

# **Charts 类库使用说明手册**

2018 年 3 月初版

上海地听信息科技有限公司

# 目录

<b>CHARTS 类库使用说明手册 .....</b>	<b>1</b>
<b>第一部分： CHART2D.....</b>	<b>1</b>
<b>1 CHART2D 结构简介.....</b>	<b>1</b>
1.1 类库组成.....	1
1.2 Chart2D 基本结构.....	1
1.2.1 Chart2D .....	1
1.2.2 绘图区 ChartStyle.....	3
1.2.3 图表组合 SubChart.....	3
1.2.4 图表类型简介 ChartType .....	4
1.3 Chart2D 基本元素.....	6
1.3.1 标题 Title .....	6
1.3.2 标签 Labels .....	6
1.3.3 坐标轴 Axes .....	7
1.3.5 图例 Legend.....	8
1.3.4 网格线 Grid.....	10
1.4 Chart2D 基本图形样式.....	11
1.4.1 文本 TextStyle .....	11
1.4.2 线条 LineStyle .....	11
1.4.3 符号 SymbolStyle.....	12
1.4.4 简单图形颜色 GraphColor .....	13
1.4.5 色彩映射 ColorMap .....	13
1.5 Chart2D 数据组织.....	15
1.5.1 数据系列 DataSeries .....	15
1.5.2 标准数据集 DataCollection 及其派生类 .....	16
<b>2 CHART2D 操作说明.....</b>	<b>19</b>
2.1 单幅单类型 Chart2D.....	19
2.2 单幅多类型 Chart2D.....	24
2.3 多幅多类型 Chart2D.....	25

# 第一部分： Chart2D

## 1 Chart2D 结构简介

### 1.1 类库组成

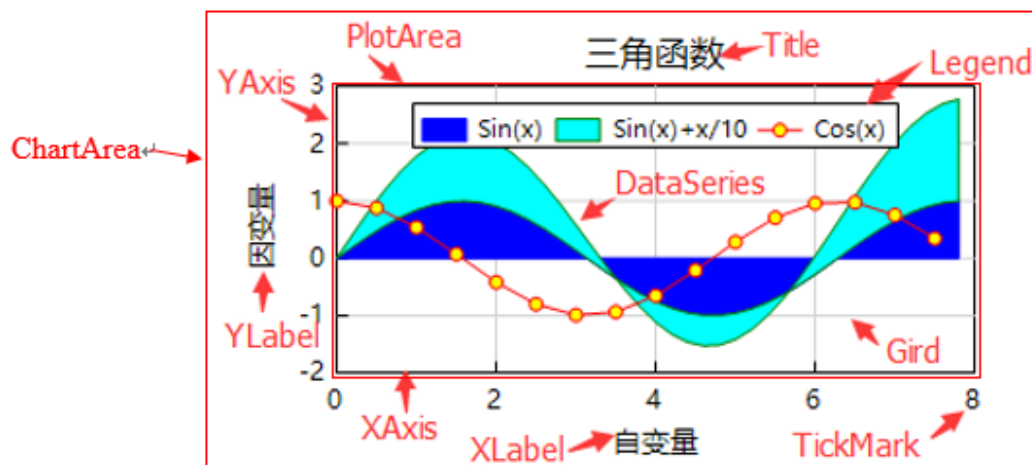
Readearth.Chart2D.dll 包括

- (1)Readearth.Chart2D: Chart2D、元素布局 (ChartStyle)、Chart2D 分类及其绘制方法、图表组合
- (2)Readearth.Chart2D.Data: 数据集 (DataCollection 及其子类) 和数据系列 (DataSeries)
- (3)Readearth.Chart2D.BasicStyle: 基本图形样式
- (4)Readearth.Chart2D.ChartElements: 基本元素
- (5)Readearth.Chart2D.Additional: 可分离复用功能 (如导出图片等)

### 1.2 Chart2D 基本结构

Readearth.Chart2D.Chart2D 类用于绘制 Chart2D，由大小 (ChartSize)、样式布局、数据集、标题、标签、坐标轴、网格线和图例这八个主要字段构成。

Chart2D 的基本结构组成大致如下图所示：



#### 1.2.1 Chart2D

Chart2D 构造函数：

名称	说明
Chart2D()	初始化 Chart2D 类的新实例，默认大小为长 600px、宽 400px。
Chart2D(System.Drawing.Size)	将 Chart2D 类的新实例初始化为 Size 指示大小的对象。

## Chart2D 属性:

名称	说明
ChartSize	获取或设置此实例的大小 (Size)。
ChartStyle	获取此实例的样式布局对象 ( <a href="#">ChartStyle</a> )。
DataCollections	获取或设置此实例的数据集合列表 ( <a href="#">List&lt;DataCollection&gt;</a> )。
Title	获取或设置此实例的标题对象 ( <a href="#">Title</a> )。
Label	获取或设置此实例坐标轴的标签对象 ( <a href="#">XYLabel</a> )。
XAxis	获取或设置此实例的横坐标轴对象 ( <a href="#">XAxis</a> )。
YAxis	获取或设置此实例的主纵坐标轴对象 ( <a href="#">YAxis</a> )。
Y2Axis	获取或设置此实例的次纵坐标轴对象 ( <a href="#">Y2Axis</a> )。
RAxis	获取或设置此实例的及坐标轴对象 ( <a href="#">RAxis</a> )。
Gird	获取或设置此实例的网格线对象 ( <a href="#">Gird</a> )。
Legend	获取或设置此实例的图例对象 ( <a href="#">Legend</a> )。

## Chart2D 方法:

名称	说明
AddChart()	返回此实例绘制的 Bitmap 对象。
SetChartSize(int, int)	将此实例的大小设置为指定的长, 宽。
SetTitle(string)	设置此实例的标题名称。
SetTitle(string, System.Drawing.Font, System.Drawing.Color)	设置此实例的标题的名称、字体和颜色。
SetLabelsName(string, string, string)	设置此实例的横坐标轴标签、主纵坐标轴标签和次纵坐标轴标签的名称为指定值。
SetXAxis(float, float, float)	设置此实例的横坐标轴的最大值、最小值和刻度间隔。
SetYAxis(float, float, float)	设置此实例的主纵坐标轴的最大值、最小值和刻度间隔。
SetY2Axis(float, float, float)	设置此实例的次纵坐标轴的最大值、最小值和刻度间隔。
SetRAxis(float, float, float)	设置此实例的极坐标轴的最大值、最小值和刻度间隔。
DataClear()	将此实例的数据集合列表的数据集清空。
Default()	将此实例除 Size 外的所有字段重置为默认值。
Clone()	返回一个新的 Chart2D 对象, 除数据集合列表外的所有元素和样式复制自此实例, 数据集合列表为空。

### 1.2.2 绘图区 ChartStyle

Readearth.Chart2D.ChartStyle 类用于设置绘图区域的颜色和边距，及元素布局绘制。

ChartStyle 属性：

名称	说明
ChartArea	获取此实例（绘图区）的 Rectangle 对象，只读属性。
PlotArea	获取此实例数据绘制区的 Rectangle 对象，只读属性。
ChartAreaColor	获取或设置此实例的图形颜色对象（ <a href="#">GraghColor</a> ）。
PlotAreaColor	获取或设置此实例的图形颜色对象（ <a href="#">GraghColor</a> ）。
XOffset	获取或设置此实例的横向边距（float）。
YOffset	获取或设置此实例的纵向边距（float）。

ChartStyle 方法：

名称	说明
Clone()	返回一个新的 ChartStyle 对象，复制自此实例。

### 1.2.3 图表组合 SubChart

Readearth.Chart2D.SubChart 类用于多表组合，可视为 Charts 的容器，主要包括大小，颜色，行、列数，边距，所含 Charts 等字段。

SubChart 构造函数：

名称	说明
SubChart()	初始化 SubChart 类的新实例，默认大小为长 800px、宽 500px。
SubChart(System.Drawing.Size)	将 SubChart 类的新实例初始化为指定大小的对象。
SubChart(System.Drawing.Size, int, int)	将 SubChart 类的新实例初始化为指定大小的、指定行列数的对象。

SubChart 属性：

名称	说明
Rows	获取或设置此实例的行数（int）。
Cols	获取或设置此实例的列数（int）。
Margin	获取或设置此实例四周的边距（int）。
SubCharts	获取或设置此实例所含的 Chart2D 数组（ <a href="#">Chart2D[,]</a> ）。
TotalChartArea	获取此实例（总绘图区）的 Rectangle 对象，只读属性。
TotalChartColor	获取或设置此实例的图形颜色对象（ <a href="#">GraghColor</a> ）。

SubChart 方法:

名称	说明
SetSubChartSize(int, int)	将此实例的大小设置为指定的长，宽。
GetSingleChartSize()	获取单个 chart 的大小。
AddSubCharts()	获取当前所含 Chart2D 绘制的 Bitmap 对象。
AddSubCharts(Chart2D[,])	添加 Chart2D 数组到此实例，并返回当前所含 charts 绘制的 Bitmap 对象。
.AddSubChart(Chart2D, int, int)	将一个 Chart2D 添加到指定位置，并返回当前所含 Chart2D 数组绘制的 Bitmap 对象。

### 1.2.4 图表类型简介 ChartType

此部分为图表类型简介，具体使用方法见第 2 章[操作说明](#)。

Chart2DTypeEnum:

名称	说明
None	无
LineChart	线形图
AreaChart	面积图
BarChart	条形图
PieChart	饼状图
PolarChart	极坐标图（暂不可用）

目前可用的类型有 LineChart，AreaChart， BarChart， PieChart，阶段性更新。

#### 1.2.4.1 线形图 LineCharts.LineChartTypeEnum

名称	说明
Line	折线图
Spline	曲线图
StairStep	阶梯图
Stem	杆状图
ErrorBar	误差线图
HiLo	股票图 1
HiLoOpenClose	股票图 2
Candle	股票图 3（又称 K 线图，蜡烛图）

**1.2.4.2 面积图 AreaCharts.AreaChartTypeEnum**

名称	说明
Area	堆积面积图
PercentArea	百分比面积图（暂不可用）

**1.2.4.3 条形图 BarCharts.BarChartTypeEnum**

名称	说明
Vertical	垂向柱状图
VerticalStack	垂向堆积柱状图
VerticalOverlay	垂向层叠柱状图
Horizontal	水平向柱状图
HorizontalStack	水平向堆积柱状图
HorizontalOverlay	水平向层叠柱状图

**1.2.4.4 饼状图 PieCharts.PieChartTypeEnum**

名称	说明
Standard	标准饼图
Partial	部分饼图
Doughnut	环形图（暂不可用）

**1.2.4.5 极坐标图 PolarCharts.PolarChartTypeEnum**

名称	说明
Radar	雷达图
Spline	极坐标曲线图
RadarPolygon	填充雷达图
Rose	玫瑰图

**极坐标图旋转方向 PolarCharts.AngleDirectionEnum**

名称	说明
CounterClockWise	逆时针方向
ClockWise	顺时针方向

## 1.3 Chart2D 基本元素

从 Chart2D 的基本结构总结可知，Chart2D 元素表现形式如下表所示：

元素名称	表现形式
标题 Title	文本
标签 Labels	文本
坐标轴 Axes	几何+文本
网格线 Grid	几何
图例 Legend	几何+文本

本节所列元素均包含在 Readearth.Chart2D.ChartElements 中。

### 1.3.1 标题 Title

Title 类用于表示图表标题元素的内容和样式。

	名称	说明
构造方法	Title()	初始化标题实例
属性	TitleName	获取或设置标题名称，默认为“Title”
属性	TitleStyle	获取或设置标题文本样式，默认字号为 12( <a href="#">TextStyle</a> )
方法	Clone()	返回一个新的标题对象，复制自此实例

在 [Chart2D](#) 类中，有直接设置方法 SetTitle(...)，方便修改。

### 1.3.2 标签 Labels

XYLabel 类用于表示图表各个坐标轴的标签元素。

	名称	说明
构造方法	XYLabel()	初始化标签实例
属性	XLabel	获取或设置横坐标轴标签内容，默认为“X Axis”
属性	YLabel	获取或设置主纵坐标轴标签内容，默认为“Y Axis”
属性	Y2Label	获取或设置次纵坐标轴标签内容，默认为“Y2 Axis”
属性	IsXLabelVisible	获取或设置横坐标轴标签是否可见，默认可见
属性	IsYLabelVisible	获取或设置主纵坐标轴标签是否可见，默认可见（暂不可用）
属性	IsY2LabelVisible	获取或设置次纵坐标轴标签是否可见，默认可见（暂不可用）
属性	XYLabelStyle	获取或设置标签文本样式 ( <a href="#">TextStyle</a> )
方法	Clone()	返回一个新的标签对象，复制自此实例



在 [Chart2D](#) 类中，有直接设置方法 `SetLabelsName(string, string, string)`，方便修改。

### 1.3.3 坐标轴 Axes

坐标轴元素包括 `XAxis`, `YAxis` 和 `Y2Axis` 三个类, `XAxis` 的刻度可能涉及时间文本，相对于 `YAxis` 和 `Y2Axis` 多一些针对刻度文本的属性和设置方法, `YAxis` 与 `Y2Axis` 较为相似，只需设置极值和文本样式，此外 `Y2Axis` 可选是否显示。

#### 1.3.3.1 XAxis

`XAxis` 类用于限制图表 X 坐标轴的值，并设置 X 坐标轴的刻度文本。

`XAxis` 构造方法：

名称	说明
<code>XAxis()</code>	初始化 X 轴实例

`XAxis` 属性：

名称	说明
<code>XLimMax</code>	获取或设置 X 轴最大值 (float)
<code>XLimMin</code>	获取或设置 X 轴最小值 (float)
<code>XTick</code>	获取或设置 X 轴刻度间隔 (float)
<code>XTickMark</code>	获取或设置 X 轴刻度文本 (string[])
<code>XMarkStartindex</code>	获取或设置 X 轴刻度文本起始刻度索引，默认为 0 (int)
<code>XMarkInterval</code>	获取或设置 X 轴刻度文本间隔，默认为 1 (int)
<code>XMarkFormat</code>	获取或设置 X 轴刻度时间文本格式 (string)
<code>XTickStyle</code>	获取或设置 X 轴刻度文本样式 ( <a href="#">TextStyle</a> )

`XAxis` 方法：

名称	说明
<code>SetXTickMark(string[])</code>	设置 X 轴刻度文本
<code>SetXTickMark(string[], int, int)</code>	设置 X 轴刻度文本、文本起始刻度和文本间隔
<code>SetXTickMark(string[], string)</code>	设置 X 轴刻度时间文本，以指定格式显示
<code>SetXTickMark(System.DateTime, long[], string)</code>	以指定时间为起始时间，将时间戳转换为指定格式的时间文本，设置为 X 轴刻度时间文本
<code>Clone()</code>	返回一个新的 X 轴对象，复制自此实例

在 [Chart2D](#) 类中，有方法 `SetXAxis(float, float, float)`，便于直接设置 X 轴的极值和刻度。

#### 1.3.3.2 YAxis

`YAxis` 类用于限制图表 Y 坐标轴的值，并设置 Y 坐标轴的刻度文本样式。

在 [Chart2D](#) 类中，有方法 `SetYAxis(float, float, float)`，便于直接设置 Y 轴的极值和刻

度。

	名称	说明
构造方法	YAxis ()	初始化 Y 轴实例
属性	YLimMax	获取或设置 Y 轴最大值 (float)
属性	YLimMin	获取或设置 Y 轴最小值 (float)
属性	YTick	获取或设置 Y 轴刻度间隔 (float)
属性	YTickStyle	获取或设置 Y 轴刻度文本样式 ( <a href="#">TextStyle</a> )
方法	Clone()	返回一个新的 Y 轴对象，复制自此实例

### 1.3.3.3 Y2Axis

Y2Axis 类用于限制图表 Y2 坐标轴的值，并设置 Y2 坐标轴的刻度文本样式。在 [Chart2D](#) 类中，有方法 SetY2Axis(float, float, float)，便于直接设置 Y2 轴的极值和刻度。

	名称	说明
构造方法	Y2Axis()	初始化 Y2 轴实例
属性	IsY2Axis	是否显示 Y2 轴
属性	Y2LimMax	获取或设置 Y2 轴最大值 (float)
属性	Y2LimMin	获取或设置 Y2 轴最小值 (float)
属性	Y2Tick	获取或设置 Y2 轴刻度间隔 (float)
属性	Y2TickStyle	获取或设置 Y2 轴刻度文本样式 ( <a href="#">TextStyle</a> )
方法	Clone()	返回一个新的 Y2 轴对象，复制自此实例

### 1.3.3.4 RAxis

RAxis 类用于限制图表极坐标轴 R 的样式和值，并设置极坐标轴的刻度文本。在 [Chart2D](#) 类中，有方法 SetRAxis(float, float, float)，便于直接设置极轴 R 轴的极值和刻度。

RAxis 构造方法：

名称	说明
XAxis()	初始化 R 轴实例

RAxis 属性：

名称	说明
RMax	获取或设置 R 轴最大值 (float)
RMin	获取或设置 R 轴最小值 (float)
RTick	获取或设置 R 轴刻度间隔 (float)
StartAngle	获取或设置 R 轴起始角度，默认为 0° (float)
AngleStep	获取或设置极径角度间隔，默认为 30° (float)

AngleDirection	获取或设置极径旋转方向，默认为逆时针 CounterClockWise ( <a href="#">PolarCharts.AngleDirectionEnum</a> )
RMark	获取或设置极径刻度文本 (string[])
RMarkInterval	获取或设置极径刻度文本间隔，默认为 1 (int)
RTickStyle	获取或设置 R 轴和极径刻度文本样式 ( <a href="#">TextStyle</a> )

RAxis 方法:

名称	说明
SetRMark(string[])	设置极径刻度文本
SetRMark(string[], int)	设置极径刻度文本和文本间隔
Clone()	返回一个新的 X 轴对象，复制自此实例

### 1.3.5 图例 Legend

Legend 类用于表示图表的图例元素，主要包括图例的方位，样式，内部布局等字段。

	名称	说明
构造方法	Legend ()	初始化图例实例
属性	IsLegendVisible	获取或设置图例是否可见
属性	IsInside	获取或设置图例是否位于数据绘制区内部，默认为内部
属性	.LegendPosition	获取或设置图例方位，默认值为右上角 NorthEast ( <a href="#">Legend.LegendPositionEnum</a> )
属性	xLgOffSet	获取或设置图例 X 方向外边距，默认值为 8 (float)
属性	yLgOffSet	获取或设置图例 Y 方向外边距，默认值为 8 (float)
属性	LegendColor	获取或设置图例图形颜色 ( <a href="#">GraphColor</a> )
属性	IsBorderVisible	获取或设置图例边框是否可见
属性	LgTextStyle	获取或设置图例文本样式，默认字号为 8 ( <a href="#">TextStyle</a> )
属性	IsVertical	获取图例文本排列方向是否为垂向，只读属性
方法	Clone()	返回一个新的图例对象，复制自此实例

Legend.LegendPositionEnum 包括八个方位，不同方位的图例文本排列方向有所区别。

名称	说明
East	右侧居中，文本垂向排列
NorthEast	右上角，文本垂向排列
SouthEast	右下角，文本垂向排列
West	左侧居中，文本垂向排列
NorthWest	左上角，文本垂向排列
SouthWest	左下角，文本垂向排列

North	顶部居中，文本水平向排列
South	底部居中，文本水平向排列

### 1.3.4 网格线 Grid

Grid 类用于表示图表的网格线元素。

	名称	说明
构造方法	Grid()	初始化网格线实例
属性	IsXGrid	获取或设置 X 方向网格线是否可见，默认可见
属性	IsYGrid	获取或设置 Y 方向网格线是否可见，默认可见
属性	GridStyle	获取或设置网格线条样式（ <a href="#">LineStyle</a> ）
属性	GridColor	获取或设置网格线颜色，等价于 GridStyle.LineColor
属性	GridPattern	获取或设置网格线虚实，等价于 GridStyle.LinePattern
属性	GridThickness	获取或设置网格线粗细，等价于 GridStyle.LineThickness
方法	Clone()	返回一个新的网格线对象，复制自此实例

## 1.4 Chart2D 基本图形样式

基于 Chart2D 中各种元素和数据的表现形式，将图形样式划分为文本样式，线条样式，符号样式，简单图形样式和颜色映射组。

样式使用情况如下表所示：

样式	使用元素
TextStyle	标题 Title，坐标轴 Labels，刻度标记 ticks，图例 Legend
LineStyle	数据系列 DataSeries，网格线 Grid
SymbolStyle	数据系列 DataSeries
GraphColor	ChartStyle（总区域 ChartArea，绘图区 PlotArea），图例 Legend
ColorMap	数据集 DataCollection

本节所列样式均包含在 Readearth.Chart2D. BasicStyle 中。

### 1.4.1 文本 TextStyle

TextStyle 类用于表示文本样式，主要指文本的颜色，字体和大小。

	名称	说明
构造方法	TextStyle ()	初始化文本实例
构造方法	TextStyle(System.Drawing.Font, System.Drawing.Color)	以指定字体、指定颜色初始化文本实例
属性	TextColor	获取或设置文本颜色
属性	TextFont	获取或设置文本字体，默认为字体为微软雅黑，字号为 10，字形为常规
属性	TextSize	获取或设置文本字号，等价于 TextFont.Size
方法	Clone()	返回一个新的文本样式对象，复制自此实例







### 1.4.2 线条 LineStyle

LineStyle 类用于表示线条样式，主要包括线条的颜色、宽度、虚实等字段。

	名称	说明
构造方法	LineStyle ()	初始化线条实例
构造方法	LineStyle(System.Drawing.Color, float, System.Drawing.Drawing2D.DashStyle)	以指定颜色、粗细，虚实初始化文本实例
属性	LineColor	获取或设置线条颜色
属性	LineThickness	获取或设置线条宽度

属性	LinePattern	获取或设置线条虚实样式 (System.Drawing.Drawing2D. DashStyle)
属性	IsVisible	获取或设置线条是否可见, 默认可见
方法	Clone()	返回一个新的线条样式对象, 复制自此实例

DashStyle 包括 6 种虚实样式:

名称	说明	图片示例
Solid	实线	
Dash	短划线状虚线	
Dot	点状虚线	
DashDot	划线点虚线	
DashDotDot	划线点点虚线	
Custom	自定义虚线	




### 1.4.3 符号 SymbolStyle

SymbolStyle 类用于表示特殊符号的样式, 主要包括符号的大小、颜色、类型等字段。

	名称	说明
构造方法	SymbolStyle ()	初始化符号实例
属性	SymbolType	获取或设置符号类型 (SymbolStyle.SymbolTypeEnum)
属性	SymbolSize	获取或设置符号大小, 默认为 6 (Size)
属性	SymbolColor	获取或设置符号颜色 ( <a href="#">GraphColor</a> )
属性	BorderThickness	获取或设置符号边框宽度, 默认为 1
方法	Clone()	返回一个新的符号对象, 复制自此实例

SymbolStyle.SymbolTypeEnum 包含 14 种符号:

名称	说明	图片示例
None	无	
Dot	圆点	
Circle	环形	
Triangle	三角形	
OpenTriangle	镂空三角形	
InvertedTriangle	倒三角形	
OpenInvertedTriangle	镂空倒三角形	
Box	正方形	
Square	镂空正方形	

Diamond	菱形	
OpenDiamond	镂空菱形	
Star	米字星形	
Plus	正十字	
Cross	斜十字	

#### 1.4.4 简单图形颜色 GraphColor

GraphColor 类用于表示简单图形的边框及内部填充颜色。

	名称	说明
构造方法	GraphColor ()	初始化图形颜色实例
构造方法	GraphColor(System.Drawing.Color, System.Drawing.Color)	以指定填充和边框颜色初始化图形颜色实例
属性	Fill	获取或设置图形内部填充颜色，默认为白色 (Color)
属性	Border	获取或设置图形边框颜色，默认为黑色 (Color)
方法	Clone()	返回一个新的符号对象，复制自此实例

#### 1.4.5 色彩映射 ColorMap

ColorMap 类表示一组渐变颜色，颜色使用 Alpha、Red、Green 和 Blue 表示，主要用于数据集合的颜色搭配。

	名称	说明
构造方法	ColorMap()	初始化颜色映射组实例
构造方法	ColorMap(int)	以指定拉伸级数初始化图形颜色实例
构造方法	ColorMap(int, int)	以指定拉伸级数、指定透明度初始化图形颜色实例
属性	Length	获取或设置颜色组拉伸级数，默认为 64
属性	AlphaValue	获取或设置颜色组透明度，默认为 255 (不透明)
方法	Spring()	返回一个从红到黄的渐变颜色组(int[Length, 4])
方法	Summer()	返回一个深绿到黄的渐变颜色组(int[Length, 4])
方法	Autumn()	返回一个从红到黄的渐变颜色组(int[Length, 4])
方法	Winter()	返回一个从蓝到绿的渐变颜色组(int[Length, 4])
方法	Hot()	返回一个从黄到白的渐变颜色组(int[Length, 4])
方法	Cool()	返回一个从青到紫的渐变颜色组(int[Length, 4])
方法	Gray()	返回一个从黑到白的渐变颜色组(int[Length, 4])

方法	<b>Jet()</b>	返回一个从蓝到青的渐变颜色组(int[Length ,4])
方法	<b>Clone()</b>	返回一个新的色彩映射对象，复制自此实例



## 1.5 Chart2D 数据组织

综合 Chart2D 的基本类型，我们把图表数据组织部分划分为数据集 DataCollection 和数据系列 DataSeries 两层，不同的图表类型使用不同类型的数据集，同时考虑到不同类型的表现形式组合在同一幅图表中的情况，因此在一个 Chart2D 对象中采用数据集集合列表的形式。

即一个 Chart2D 对象含有一个 List<DataCollection>（多个不同类型的数据集），每个 DataCollection 中含有多个 DataSeries。

本节罗列的数据类均包含在 Readearth.Chart2D.Data 中。

### 1.5.1 数据系列 DataSeries

DataRow 类表示数据系列，通过设置不同字段来适用不同类型的数据集。

DataRow 构造函数：

名称	说明
DataRow ()	初始化 DataRow 类的新实例。

DataRow 属性：

名称	说明	适用情况
SeriesName	获取或设置此数据系列的名称。	通用
IsY2Data	获取或设置此数据系列是否使用次纵坐标轴。	通用
PointList	获取或设置此数据系列的常规数据点列表（ArrayList）。	通用
LineStyle	获取或设置此数据系列的常规线条样式（LineStyle）。	通用
FillColor	获取或设置此数据系列的填充颜色（Color）。	通用
SymbolStyle	获取或设置此数据系列的符号样式（SymbolStyle）。	通用
LineChartType	获取或设置此数据系列的线形图表现形式（LineCharts.LineChartTypeEnum）	仅用于此数据系列置于标准数据集时
DataString	获取或设置此数据系列的股票数据点列表（string[,]）。	仅用于此数据系列置于标准数据集，且 LineChartType 为股票类图时
ErrorList	获取或设置此数据系列的误差数据点列表（ArrayList）。	仅用于此数据系列置于标准数据集，且 LineChartType 为误差线图 ErrorBar 时

ErrorLineStyle	获取或设置此数据系列的误差线条样式 ( <a href="#">LineStyle</a> )。	仅用于此数据系列置于 <a href="#">标准数据集</a> ，且 LineChartType 为误差线图 ErrorBar 时
IsExplode	获取或设置此数据系列是否突出显示。	仅用于此数据系列置于饼图数据集时

DataSet 方法:

名称	说明	适用情况
AddPoint(System.Drawing.PointF)	将指定二维数据点添加到此数据系列的常规数据点列表。	通用
AddErrorData(System.Drawing.PointF)	将指定二维数据点添加到此数据系列的误差数据点列表。	仅用于此数据系列置于 <a href="#">标准数据集</a> ，且 LineChartType 为误差线图 ErrorBar 时
AddData(float)	将指定一维数据添加到此数据系列的常规数据点列表。	仅用于此数据系列置于饼图数据集时

## 1.5.2 标准数据集 DataCollection 及其派生类

DataCollection 类表示标准数据集，主要用于线形图、散点图等简单二维图表的展示，也可作为基类派生出适用于其他二维图表类型的数据集。

派生标准：当某种图表的绘制过程中 DataSet 之间互不影响，且无特殊参数时，采用标准数据集，反之，则派生出适用此类图表的专用数据集。

DataCollection 构造函数:

名称	说明
DataCollection ()	初始化 DataCollection 类的新实例。

DataCollection 属性:

名称	说明
ChartType	获取或设置此数据集的展示类型。 (区别标准数据集和其他衍生数据集的字段)
DataSetList	获取或设置此数据集的数据系列列表 (ArrayList)。
CMap	获取或设置此数据集的颜色映射组 (int [DataSetList.Count, 4])。

DataCollection 方法:

名称	说明
Add(Readearth.Chart2D.Data.DataSeries)	添加一个数据系列对象到此数据集实例的数据系列列表的末尾。
Insert(int, Readearth.Chart2D.Data.DataSeries)	添加一个数据系列对象到此数据集实例的数据系列列表的指定位置。
Remove(string)	移除此数据集实例的数据系列列表中指定名称的数据系列对象。
RemoveAll()	清空此数据集实例的数据系列列表。
GetTypeCount(System.Enum)	返回此数据集实例的数据系列列表中某一种图表（如线形图等子类 Enum）的数据列个数，若传入的 Enum 不是图表类型则返回-999。

### 1.5.2.1 面积图专用数据集 DataCollectionArea

DataCollectionArea 类继承自标准数据集，适用于面积类图表，包含面积图种类、起始累积值等专有字段。

	名称	说明
构造方法	DataCollectionArea ()	初始化专用数据集实例。
重写属性	ChartType	获取此数据集适用类型，只读属性。
专有属性	AreaChartType	获取或设置此数据集的具体面积图种类。
专有属性	AreaAxis	获取或设置此数据集的起始累积值，默认为 0。
重写方法	GetTypeCount(System.Enum)	返回此数据集含有的某种面积图的数据列个数，若传入的 Enum 不是面积图种类则返回-999。

### 1.5.2.2 条形图专用数据集 DataCollectionBar

DataCollectionBar 类继承自标准数据集，适用于条形类图表，包含条形图种类、宽度占比等专有字段。

	名称	说明
构造方法	DataCollectionBar ()	初始化专用数据集实例。
重写属性	ChartType	获取此数据集适用类型，只读属性。
专有属性	BarChartType	获取或设置此数据集的具体条形图种类。
专有属性	BarWidth	获取或设置此数据集条形宽度占比，默认为 0.8。
重写方法	GetTypeCount(System.Enum)	返回此数据集含有的某种条形图的数据列个数，若传入的 Enum 不是条形图种类则返回-999。

### 1.5.2.3 饼状图专用数据集 DataCollectionPie

DataCollectionPie 类继承自标准数据集，适用于饼状类图表，包含饼状图种类、整体边框颜色等专有字段。

	名称	说明
构造方法	DataCollectionPie ()	初始化专用数据集实例。
重写属性	ChartType	获取此数据集适用类型，只读属性。
专有属性	PieChartType	获取或设置此数据集的具体饼状图种类。
专有属性	IsBorderVis	获取或设置此数据集边框是否可见，默认可见。
专有属性	BorderColor	获取或设置此数据集边框颜色，默认为黑色。
重写方法	GetTypeCount(System.Enum)	返回此数据集含有的某种饼状图的数据列个数，若传入的 Enum 不是饼状图种类则返回-999。

### 1.5.2.4 极坐标图专用数据集 DataCollectionPolar

DataCollectionPolar 类继承自标准数据集，适用于极坐标类图表，包含极坐标种类、整体边框颜色等专有字段。

	名称	说明
构造方法	DataCollectionPolar ()	初始化专用数据集实例。
重写属性	ChartType	获取此数据集适用类型，只读属性。
专有属性	PolarChartType	获取或设置此数据集的具体极坐标图种类。
专有属性	IsBorderVis	获取或设置此数据集边框是否可见，默认可见。
专有属性	RoseBarWidth	获取或设置此数据集玫瑰图宽度比例，默认 0.9。
专有方法	CheckNegative ()	是否通过玫瑰图数据检查，不允许出现负值。
重写方法	GetTypeCount(System.Enum)	返回此数据集含有的某种极坐标图的数据列个数，若传入的 Enum 不是极坐标图种类则返回-999。

## 2 Chart2D 操作说明

### 2.1 单幅单类型 Chart2D

主要操作步骤：

- (1) 创建一个 Chart2D；
- (2) 设置此 Chart2D 中各元素的样式；
- (3) 创建一个或多个数据系列，向各系列添加数据，并设置数据系列样式；
- (4) 创建一个数据集，向数据集中添加数据系列，并设置数据集的相关样式；
- (5) 将数据集添加到此 Chart2D；
- (6) 获取 Chart2D 绘制的位图对象。
- (7) 导出为图片文件。（可选步骤）

详细步骤说明：

#### 第一步：创建 Chart2D

两个方法：

- (1) `Chart2D chart2D1 = new Chart2D();`
- (2) `Size chartsize = new Size(500, 300);`  
`Chart2D chart2D2 = new Chart2D(chartsize);`

#### 第二步：设置Chart2D中各元素：

此部分可根据需要选择性设置，未设置元素按照默认样式显示，各元素具体样式属性可参考1.3节。

例如：

```
chart2D1.SetTitle("图一标题");
chart2D1.SetLabelsName("X轴", "Y轴", "");
chart2D1.SetXAxis(0, 8, 2);
chart2D1.SetYAxis(-2, 3, 0.5f);
chart2D1.Legend.LegendPosition =
Readearth.Chart2D.ChartElements.Legend.LegendPositionEnum.North;
chart2D1.ChartStyle.PlotAreaColor.Border = Color.White;
```

**注意1：**若未设置坐标轴极值或设置值与数据不相符，则按照传入数据系列自动调整，否则不调整，是否经过调整用`IsAxisFit`属性表示。

**注意2：**`XTickMark`的设置方式有三种，适用于不同类型的marks。

1: 普通字符串

```
string[] ticks1 = new string[] { "2015.01", "2016.1", "2017.01", "2018.01",  
"2019.01" };
```

```
chart2D1.XAxis.XTickMark = ticks1;
```

2: 时间字符串

```
string[] ticks2 = new string[] { "2015.01", "2016.1", "2017.01", "2018.01",  
"2019.01" };
```

```
chart2D1.XAxis.SetXTickMark(ticks2, "yyyyMMdd");
```

3: 时间戳

```
long[] times = new long[] { -3122798400, -3120249600, -3117700800, -3115065600,  
-3112430400, -3109795200, -3107160000, -3104481600, -3101846400, -3099211200,  
-3096576000, -3093940800, -3091262400};  
  
chart2D1.SetXAxis(times[0], times[times.Length - 1], (times[times.Length - 1] -  
times[0]) / (times.Length - 1));  
  
chart2D1.XAxis.SetXTickMark(Convert.ToDateTime("01-JAN-1970 00:00:00"), times,  
"yyyyMMdd HH");
```

**注意 3:** 条形图垂向绘制时, XTickMark 起始位置需要设置为 1 (条形宽度特性要求)。

```
chart2D3.XAxis.XMarkStartindex = 1; //默认为 0
```

**注意 4:** 绘制极坐标图时, 应设置的坐标轴为 RAxis, 且 AngleStep 应与下一步中各数据点之间的角度间隔一致。

```
chart2D2.SetRAxis(0, 16f, 4f);  
chart2D2.RAxis.AngleStep = 30; //默认为30  
chart2D2.RAxis.AngleDirection = PolarCharts.AngleDirectionEnum.ClockWise; //默认为  
CounterClockWise  
  
chart2D2.RAxis.SetRMark(new string[] { "E", "S", "W", "N" }, 3);
```

### 第三步: 创建数据系列并设置样式:

----创建数据系列:

```
DataSet ds1 = new DataSet();
```

----设置名称和样式:

```
ds1.SeriesName = "Cos(x)";  
ds1.LineChartType = LineCharts.LineChartTypeEnum.Stem; //默认为Line折线图  
ds1.LineStyle.LineColor = Color.Red; //默认线性为Color.Black, 1f, DashStyle.Solid,  
//ds1.LineStyle.IsVisible = false;  
ds1.SymbolStyle.SymbolType = SymbolStyle.SymbolTypeEnum.Dot;  
ds1.SymbolStyle.SymbolColor = new GraphColor(Color.Yellow, Color.Red);
```

----添加数据点:

```
for (int i = 0; i < 16; i++)
{
    ds1.AddPoint(new PointF(i / 2.0f, (float)Math.Cos(i / 2.0f)));
}
```

**注意 1:** 若此数据列绘制为线形图的一种，则需要设置 `LineChartType` 为所需种类，若不绘制为线形图，则不需设置 `LineChartType`。

**注意 2:** 常规数据点添加使用 `AddPoint` 的方法（直角坐标数据 `Point` 为 (X, Y)，极坐标数据 `Point` 为 (Angle, Value)，Angle 的范围为 [0, 360)），若此数据列绘制为特殊线形图或饼状图，则使用专用方法。

1: 误差线图：添加常规数据 `AddPoint` 和误差数据 `AddErrorData`

```
for (int i = 4; i < 22; i++)
{
    ds1.AddPoint(new PointF(i / 2f, 10.0f * (float)Math.Exp(-i / 2f)));
}
for (int i = 4; i < 22; i++)
{
    ds1.AddErrorData(new PointF(i / 2f, 2.0f / (i / 2f)));
}
```

2: 三种股票类线形图：使用 `TextFileReader` 类解析文件，并添加到 `DataString`

```
TextFileReader tfr = new TextFileReader();
ds1.DataString = tfr.ReadTextFile("GE.txt");
```

3: 饼状图：一维数据，每一数据系列只添加一个值

```
float[] data = new float[5] { 15, 135, 10, 8, 30 }; // 各系列数据
string[] labels = new string[5] { "Soc. Sec. Tax", "Income Tax", "Borrowing",
"Corp. Tax", "Misc." }; // 各系列名称
DataSeries ds0 = new DataSeries();
ds0.SeriesName = labels[0];
ds0.AddData(data[0]);
```

**注意 3:** 请根据要展示的方式设置相应的样式，例如面积图、柱状图、饼状图等需要设置 `FillColor`，误差线图需要设置 `ErrorLineStyle` 等。

**注意 4:** 颜色设置方式有两种，一种是在每个 `DataSeries` 中设置 `LineStyle.LineColor` 或 `FillColor`，另一种是将所有 `DataSeries` 添加到 `DataCollection` 中后设置 `CMap`。

**第四步：创建数据集并设置样式：**

----创建数据集：

```
DataCollection dc11 = new DataCollection();
```

----设置数据集类型和相关参数：

```
//dc11.ChartType = Chart2DTypeEnum.LineChart; //标准数据集无需设置
```

----添加上一步创建的数据系列：

```
dc11.Add(ds1);
```

```
dc11.Add(ds2);
```

----设置颜色映射组（可选）：

```
Readearth.Chart2D.BasicStyle.ColorMap cm = new
```

```
Readearth.Chart2D.BasicStyle.ColorMap(dc11.DataSeriesList.Count);
```

```
dc11.CMap = cm.Jet ();
```

**注意 1：**创建标准数据集时，无需设置 **ChartType** 属性（即标准数据集中设置 **DataSeries** 层的 **LineChartType**），创建专用数据集时，设置专用 **Type** 属性和相关参数。

1: 面积图

```
DataCollectionArea dc12 = new DataCollectionArea();
```

```
dc12.AreaChartType = AreaCharts.AreaChartTypeEnum.Area;
```

```
//dc12.AreaAxis = 0; //默认值为 0
```

2: 条形图

```
DataCollectionBar dc2 = new DataCollectionBar();
```

```
dc2.BarChartType = BarCharts.BarChartTypeEnum.Vertical ;
```

```
dc2.BarWidth = 0.9f;
```

3: 饼状图

```
DataCollectionPie dc3 = new DataCollectionPie();
```

```
dc3.PieChartType = PieCharts.PieChartTypeEnum.Standard;
```

```
dc3.IsBorderVis = false;
```

4: 极坐标图

```
DataCollectionPolar dc4 = new DataCollectionPolar();
```

```
dc4.PieChartType = PolarCharts.PolarChartTypeEnum.Rose;
```

```
dc4.IsBorderVis = false;
```

```
dc4.RoseBarWidth = 0.9f;
```

**注意 2：**设置 **CMap** 时可以采用 **ColorMap** 类映射赋值 **CMap**，也可以自定义颜色组赋值 **CMap**，但必须确保赋值之前所有所需设置颜色的数据系列已添加入此数据集。

**第五步：将数据集添加到此 Chart2D**

```
chart2D1.DataCollections.Add(dc11);
```

**第六步：获取Chart2D绘制的位图对象**



```
Bitmap chartImage = chart2D1.AddChart();
```

### 第七步：导出图片

设置文件路径，使用Readearth.Chart2D.Additional中MakePicture类的导出方法。

```
string folder = "E:/Readearth/charts01.bmp";  
Readearth.Chart2D.Additional.MakePicture.OutTheImageToPic(chartImage, folder);
```

## 2.2 单幅多类型 Chart2D

主要操作步骤：

- (1) 创建一个 Chart2D；
- (2) 设置此 Chart2D 中各元素的样式；
- (3) 创建一个或多个数据系列，向各系列添加数据，并设置数据系列样式；
- (4) 创建多个数据集，向数据集中添加数据系列，并设置数据集的相关样式；
- (5) 将各数据集添加到此 Chart2D；
- (6) 获取 Chart2D 绘制的位图对象。
- (7) 导出为图片文件。（可选步骤）

单幅多类型 Chart2D 的绘制步骤与单幅单类型 Chart2D 的相似，唯一区别是在第五步，单类型图表的数据集列表只有一个专用数据集或多个标准数据集，多类型图表的数据集可以是不同类型的。

不同直角坐标数据集的叠加显示顺序从上到下依次为Lines、Bar、Area，与数据集加入列表的顺序无关；各数据集内部绘制顺序为数据列的顺序，显示顺序与之相反。

不同极坐标数据集的叠加显示顺序与数据集加入列表的顺序相反，因此需注意填充雷达图（RadarPolygon）与雷达图、曲线图的绘制顺序，先添加填充雷达图的数据集；各数据集内部绘制顺序为数据列的顺序，显示顺序与之相反。

本类库绘制前进行数据集类型兼容性检查，若数据集不兼容或列表为空时，返回的位图对象只有 Chart2D 基本元素，即无数据无图例的空白底表。

## 2.3 多幅多类型 Chart2D

主要操作步骤：

- (1) 创建一个 SubChart;
- (2) 获取 SubChart 中单个 Chart 的大小，使用此指定大小创建多个 Chart2D;
- (3) 按照 2.2 或 2.3 中前五个步骤设置各 Chart2D;
- (4) 将各 Chart2D 添加到此 SubChart;
- (5) 获取 SubChart 绘制的位图对象。
- (6) 导出为图片文件。(可选步骤)

详细步骤说明：

### 第一步：创建 SubChart

三种方法：

- (1) `SubChart subs = new SubChart();`
- (2) `Size subchartsize = new Size(800, 500);`  
`SubChart subs = new SubChart(subchartsize);`
- (3) `Size subchartsize = new Size(800, 500);`  
`SubChart subs = new SubChart(subchartsize, 2, 2);`

### 第二步：获取 SubChart 中单个 Chart 的大小，并创建 Chart2D

当 SubChart 中为多行、多列时，每个 Chart2D 的大小是均等的。

```
Size chartSize = subs.GetSingleChartSize();
chart2D1 = new Chart2D(chartSize);
chart2D2 = chart2D1.Clone();
chart2D3 = chart2D1.Clone();
chart2D4 = chart2D1.Clone();
```

注意：若单个 Chart2D 的大小与 SubChart 均分之后的大小不一致时，图片容易出现拉伸或缺损现象，因此建议使用 `subs.GetSingleChartSize()` 方法，或 `chart2D1.SetChartSize(int,int)` 方法，或修改 `chart2D1.ChartSize` 属性，使图片正确显示。

### 第三步：按照 2.2 或 2.3 中前五个步骤设置各 Chart2D

此部分参考 2.2 和 2.3。

### 第四步：将各 Chart2D 添加到此 SubChart

若采用如下形式添加，则此步骤与第三步顺序可调换。也可与第五步合并添加。

```
subs.SubCharts = new Chart2D[2, 2] { { chart2D1, chart2D2 }, { chart2D3, chart2D4 } };
```

### 第五步：获取 SubChart 绘制的位图对象

三种方法：

1：若已添加所有 Chart2D，可直接获取位图对象。（结合第四步使用）

```
Bitmap chartImage = subs.AddSubCharts();
```

2：若未添加所有 Chart2D，可添加后获取位图对象。

```
Bitmap chartImage = subs.AddSubCharts( new Chart2D[2, 2]
                                         { { chart2D1, chart2D2 }, { chart2D3, chart2D4 } } );
```

3：也可添加某一 Chart2D 到一指定位置，获取位图对象。

```
Bitmap chartImage = subs.AddSubCharts(chart2D1, 0, 0); //行列起始索引均为 0
```

### 第六步：导出为图片文件

与 2.1 导出方法相同。

图片示例如下：

