginAccessConstants
LayerEndAccessConstants	LayerInfoTypeConstants
LayerSrvLayerOptions	LayerTypeConstants
MapDrawConstants	MapUnitConstants
MouseWheelSupportConstants	PaperUnitConstants
PenStyleConstants	PolyToolFlagConstants
PositionConstants	ResolveDataBindConstants
SearchTypeConstants	SelectionTypeConstants
SpreadByConstants	StyleUnitConstants
SymbolTypeConstants	ThemeTypeConstants
ToolConstants	ToolFlagConstants
ToolTypeConstants	

AggregationFunctionConstants

`miAggregationSum = 0`

`miAggregationAverage = 1`

`miAggregationCount = 2`

`miAggregationIndividual = 4`

`miAggregationAuto = 5`

AnnotationChangedTypeConstants

`miAddAnnotation = 0`

`miDeleteAnnotation = 1`

`miSelectAnnotation = 2`

`miEditAnnotation = 3`

AnnotationTypeConstants

`miSymbolAnnotation = 1`

`miTextAnnotation = 6`

ApplyAttributeConstants

`miApplyAttributeAll = 0`

`miApplyAttributeColor = 1`

`miApplyAttributeSize = 2`

AreaUnitConstants

`miUnitSquareMile = 14`

`miUnitSquareKilometer = 15`

`miUnitSquareInch = 16`

`miUnitSquareFoot = 17`

`miUnitSquareYard = 18`

`miUnitSquareMillimeter = 19`

`miUnitSquareCentimeter = 20`
`miUnitSquareMeter = 21`
`miUnitSquareSurveyFoot = 22`
`miUnitSquareNauticalMile = 23`
`miUnitSquareTwip = 24`
`miUnitSquarePoint = 25`
`miUnitSquarePica = 26`
`miUnitSquareDegree = 27`
`miUnitAcre = 28`
`miUnitHectare = 29`
`miUnitSquareLink = 33`
`miUnitSquareChain = 34`
`miUnitSquareRod = 35`
`miUnitPerch = 36`
`miUnitRood = 37`

BindLayerTypeConstants

`miBindLayerTypeNormal = 0`
`miBindLayerTypeXY = 1`
`miBindLayerTypePointRef = 2`

CircleTypeConstants

Constant	Value	Описание
<code>miCircleTypeScreen</code>	0	Метод <code>CreateCircularRegion</code> создает полигон, появляющийся на экране в виде круга. Таким образом, различные точки такого круга могут быть на разном географическом расстоянии от центра.
<code>miCircleTypeMap</code>	1	Метод <code>CreateCircularRegion</code> создает полигон, чьи

точки все находятся на равном географическом расстоянии от центра. Таким образом, в зависимости от системы координат и используемой проекции, полигон может появиться в деформированном виде, отличном от круга.

ColorConstants

Внимание: Эти номера представляют значения OLE_COLOR (BGR), которые могут быть присвоены свойствам, таким как Style.LineColor. Когда указанные цвета используются в рабочем наборе, используйте цвета RGB.

```
miColorBlack = 0
miColorRed = 255
miColorGreen = 65280
miColorBlue = 16711680
miColorMagenta = 16711935
miColorCyan = 16776960
miColorWhite = 16777215
miColorLightGray = 12632256
miColorDarkGray = 4210752
miColorGray = 8421504
miColorPaleYellow = 13697023
miColorLightYellow = 8454143
miColorYellow = 65535
miColorLimeGreen = 12639424
miColorTeal = 8421440
miColorDarkGreen = 16384
miColorMaroon = 128
miColorPurple = 8388736
miColorOrange = 33023
miColorKhaki = 7051175
miColorOlive = 32896
miColorBrown = 4210816
miColorNavy = 8404992
miColorScrollBars = 0x80000000
miColorDesktop = 0x80000001
```

miColorActiveTitleBar = 0x80000002
miColorInactiveTitleBar = 0x80000003
miColorMenuBar = 0x80000004
miColorWindowBackground = 0x80000005
miColorWindowFrame = 0x80000006
miColorMenuText = 0x80000007
miColorWindowText = 0x80000008
miColorTitleBarText = 0x80000009
miColorActiveBorder = 0x8000000A
miColorInactiveBorder = 0x8000000B
miColorApplicationWorkspace = 0x8000000C
miColorHighlight = 0x8000000D
miColorHighlightText = 0x8000000E
miColorButtonFace = 0x8000000F
miColorButtonShadow = 0x80000010
miColorGrayText = 0x80000011
miColorButtonText = 0x80000012
miColorInactiveCaptionText = 0x80000013
miColor3DHighlight = 0x80000014
miColor3DDarkShadow = 0x80000015
miColor3DLight = 0x80000016
miColorInfoText = 0x80000017
miColorInfoBackground = 0x80000018

ColorSpreadingMethodConstants

miColorMethodRGB = 0
miColorMethodHSV = 1

ConversionConstants

miMapToScreen = 0
miScreenToMap = 1

CoordSysTypeConstants

Константа	Значение	Описание
-----------	----------	----------

miAlbersEqualAreaConic	9	
miAzimuthalEquidistant	5	Polar aspect only
miCylindricalEqualArea	2	
miEckertIV	14	
miEckertVI	15	
miEquidistantConic	6	Also known as Simple Conic
miGall	17	
miHotineObliqueMercator	7	
miLambertAzimuthalEqualArea	4	Polar aspect only
miLambertConformalConic	3	
miLambertConformalConicBelgiu	19	Modified for Belgium 1972
m		
miLongLat	1	Longitude/Latitude
miMercator	10	
miMillerCylindrical	11	
miMollweide	13	
miNewZealandMapGrid	18	
miNonEarth	0	Unprojected; cartesian coords.
miRobinson	12	
miSinusoidal	16	
miStereographic	20	
miSwissObliqueMercator	25	
miTransverseMercator	8	Also known as Gauss-Kruger
miTransverseMercatorDenmarkS	21	Modified for Danish System 34
34J		Jylland-Fyn
miTransverseMercatorDenmarkS	22	Modified for Sjaelland
34S		
miTransverseMercatorDenmarkS	23	Modified for Danish System 45
45B		Bornholm
miTransverseMercatorFinland	24	Modified for Finnish KKJ

CursorConstants

miDefaultCursor = 0

miArrowCursor = 1
miCrossCursor = 2
miIBeamCursor = 3
miIconCursor = 4
miSizeCursor = 5
miSizeNESWCursor = 6
miSizeNSCursor = 7
miSizeNWSECursor = 8
miSizeEWCursor = 9
miUpArrowCursor = 10
miHourglassCursor = 11
miNoDropCursor = 12
miArrowHourglassCursor = 13
miArrowQuestionCursor = 14
miSizeAllCursor = 15
miArrowToolCursor = 16
miPanCursor = 17
miCenterCursor = 18
miZoomInCursor = 19
miZoomOutCursor = 20
miSymbolCursor = 21
miTextCursor = 22
miSelectCursor = 23
miRadiusSelectCursor = 24
miRectSelectCursor = 25
miRegionSelectCursor = 26
miInfoCursor = 27
miSelectPlusCursor = 28
miSelectRadiusPlusCursor = 29
miSelectRectPlusCursor = 30
miSelectRegionPlusCursor = 31
miSelectMinusCursor = 32
miSelectRadiusMinusCursor = 33
miSelectRectMinusCursor = 34
miSelectRegionMinusCursor = 35
miLabelCursor = 36

miDrillDownExpandCursor = 37
miDrillDownContractCursor = 38
miCustomCursor = 39

DatasetTypeConstants

miDataSetDAO = 1
miDataSetODBC = 2
miDataSetUnbound = 3
miDataSetGlobalHandle = 4
miDataSetOLEData = 5
miDataSetLayer = 6
miDataSetNotesView = 7
miDataSetNotesQuery = 8
miDataSetSafeArray = 9
miDataSetOEO = 10
miDataSetDelphi = 1010
miDataSetDelphi4 = 1011
miDataSetADO = 12
miDataSetRDO = 13

DistribMethodConstants

miCustomRanges = 0
miEqualCountPerRange = 1
miEqualRangeSize = 2
miNaturalBreak = 3
miStandardDeviation = 4

DotSizeConstants

miDotSizeSmall = 0
miDotSizeLarge = 1

ExportFormatConstants

miFormatWMF = 0
miFormatBMP = 1
miFormatGIF = 2
miFormatJPEG = 3
miFormatTIF = 4
miFormatPNG = 5
miFormatPSD = 6

FeatureTypeConstants

miFeatureTypeRegion = 0
miFeatureTypeLine = 1
miFeatureTypeSymbol = 2
miFeatureTypeMixed = 3
miFeatureTypeUnknown = 4
miFeatureTypeText = 5
miFeatureTypeNull = 6

FieldTypeConstants

miTypeString = 0
miTypeNumeric = 1

FillPatternConstants

miPatternNoFill = 0
miPatternHollow = 1
miPatternSolid = 2
miPatternHorizontal = 3
miPatternVertical = 4
miPatternFDiag = 5
miPatternFilBDiag = 6
miPatternCross = 7
miPatternDiagCross = 8

Graduation Constants

miGraduateBySquareRoot = 0

miGraduateByConstant = 1

miGraduateByLogarithm = 2

IntersectionPointConstants

miIntersectCrossings = 9

miIntersectCommon = 10

miIntersectAll = 11

IntersectionTestConstants

miIntersectCentroidWithinFeature = 0

miIntersectFeature = 1

miIntersectEntirelyWithinFeature = 2

LineTypeConstants

miLineTypeNone = 0

miLineTypeSimple = 1

miLineTypeArrow = 2

LayerBeginAccessConstants

miAccessRead = 0,

miAccessReadWrite = 1

LayerEndAccessConstants

miAccessEnd = 0,

LayerInfoTypeConstants

miLayerInfoTypeTab = 0

miLayerInfoTypeUserDraw = 1

miLayerInfoTypeRaster = 2

miLayerInfoTypeShape = 3
miLayerInfoTypeServer = 4
miLayerInfoTypeGeodictUserName = 5

LayerSrvLayerOptions

miLayerCacheOn = 0
miLayerCacheOff = 1
miLayerMBRSearchOn = 0
miLayerMBRSearchOff = 2

LayerTypeConstants

miLayerTypeNormal = 0
miLayerTypeRaster = 2
miLayerTypeSeamless = 4
miLayerTypeUnknown = 5
miLayerTypeUserDraw = 6
miLayerTypeDrilldown = 7

MapDrawConstants

miDrawBegin = 1
miDrawEnd = 2

MapUnitConstants

miUnitMile = 0
miUnitKilometer = 1
miUnitInch = 2
miUnitFoot = 3
miUnitYard = 4
miUnitMillimeter = 5
miUnitCentimeter = 6
miUnitMeter = 7
miUnitSurveyFoot = 8
miUnitNauticalMile = 9

miUnitTwip = 10
miUnitPoint = 11
miUnitPica = 12
miUnitDegree = 13
miUnitLink = 30
miUnitChain = 31
miUnitRod = 32

MouseWheelSupportConstants

miNoMousewheelSupport = 1
miMousewheelNoAutoScroll = 2
miFullMousewheelSupport = 3

PaperUnitConstants

miPaperUnitMile = 0
miPaperUnitKilometer = 1
miPaperUnitInch = 2
miPaperUnitFoot = 3
miPaperUnitYard = 4
miPaperUnitMillimeter = 5
miPaperUnitCentimeter = 6
miPaperUnitMeter = 7
miPaperUnitSurveyFoot = 8
miPaperUnitNauticalMile = 9
miPaperUnitTwip = 10
miPaperUnitPoint = 11
miPaperUnitPica = 12
miPaperUnitDegree = 13
miPaperUnitLink = 30
miPaperUnitChain = 31
miPaperUnitRod = 32

PenStyleConstants

miPenNone = 0

`miPenSolid = 1`

Внимание: Другие стили линий возможны, хотя константы не определены для таких стилей; смотрите свойство `Style.LineStyle`.

PolyToolFlagConstants

`miPolyToolBegin = 0`

`miPolyToolEnd = 1`

`miPolyToolEndEscaped = 2`

`miPolyToolInProgress = 3`

PositionConstants

`miPositionCC = 0`

`miPositionTL = 1`

`miPositionTC = 2`

`miPositionTR = 3`

`miPositionCL = 4`

`miPositionCR = 5`

`miPositionBL = 6`

`miPositionBC = 7`

`miPositionBR = 8`

ResolveDataBindConstants

`miChooseField = 0`

`miChooseLayer = 1`

`miChooseGeoSet = 2`

SearchTypeConstants

`miSearchTypeCentroidWithin = 0`

`miSearchTypePartiallyWithin = 1`

`miSearchTypeEntirelyWithin = 2`

SelectionTypeConstants

miSelectionNew = 0
miSelectionAppend = 1
miSelectionRemove = 2

SpreadByConstants

miSpreadByColor = 1
miSpreadByNone = 0
miSpreadBySize = 2

StyleUnitConstants

miStyleUnitPixel = 0
miStyleUnitTenthsOfPoint = 1

SymbolTypeConstants

miSymbolTypeBitmap = 1
miSymbolTypeTrueTypeFont = 0
miSymbolTypeVector = 2

ThemeTypeConstants

miThemeRanged = 0
miThemeBarChart = 1
miThemePieChart = 2
miThemeGradSymbol = 3
miThemeDotDensity = 4
miThemeIndividualValue = 5
miThemeAuto = 6
miThemeNone = 9

ToolConstants

miArrowTool = 1000

```
miPanTool = 1001
miCenterTool = 1002
miZoomInTool = 1003
miZoomOutTool = 1004
miSymbolTool = 1005
miTextTool = 1006
miSelectTool = 1007
miRadiusSelectTool = 1008
miRectSelectTool = 1009
miPolygonSelectTool = 1010
miLabelTool = 1011
miAddLineTool = 1012,
miAddPolylineTool = 1013,
miAddRegionTool = 1014,
miAddPointTool = 1015
```

ToolFlagConstants

```
miToolBegin = 0,
miToolEnd = 1,
miToolEndEscaped = 2,
miToolInProgress = 3
```

ToolTypeConstants

```
miToolTypePoint = 0
miToolTypeLine = 1
miToolTypeCircle = 2
miToolTypeMarquee = 3
miToolTypePoly = 4
miToolTypePolygon = 5
```

Приложение С: Создание выражений

Выражения в MapX полезны в нескольких смыслах. Метод Layer.Search может использоваться для выбора подмножества объектов на слое, основываясь на условии из выражения. В этом случае, выражение должно состоять из логических условий "year(Received) = 1990 and month(Received) > 4".

МетодDataset.AddField использует выражения для создания новых вычисляемых колонок. Эти выражения могут содержать как строковые так и числовые значения, например, "Female_Pop + Male_Pop" or "Proper\$(City) + "" , "" + UCase\$(State)"

Выражения формируются с использованием имен колонок (или полей) и констант (то есть специфических значений данных, а также функций и операторов, действующих на колонки и константы.

Строковые константы в выражениях должны заключаться в двойные кавычки, чтобы MapX мог воспринимать их как строковые, иначе будет их считать именем колонки.

Даты состоят из месяцев, дней и годов. Год задается двузначным или четырехзначным числом. Строка с датой должна быть заключена в кавычки. Компоненты даты могут отделяться дефисом или наклонной чертой.

Операторы

Математические операторы

A + B	+	addition
A - B	-	subtraction
A (negative)	-	negation
A * B	*	multiplication
A / B	/	division
A\B	\	integer division
A ^ B	^	exponentiation

Обратите внимание, что вы можете:

- Добавлять числа к дате с целью получения другой даты.
- Вычитать число из даты для получения другой даты.
- Вычитать дату из даты для получения числа.

Когда Вы добавляете числа к датам или вычитаете числа из дат, MapX рассматривает числа как определенное количество дней. Если Вы добавляете или вычитаете неделю, надо использовать число 7. Когда Вы хотите добавить или вычесть месяц, надо использовать 30 или 31. Когда MapX вычитает дату из даты, результатом будет число дней.

Строковый оператор

- & “concatenation” — connects strings and string expressions.
- + “concatenation” — connects strings and string expressions.

Внимание: Помните что все строковые величины должны быть в кавычках.

Операторы сравнения

=	“equals”
<>	“not equals”
>	“greater than”
<	“less than”
>=	“greater than or equal to”
<=	“less than or equal to”

Численное сравнение

Числовые сравнения базируются на числовых значениях в выражениях и числовых константах.

Пример: Все строки, где средний возраст равен 42.

- MED_AGE=42

Внимание: Это выражение выбирает только те записи, где средний возраст точно равен 42. Если данные с возрастом содержат десятичные знаки, то точного значения 42 может и не встретиться. В этом случае Вы можете использовать функцию Round для поиска записей с приблизительным возрастом 42.

Сравнение строк

Сравнение строк основывается на точном сравнении символов в строке. В этом случае “>” означает “по алфавитному порядку больше чем” и “<” означает что меньше по алфавитному проядку.”

Когда печатается строковая константа в выражении, надо заключить ее в кавычки, так чтобы MarX знал что это строка а не имя колонки.

Пример: Все строки где vendor это Acme.

- `VENDOR="Acme"`

Внимание: Оператор равенства весьма чувствительный.

Сравнение данных

Не забывайте заключать константы дат в кавычки и использовать в порядке Месяц, День, Год.

Пример: Все received после 9 октября 1991.

- `RECEIVED>"10-9-91"`

Внимание: Это выражение не выберет записи received в день 9 октября 1991. Для этого надо использовать оператор ">="

- `RECEIVED>="10-9-91"`

Пример: Все записи received до августа, независимо от года.

- `Month(RECEIVED)<8`

Логическое сравнение

Пример: All that have shipped.

- `Shipped`

Внимание: Колонка “Shipped” это логическая колонка. Она содержит “T” для true, или yes, и “F” для false, или no.

Пример: Выбрать все что не shipped.

- `Str$(Shipped)="F"`
- `Not Shipped`

Логические операторы

and	это “true” если (и только если) оба их аргумента (выражение объединяет их вместе) это true. Запись должна
-----	--

	удовлетворять обоим этим условиям, чтобы быть выбранной.
or	это “true” если или одно или оба их аргумента (выражение объединяет их вместе) это true. Запись должна удовлетворять только одному из этих условий, чтобы быть выбранной. Если выполняются оба условия одновременно, выборка тоже произойдет.
not	это “true” если аргумент (к которому применяется выражение) является false. Запись выбирается если не встречается состояние удовлетворяющее логическому условию.

“And”, “or” и “not” это логические операторы. Их надо использовать для составления выражений с использованием методов вроде Layer.Search.

Например, предположим что Вам надо выбрать все объекты, имеющие \$250,000 и более и находящиеся в графстве Columbia. Так Вы сможете скомбинировать два простых выражения используя оператор “and”:

COUNTY = “Columbia” and VALUE >= 250000

Теперь если Вам нужны собственники с \$250,000 или более вне графства Columbia? Вы можете использовать “not” перед первым выражением:

- not(COUNTY=“Columbia”) and VALUE>=250000

Можно использовать or для объединения независимых условий:

- COUNTY=“Columbia” or COUNTY=“Greene”

Строковые предложения

Пример: Все покупатели от N до Q.

- LAST_NAME >= “N” and LAST_NAME < “R”

Внимание: Первая часть выражения проверяет имена на равенство или превышение “n”. Вторая часть выражения проверяет имена на предмет того меньше ли они чем “r”. Любое имя начинающееся с “n” до “q” удовлетворяет этим условиям.

Пример: Все покупатели, чья фамилия начинается на C.

- LAST_NAME>=“C” and LAST_NAME<“D”

Предложения с датами

Пример: Записи для всего, полученного(received) в августе 1990

- `Month(RECEIVED)=8 and Year(RECEIVED)=1990`

Внимание: В этом выражении мы указываем год явно, используя функцию “year(<somecolumn>)” для извлечения из даты.

Пример: Выбрать все записи (received) в июле или сентябре 1989

- `month(RECEIVED)=(7, 9) and year(RECEIVED)=89`

Ключевые слова

MapX поддерживает использование ключевых слов “like” и “between”.

Рассмотрим пример, иллюстрирующий применение “between”:

- `PRICE between 50000 and 100000`

Можно также использовать between с символьными строками.

Ключевое слово “like” используется для сравнения строковых величин.

- Символ подчеркивания приводит к сравнению одного символа.
- Символ процента сравнивает ноль или более символов.

Пример: Определить все символьные строки начинающиеся с “South”

- `STATE like “South% ”`

Оператор Precedence

Когда MapX оценивает выражения он нуждается в том чтобы знать какие компоненты выражения оцениваются первыми. Это называется precedence (предшествование). По существующим условиям, некоторые операторы обладают разным уровнем знания предшествующих событий. Операторы с наивысшим уровнем оцениваются первыми. Следующая таблица операторов MapX показывает порядок оценивания операторов. Операторы одного уровня оцениваются слева направо, кроме экспоненты, которая оценивается справа налево.

Высший приоритет:	parenthesis
	exponentiation
	negation
	multiplication, division

	addition, subtraction
	geographic operators
	comparison operators
	Not
	And
Наименьший приоритет:	Or

Например, выражение $3+4*2$ даст результат 11. Это потому что операция умножения имеет больший приоритет чем сложение в списке выше. Скобки должны использоваться если надо повысить приоритет операции MapX : $(3+4)*2$.

Рассмотрим выражение, которое должно выбрать все записи в июле или сентябре 1989.

- `year(RECEIVED)=89 and month(RECEIVED)=7 or month(RECEIVED)=9`

Поскольку “and” имеет более высокий приоритет чем “or”, MapX будет понимать это выражение, как если бы были такие скобки “`year(RECEIVED)=89 and month(RECEIVED)=7`”. В этом случае, выберутся любы записи июля 89 или сентября любого года. Это может быть не то что Вам надо. Таким образом, при добавлении скобок ко второму выражению, можете получить желаемое выражение:

- `year(RECEIVED)=89 and (month(RECEIVED)=7 or month(RECEIVED)=9)`

Внимание: Если Вы не уверены в том, как MapX оценивает выражение из нескольких операторов, используйте скобки для группирования элементов, как Вам надо.

Функции

Функции берут данные и производят с ними некоторые преобразования, чтобы получить новые значения. Функции имеют следующий вид:

`SomeFunction(parameters)`

Большинство функций MapX берут один или два параметра. Параметр может быть колонкой или другим выражением.

MapX использует ключевые слова “obj” или “object” с географическими функциями: Area, CentroidX, CentroidY, ObjectLen и Perimeter. Это ключевое слово сообщает MapX что он получает значения базирующиеся на графическом объекте а не из таблицы.

Abs

Синтаксис: Abs(num_expr)

Действие: Возвращает абсолютное значение числового выражения. Когда выражение имеет значение большее чем ноль, Abs возвращает тоже значение. Когда выражение имеет значение меньшее чем ноль, Abs возвращает значение умножив его на минус единицу.

Acos

Синтаксис: Acos(num_expr)

Действие: Возвращает значение arc-cosine из числового значения num_expr. Другими словами возвращает значение угла, чей косинус равен num_expr.

Возвращаемый результат это угол в радианах. Этот угол будет между 0 и Pi радиан (Pi это примерно 3.14159, и величина Pi/2 радиан соответствует 90 градусам).

Поскольку косинус принимает значения от единицы до минус единицы, выражение num_expr должно представлять величину в этих пределах.

Area

Синтаксис: Area(obj, units)

Действие: Возвращает площадь объекта. 0 возвращается если объект не имеет площади или нет объекта для записи. Дуги, текст, точки, линии и полилинии не имеют площади.

Как и во всех функциях Geoobject, первый параметр всегда указывает специальную колонку с именем obj. Второй параметр это строка, определяющая в каких единицах будет оценена площадь. Допустимые единицы измерения:

Имя единиц измерения	Представляемая величина
“sq mi”	квадратные мили
“sq km”	квадратные километры
“sq in”	квадратные дюймы
“sq ft”	квадратные футы
“sq survey ft”	квадратные геодезические футы
“sq yd”	квадратные ярды
“sq mm”	квадратные миллиметры

“sq cm”	квадратные сантиметры
“sq m”	квадратные метры
“sq ch”	квадратные чейны
“sq li”	квадратные линксы
“sq rd”	квадратные родсы
“perch”	перчи (5.03 метра)
“rood”	родсы (четверть акра)
“acre”	акры
“hectare”	гектары

Пример: Выбрать все записи для объектов с площадью больше чем 59 квадратных миль:

- `Area(obj, "sq mi") > 59`

Выбрать все записи для объектов с населением меньше чем 250 человек на квадратную милю:

- `Population/Area(obj, "sq mi") < 250`

Asc

Синтаксис: `Asc(string_expr)`

Действие: Возвращает символьный код соответствующий первому символу в строке заданной переменной `string_expr`.

Если `string_expr` это нулевая строка, функция `Asc()` возвращает ноль.

Для систем поддерживающих двойную байтовую кодировку символов (типа Japanese Windows): если первый символ строковой величины `string_expr` это однобайтный символ, `Asc()` возвращает число в диапазоне 0 - 255; если первый символ величины `string_expr` это двубайтный символ, `Asc()` возвращает значение в диапазоне 256 - 65,535.

Для систем, не поддерживающих двухбайтовую кодировку символов, `Asc()` возвращает число в диапазоне 0 - 255.

Asin

Синтаксис: `Asin(num_expr)`

Действие: Возвращает арксинус значения `num_expr value`. Другими словами, `Asin()` возвращает угол, чей синус равен `num_expr`.

Возвращаемый результат это угол в радианах. Этот угол будет между 0 и π радиан (π это примерно 3.14159, и величина $\pi/2$ радиан соответствует 90 градусам).

Поскольку косинус принимает значения от единицы до минус единицы, выражение `num_expr` должно представлять величину в этих пределах.

Atan

Синтаксис: `Atan(num_expr)`

Действие: Возвращает арктангенс величины `num_expr`. Другими словами, `Atan()` возвращает угол, тангенс которого равен `num_expr`. Величина `num_expr` может иметь любое значение.

Возвращаемый результат это угол в радианах в диапазоне от $-\pi/2$ радиан до $\pi/2$.

Centroid

Синтаксис: `Centroid(obj_expr)`

Действие: Возвращает точечный объект, расположенный в месте центраида объекта `obj_expr` object.

Если параметр `obj_expr` является точечным объектом, то функция `Centroid()` возвратит положение этой очки.

Если параметр `obj_expr` это линейтый объект, функция `Centroid()` вернет точку посреди концами этой линии.

Если параметр `obj_expr` представляет объект полилиния, функция `Centroid()` возвратит точку, расположенную в середине среднего сегмента полилинии.

Если параметр `obj_expr` представляет другие типы объектов, функция `Centroid()` возвращает точку, расположенную в тойже точке, что и центроид исходного объекта. Для прямоугольника, дуги, текста, и эллипса, положение центроида это среднее место между верхней и нижней границей, между левой и правой границей. Для объектов полигонов, таким образом, положение центроида необязательно окажется внутри самого объекта.

CentroidX

Синтаксис: `CentroidX(object)`

Действие: Возвращает координату x центроида объекта, это долгота. Центроид это всегда центр минимального описывающего объект прямоугольника (MBR).

Значение возвращается в десятичных градусах, если координаты это широта и долгота.

Значение будет в тех же единицах, что и таблица, если координаты плановые.

Пример: Выбрать все объекты к западу от New York City:

- `CentroidX(obj)<-73.997890`

CentroidY

Синтаксис: `CentroidY(object)`

Действие: Возвращает координату y центроида объекта, это широта. Центроид это всегда центр минимального описывающего объект прямоугольника (MBR).

Значение возвращается в десятичных градусах, если координаты это широта и долгота.

Значение будет в тех же единицах, что и таблица, если координаты плановые.

Пример: Выбрать все объекты к северу от New York City:

- `CentroidY(obj)>40.750450`

Chr\$

Синтаксис: `Chr$(num_expr)`

Действие: `Chr$` интерпретирует `num_expr` как строковую величину. Функция возвращает символ, соответствующий этой величине. Таким образом, `num_expr` должно быть целым между 0 и 255.

Пример: Вам надо сделать подпись из двух строк. Вы можете использовать `Chr$` чтобы вставить возврат каретки в выражение подписи. Значение ANSI для возврата каретки это 13. Предположим, что в первой строке должно быть название страны, а во второй численность населения. Выражение будет таким:

- `Proper$(County)+Chr$(13)+Population`

Внимание: Некоторые языки используют для конца строки символ `Chr$(10)`. `MapX` позволяет Вам выбрать `Chr$(13)` или `Chr$(10)`.

Cos

Синтаксис: `Cos(num_expr)`

Действие: Функция Cos возвращает косинус от значения числового выражения, где выражение представляет угол в радианах.

CurDate

Синтаксис: CurDate()

Действие: Возвращает значение текущей даты.

Пример: Выбрать все записи за период 30 назад и ранее:

- Received<=CurDate()-30

Day

Синтаксис: Day(datefield)

Действие: Возвращает день из значения даты. День это целое число от (1) до (31).

Пример: Выбрать все записи в которых дата имеет первый день месяца:

- Day(date)=1

Выбрать все записи где день это пятница и день не тринадцатое число:

- Weekday(date)=6 and Day(date)<>13

Distance

Синтаксис: Distance(x1, y1, x2, y2, units)

- x1 и y1 это Восток-Запад (Долгота) в градусах
- x2 и y2 это Север-Юг (Широта) в градусах
- текущие единицы измерения расстояния

Действие: Вычисляет расстояние между двумя точками.

Параметр единиц измерения это строковая константа. Допустимые единицы измерения:

Имя единицы	Описание единиц
“mi”	мили
“km”	километры
“yd”	ярды
“mm”	миллиметры
“cm”	сантиметры
“m”	метры

“nmi”

морские мили (1 морская миля это 1852 метра)
--

Примеры: Подсчитать расстояние между некоторыми объектами и New York City, который расположен в месте -73.997890 долготы и 40.750450 широты:

- `Distance(-73.997890, 40.750450, CentroidX(obj), CentroidY(obj), "mi")`

Exp

Синтаксис: `Exp(num_expr)`

Действие: Возвращает показатель степени для основания натуральных логарифмов, чтобы выразить заданное число через экспоненту; e приблизительно 2.7182818.

Обратите внимание что выражения `MapX` поддерживают экспоненциальную зависимость символом (^).

Fix

Синтаксис: `Fix(num_expr)`

Действие: Удаляет часть числа и возвращает целое значение.

Функция `Fix()` похожа, но идентична функции `Int()`. Эти две функции отличаются способом обработки отрицательных значений. Для отрицательных значений `Fix()` возвращает ближайшее целое значение, большее или равное исходному:

```
Fix(-2.3)
```

возвращает -2. Но когда функция `Int()` обрабатывает отрицательное значение, она возвращает ближайшее целое, меньшее или равное исходному:

```
Int(-2.3)
```

возвращает -3.

InStr

Синтаксис: `InStr(position, string, substring)`

Действие: `InStr` проверяет есть ли в строковой величине более короткая, заданная строковая величина. `MapX` ищет строковую величину, начиная с указанной позиции в исходной строковой величине. Если эта позиция равна единице, `MapX` начинает искать с начала строки. Если это шесть, `MapX` начинает искать с шестого символа.

Если строковая величина содержит искомую, более короткую строковую величину, InStr возвращает позицию, с которой она начинается. Если такая подстрока не найдена, InStr возвращает ноль.

Пример: Вы начинаете геокодировать файл и хотите определить все записи с почтовыми отделениями для адресов. Это означает, что Вам нужны все записи, имеющие строковую величину “box” в адресной колонке. Они так же могут иметь аббревиатуру “P.O.”, но Вы в этом не уверены. Вы также не уверены какими буквами (заглавными или нет) набрано слово “box”. Выражение будет таким:

- `InStr(1,UCase$(Address),”BOX”)>0`

Это выражение указывает MapX искать в адресной колонке слово “BOX”. Все записи, содержащие это слово, будут выбраны.

Int

Синтаксис: `Int(num_expr)`

Действие: Возвращает ближайшее целое которое меньше или равно указанному значению (num_expr).

Примеры: Таблица показывает примеры того, как Int обрабатывает значения:

Number	Int(Number)
5.2	5
5.999	5
7.8	8
7.2	8

LCase\$

Синтаксис: `LCase$(string)`

Действие: Возвращает прописные буквы в строковой величине.

Примеры: Рассмотрим следующее выражение:

- `LCase$(City)`

Следующая таблица показывает как преобразуется строковая величина (из колонки таблицы):

Исходная строка	Результат
-----------------	-----------

NEW YORK	new york
New yorK	new york
new york	new york

Left\$

Синтаксис: Left\$(string, number)

Действие: Возвращает строку, состоящую из определенного числа символов в левой части исходной строки.

Примеры: Таблица показывает, как Left\$() действует с параметром 5:

Строка	Left\$(String, 5)
New York	New Y
Denver	Denve
Singapore	Singa

Теперь рассмотрим следующее выражение, где Phone это колонка, содержащая телефонные номера, первые три цифры это код города:

- Left\$(Phone,3)="404"

Выражение будет выбирать все строки, где встречен код "404".

Log

Синтаксис: Log(num_expr)

Действие: Возвращает значение натурального логарифма исходного числа num_expr.

Натуральный логарифм это число, показывает в какую степень должно быть возведено число e чтобы получить величину num_expr. e примерно равно 2.7182818.

Логарифм определен только для положительных значений; функция Log() будет генерировать ошибку, если num_expr имеет отрицательное значение.

Вы можете подсчитать логарифм по другому основанию (например, 10) используя натуральный логарифм.

LTrim\$

Синтаксис: LTrim\$(string)

Действие: LTrim\$ удаляет все непечатаемые символа вначале строки (пробелы, табуляторы и др.).

Maximum

Синтаксис: Maximum(num_expr, num_expr)

Действие: Возвращает максимальное из двух значений.

MBR

Синтаксис: MBR(obj_expr)

Действие: Возвращает объект прямоугольник, представляющий минимальный описывающий прямоугольник для объекта obj_expr.

MBR точечного объекта вернет объект с нулевой шириной и высотой.

Mid\$

Синтаксис: Mid\$(string, position, length)

string это строковая величина

position это целое, стартовая позиция

length это целое, показывающее сколько символов из строки надо извлечь.

Действие: Mid\$ возвращает строку заданной длины, начиная с указанной позиции, в заданной строке.

Примеры: Следующая таблица показывает как Mid\$ действует для позиции 5 и длины 4:

Строка	Mid\$(String, 5, 4)
New York City	York
Denver	er
Singapore	apor

Внимание: Выражение вернуло два символа из “Denver”. Это потому что “Denver” слишком короткий.

Minimum

Синтаксис: Minimum(num_expr, num_expr)

Действие: Возвращает минимальное значение из двух заданных.

Month

Синтаксис: Month(datefield)

Действие: Возвращает месяц из значения даты, от 1 до 12.

Примеры: Выбрать все записи полученные (received) до августа:

- Month(received)<8

Выбрать все записи полученные (received) в августе и сентябре 1990:

- Month(received)=(8,9) and Year(received)=1990

ObjectLen

Синтаксис: ObjectLen(obj, unit)

Действие: Возвращает длину линии и полилинии. Возвращает 0 для других объектов.

Параметр unit указывает единицы измерения длины. Возможны такие:

Единицы	Описание
“mi”	мили
“km”	километры
“in”	дюймы
“ft”	футы
“survey ft”	геодезические футы
“yd”	ярды
“mm”	миллиметры
“cm”	сантиметры
“li”	линки
“rd”	родсы
“ch”	чейны
“m”	метры
“nmi”	морские мили (1 морская миля равна 1852 метрам)

Примеры: Рассчитать длину объекта в километрах:

- ObjectLen(obj,”km”)

Выбрать все объекты длиной менее 10 миль:

- `ObjectLen(obj, "mi")>10`

Perimeter

Синтаксис: `Perimeter(obj, unit)`

Действие: Возвращает периметр объектов полигон, эллипс, прямоугольник и скругленный прямоугольник. Параметр `unit` указывает единицы измерения периметра. Возможны следующие:

Единицы	Описание
"mi"	мили
"km"	километры
"in"	дюймы
"ft"	футы
"survey ft"	геодезические футы
"yd"	ярды
"mm"	миллиметры
"cm"	сантиметры
"li"	линки
"rd"	родсы
"ch"	чейны
"m"	метры
"nmi"	морские мили (1 морская миля равна 1852 метрам)

Пример: Выбрать все объекты с периметром большим чем 35 километров:

- `Perimeter(obj, "km")>35`

Proper\$

Синтаксис: `Proper$(string)`

Действие: Возвращает строку в которой каждая первая буква каждого слова заглавная а остальные прописные.

Примеры: Следующая таблица показывает действие функции `Proper$` на колонку таблицы:

Строка	<code>Proper\$(String)</code>
--------	-------------------------------

NEW YORK	New York
new YorK	New York
new york	New York
New York	New York

Right\$

Синтаксис: Right\$(string_expr, num_expr)

Действие: Возвращает последние символы, числом num_expr из строки string_expr.

Round

Синтаксис: Round(number1, number2)

Действие: Возвращает округленное значение числа number1. Число number2 показывает как надо округлять.

Примеры: Рассмотрим следующее выражение:

- Round(number1,number2)

Таблица ниже дает примеры. Первая колонка таблицы показывает аргументы (number1, number2) а вторая колонка показывает результат действия функции.

Аргументы	Округленные значения
14347,10000	10000
14347, 100	14300
14347, 10	14350
12.18353, .1	12.20000
12.18353, .001	12.18400

Чтобы выбрать все записи для которых средний возраст равен 42, используйте выражение:

- Round(MED_AGE, 1)=42

В зависимости от исходных данных, выражение может дать хороший результат, если попадутся записи с точно таким числом (MED_AGE=42). Если таблица содержит данные с десятичными и сотыми долями то необходимо использовать округление, например, 41.7, 42.1 или 42.4. При округлении и получится число 42.

RTrim\$

Синтаксис: RTrim\$(string)

Действие: RTrim\$ удаляет все непечатаемые символы (пробелы, табуляция и др.) в конце строки.

Sgn

Синтаксис: Sgn(num_expr)

Действие: Возвращает значение -1 если num_expr меньше чем ноль, ноль если num_expr равно нулю и 1 если num_expr больше чем ноль.

Sin

Синтаксис: Sin(num_expr)

Действие: Функция Sin возвращает синус заданного выражения, представляющего собой угол в радианах.

Space\$

Синтаксис: Space(num_expr)

Действие: Возвращает строку длиной num_expr, состоящую из символов пробела.

Если num_expr меньше или равна нулю, функция Space\$() возвратит нулевую строку.

Str\$

Синтаксис: Str\$(expr)

Действие: Str\$ конвертирует числовое выражение в строковую величину. Если числовое выражение отрицательное, то первый символ будет знак минус (-). Первый символ для положительного числа будет такой же как и в исходном числе. Если выражение это объект, Str\$() возвращает описание типа объекта (например, "region").

Sqr

Синтаксис: Sqr(num_expr)

Действие: Возвращает корень квадратный численного выражения указанного величиной num_expr. Поскольку извлечение корня квадратного неопределено для отрицательных чисел, num_expr должна быть больше или равна нулю.

Извлечение квадратного корня эквивалентно возведению в степень 0.5. Таким образом, выражение Sqr(n) эквивалентно выражению $n^{0.5}$; функция Sqr() обеспечивает более быстрые вычисления.

Tan

Синтаксис: Tan(num_expr)

Действие: Возвращает тангенс числового выражения num_expr, представляющего угол в радианах.

UCase\$

Синтаксис: UCase\$(string)

Действие: Возвращает строку с заглавными буквами.

Примеры: Таблица показывает как действует функция UCase\$:

Строка	UCase\$(String)
NEW YORK	NEW YORK
New yorK	NEW YORK
new york	NEW YORK

Val

Синтаксис: Val(char_expr)

Действие: Функция Val извлекает числовое значение из строковой величины. Она игнорирует все табуляции, пробелы, возврат каретки и воспринимает строку с первого значимого символа. Она останавливает процесс как только встретит первый нецифровой символ, исключая знак минус (перенос) и десятичную точку. Если первый символ после начальных табуляций, пробелов и конца строки, не является цифровым, то Val вернет ноль.

Примеры: Следующая таблица показывает действие функции Val:

Строка	Val(String)
12 thousand	12

52 62 Brunswick	52
Eighteen	0
Box 239	0

Weekday

Синтаксис: Weekday(datefield)

Действие: Возвращает день недели из даты. Это число от 1 до 7. 1 это Воскресенье, а 7 это Суббота.

Пример: Выбрать все записи где есть Wednesday, Thursday или Friday:

- `Weekday(date)=(4,5,6)`

Year

Синтаксис: Year(datefield)

Действие: Возвращает год из даты.

Пример: Найти все объекты, полученные (received) в 1990 или 1991:

- `Year(date)=(1990,1991)`

Приложение D: Ключи Geoset

Рабочие наборы Geosets это файлы (с расширением .GST) содержащие информацию о слоях, которые могут быть загружены одновременно. Рабочий набор geoset загружается по требованию пользователя (как свойство), или используя метод AddGeosetLayers объекта Layers. Файл рабочего набора это текст в ASCII. Строки в файле состоят из ключей и значений. Ключи соотносятся со свойствами в MapX – свойства для главного Объекта карты, как для объекта Layers.

Ключи имеют иерархическую природу, и указаны в кавычках. Величина ключа тоже в кавычках – даже число в кавычках. Следующие примеры показывают ключи и их значения:

Пример рабочего набора Geoset

```
!GEOSET
!VERSION 100
begin_metadata
"\GEOSET" = ""
"\GEOSET\NAME" = "SAMPLE GEOSET"
"\GEOSET\PROJECTION" = "3,62,7,-96,23,20,60,0,0"
"\GEOSET\CENTER" = "-54851.35483414936,1844196.997419479"
"\GEOSET\MBR" = ""
"\GEOSET\MBR\LOWERLEFT" = "-3093309.705881681,-450646.671927353"
"\GEOSET\MBR\UPPERRIGHT" = "2983606.996213382,4139040.666766311"
"\GEOSET\ZOOMLEVEL" = "4019.82"
"\GEOSET\AUTOLAYER" = "FALSE"
"\GEOSET\MAPUNIT" = "0"
"\GEOSET\ROTATION" = "0"
"\TABLE" = ""
"\TABLE\1" = ""
"\TABLE\1\FILE" = "usa_caps.tab"
"\TABLE\1\DESCRIPTION" = "US Capitals"
"\TABLE\1\ISVISIBLE" = "TRUE"
"\TABLE\1\AUTOLABEL" = "FALSE"
"\TABLE\1\DRAWLABELSAFTER" = "FALSE"
"\TABLE\1\SHOWLINEDIRECTION" = "FALSE"
"\TABLE\1\SHOWNODES" = "FALSE"
"\TABLE\1\SHOWCENTROIDS" = "FALSE"
"\TABLE\1\EDITABLE" = "FALSE"
"\TABLE\1\SELECTABLE" = "TRUE"
"\TABLE\1\REGISTERINGEODICT" = "TRUE"
"\TABLE\1\DISPLAY" = ""
"\TABLE\1\DISPLAY\BRUSH" = ""
"\TABLE\1\DISPLAY\BRUSH\Pattern" = "2"
"\TABLE\1\DISPLAY\BRUSH\Forecolor" = "16777215"
"\TABLE\1\DISPLAY\BRUSH\Backcolor" = "16777215"
"\TABLE\1\DISPLAY\BRUSH\Transparent" = "FALSE"
```



```

"\TABLE\1\DISPLAY\PEN" = ""
"\TABLE\1\DISPLAY\PEN\LineWidth" = "1"
"\TABLE\1\DISPLAY\PEN\LineStyle" = "1"
"\TABLE\1\DISPLAY\PEN\Pattern" = "2"
"\TABLE\1\DISPLAY\PEN\Color" = "0"
"\TABLE\1\DISPLAY\LINEPEN" = ""
"\TABLE\1\DISPLAY\LINEPEN\LineWidth" = "1"
"\TABLE\1\DISPLAY\LINEPEN\LineStyle" = "1"
"\TABLE\1\DISPLAY\LINEPEN\Pattern" = "2"
"\TABLE\1\DISPLAY\LINEPEN\Color" = "0"
"\TABLE\1\DISPLAY\FONT" = ""
"\TABLE\1\DISPLAY\FONT\Style" = "3"
"\TABLE\1\DISPLAY\FONT\ExtStyle" = "1"
"\TABLE\1\DISPLAY\FONT\Description" = "Arial"
"\TABLE\1\DISPLAY\FONT\Size" = "36"
"\TABLE\1\DISPLAY\FONT\Forecolor" = "0"
"\TABLE\1\DISPLAY\FONT\Backcolor" = "16777215"
"\TABLE\1\DISPLAY\FONT\Opaque" = "FALSE"
"\TABLE\1\DISPLAY\SYMBOL" = ""
"\TABLE\1\DISPLAY\SYMBOL\Type" = "1"
"\TABLE\1\DISPLAY\SYMBOL\Code" = "35"
"\TABLE\1\DISPLAY\SYMBOL\Color" = "255"
"\TABLE\1\DISPLAY\SYMBOL\Pointsize" = "14"
"\TABLE\1\DISPLAY\SYMBOL\Font" = ""
"\TABLE\1\DISPLAY\SYMBOL\Font\Style" = "0"
"\TABLE\1\DISPLAY\SYMBOL\Font\ExtStyle" = "0"
"\TABLE\1\DISPLAY\SYMBOL\Font\Description" = "Map Symbols"
"\TABLE\1\DISPLAY\SYMBOL\Font\Size" = "14"
"\TABLE\1\DISPLAY\SYMBOL\Font\Forecolor" = "255"
"\TABLE\1\DISPLAY\SYMBOL\Font\Backcolor" = "16777215"
"\TABLE\1\DISPLAY\SYMBOL\Font\Opaque" = "FALSE"
"\TABLE\1\DISPLAY\SYMBOL\Font\Rotation" = "0"
"\TABLE\1\LABEL" = ""
"\TABLE\1\LABEL\FONT" = ""

```

```

"\TABLE\1\LABEL\FONT\Style" = "0"
"\TABLE\1\LABEL\FONT\ExtStyle" = "0"
"\TABLE\1\LABEL\FONT\Description" = "Arial"
"\TABLE\1\LABEL\FONT\Size" = "9"
"\TABLE\1\LABEL\FONT\Forecolor" = "0"
"\TABLE\1\LABEL\FONT\Backcolor" = "16777215"
"\TABLE\1\LABEL\FONT\Opaque" = "FALSE"
"\TABLE\1\LABEL\DUPLICATE" = "TRUE"
"\TABLE\1\LABEL\PARALLEL" = "TRUE"
"\TABLE\1\LABEL\OVERLAP" = "FALSE"
"\TABLE\1\LABEL\PARTIALSEGMENTS" = "FALSE"
"\TABLE\1\LABEL\LINETYPE" = "2"
"\TABLE\1\LABEL\OFFSET" = "2"
"\TABLE\1\LABEL\POSITION" = "5"
end_metadata

```

Первые одиннадцать строк показывают ключи, начинающиеся со слов 'GEOSET' – эти ключи устанавливают свойства для всей карты, или свойства для Объекта Map. Обратите внимание, что некоторые ключи, типа GEOSET\MBR являются многоуровневыми – здесь есть LOWERLEFT и UPPERRIGHT.

Следующие тринадцать строк показывают ключи устанавливающие свойства для одного слоя – usa_caps. Ключи начинаются с 'TABLE'. Следующее слово это любое ключевое слово ссылающееся на установку свойств для указанного слоя, в этом случае ключевое слово совпадает с именем таблицы. Затем следуют иерерхические ключи для указанного слоя.

Поддерживаемые ключи

Список ключей, поддерживаемых в файле geoset:

\GEOSET\

NAME	'Friendly' name of geoset
PROJECTION	Projection clause
CENTER	Number, Number – Long/Lat of the center of the map
\MBR\LOWERLEFT	Number,Number - Long/Lat of lower left corner
\MBR\UPPERRIGHT	Number,Number - Long/Lat of upper right corner

ZOOMLEVEL	Number - Zoom level of the map
AUTOLAYER	TRUE or FALSE - order layers automatically
MAPUNIT	Number - corresponds to Map.MapUnit. Whenever the geoset is loaded into the map control, if there are no layers already loaded, the MapUnit will be set to the geoset's MapUnit. However, if there are layers already loaded, MapX will load the geoset and use the current map unit.
ROTATION	Number - Rotation angle of the map. Corresponds to Map.Rotation
\TABLE<number>\	
FILE	String - location of .TAB file
DESCRIPTION	String - Table description
ISVISIBLE	TRUE or FALSE - whether layer is visible
AUTOLABEL	TRUE or FALSE - whether layer is autolabeled
DRAWLABELSAFTER	TRUE or FALSE - whether to draw labels after the layer is drawn
SHOWLINEDIRECTION	TRUE or FALSE – Corresponds to Layer.ShowLineDirection
SHOWNODES	TRUE or FALSE – Corresponds to Layer.ShowNodes
SHOWCENTROIDS	TRUE or FALSE – Corresponds to Layer.ShowCentroids
EDITABLE	TRUE or FALSE – Corresponds to Layer.Editable
SELECTABLE	TRUE or FALSE – Corresponds to Layer.Selectable
REGISTERINGEODICT	TRUE or FALSE – whether to register the geoset in the geodictionary
ZOOM\MIN	Number - minimum zoom value to display layer (Layer.ZoomMin)
ZOOM\MAX	Number - maximum zoom value to display layer (Layer.ZoomMax)

DISPLAY\BRUSH

PATTERN	Number - corresponds to Style.RegionPattern
FORECOLOR	Number - corresponds to Style.RegionColor
BACKCOLOR	Number - corresponds to Style.RegionBackColor
TRANSPARENT	TRUE or FALSE - corresponds to Style.RegionTransparent

DISPLAY\PEN

LINEWIDTH	Number - corresponds to Style.RegionBorderWidth
LINESTYLE	Number - corresponds to Style.RegionBorderStyle
COLOR	Number - corresponds to Style.RegionBorderColor

DISPLAY\LINEPEN

LINEWIDTH	Number - corresponds to Style.LineWidth
LINESTYLE	Number - corresponds to Style.LineStyle
COLOR	Number - corresponds to Style.LineColor

DISPLAY\FONT

STYLE	Number - Bit Mask (BOLD 0x01, ITALIC 0x02, UNDER 0x04, STRIKEOUT 0x08, OUTLINE 0x10, SHADOW 0x20, INVERSE 0x40, BLINK 0x80)
EXTSTYLE	Number - Bit Mask (HALO 0x01, ALLCAPS 0x02, DBLSPACE 0x04)
DESCRIPTION	String - font name
SIZE	Number - font size in points
FORECOLOR	Number - Corresponds to Style.TextFontColor
BACKCOLOR	Number - Corresponds to Style.TextFontBackColor
OPAQUE	TRUE or FALSE - Corresponds to Style.TextFontOpaque

DISPLAY\SYMBOL

TYPE	Number – Corresponds to Style.SymbolType
CODE	Number - Corresponds to Style.SymbolCharacter
COLOR	Number - Corresponds to Style.SymbolFontColor
POINTSIZE	Number - size of symbol in points

TrueType Font Symbol Keys

FONT	This is the font of the symbol. This key is present if the symbol type is miSymbolTypeTrueTypeFont. It has the same sub-keys as DISPLAY\FONT
FONT\ROTATION	The rotation angle of the font symbol. Corresponds to Style.SymbolFontRotation.

Bitmap Symbol Keys

BITMAP\NAME	String - Name of the bitmap symbol. Corresponds to Style.SymbolBitmapName.
BITMAP\OVERRIDECOLOR	TRUE or FALSE – Whether to apply color to the symbol. Corresponds to Style.SymbolBitmapOverrideColor
BITMAP\TRANSPARENT	TRUE or FALSE – Whether the bitmap is transparent. Corresponds to Style.SymbolBitmapTransparent.

LABEL

ZOOM\MIN	Number - minimum zoom value to label layer
ZOOM\MAX	Number - maximum zoom value to label layer
FONT	This is the label font. It has the same sub-keys as DISPLAY\FONT above
DUPLICATE	TRUE or FALSE - permit duplicate labels
PARALLEL	TRUE or FALSE - Corresponds to LabelProperties.Parallel

OVERLAP	TRUE or FALSE - Corresponds to LabelProperties.Overlap
PARTIALSEGMENTS	TRUE or FALSE – Corresponds to LabelProperties.PartialSegments
MAXLABELS	Number - Corresponds to LabelProperties.LabelMax
LINETYPE	Number - Corresponds to LabelProperties.LineType
OFFSET	Number - Corresponds to LabelProperties.Offset
POSITION	Number - Corresponds to LabelProperties.Position

Приложение E: Значения OLE_COLOR

Значение OLE_COLOR это BGR (Blue, Green, Red). Для определения значений BGR, укажите blue, green или red (каждое от 0 до 255) по следующей формуле:

BGR value = (blue * 65536) + (green * 256) + red

Внимание: Если Вы используете язык программирования MapBasic, помните, что в MapBasic используется такая формула для RGB:

RGB value = (red * 65536) + (green * 256) + blue

Если Вы работаете с цветами RGB (как в приложениях MapBasic), вам надо произвести конвертацию цветов в BGR перед использованием цветов в MapX.

Смотрите также

ColorConstants