[**DevExpress之ChartControl实现时间轴实例 z**](http://www.cnblogs.com/zeroone/p/4790004.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 | using System;    using System.Data;    using System.Windows.Forms;    using DevExpress.XtraCharts;   namespace DevExpressChart    {        public partial class winDateTime : Form        {            public winDateTime()            {                InitializeComponent();            }           private void winDateTime\_Load(object sender, EventArgs e)            {                BuilderDevChart();            }            private DataTable CreateChartData()            {                DataTable table = new DataTable("Table1");                table.Columns.Add("Date", typeof(DateTime));                table.Columns.Add("Value", typeof(Int32));                table.Rows.Add(new object[] { new DateTime(2014, 5, 21), 10 });                table.Rows.Add(new object[] { new DateTime(2014, 6, 22), 20 });                table.Rows.Add(new object[] { new DateTime(2014, 7, 23), 40 });                table.Rows.Add(new object[] { new DateTime(2014, 8, 24), 20 });                table.Rows.Add(new object[] { new DateTime(2014, 9, 25), 30 });                return table;            }            private void BuilderDevChart()            {                Series \_lineSeries = new Series("利润", ViewType.Line);                \_lineSeries.ArgumentScaleType = ScaleType.DateTime;                \_lineSeries.ArgumentDataMember = "Date";                \_lineSeries.ValueDataMembers[0] = "Value";                \_lineSeries.DataSource = CreateChartData();                chartControl1.Series.Add(\_lineSeries);               chartControl1.SetTimeAxisX(DateTimeMeasurementUnit.Month, DateTimeMeasurementUnit.Month, "yyyy-MM");                //---------------------------------------------------------------               //\_lineSeries.ValueScaleType = ScaleType.DateTime;                //\_lineSeries.ArgumentDataMember = "Value";                //\_lineSeries.ValueDataMembers[0] = "Date";                //\_lineSeries.DataSource = CreateChartData();                //chartControl1.Series.Add(\_lineSeries);               //XYDiagram \_diagram = (XYDiagram)chartControl1.Diagram;                //\_diagram.AxisY.DateTimeMeasureUnit = DateTimeMeasurementUnit.Month;                //\_diagram.AxisY.DateTimeGridAlignment = DateTimeMeasurementUnit.Month;                //\_diagram.AxisY.DateTimeOptions.Format = DateTimeFormat.Custom;                //\_diagram.AxisY.DateTimeOptions.FormatString = "MMMM";            }        }    } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243 | using DevExpress.Utils;    using DevExpress.XtraCharts;   namespace DevExpressChart    {        public static class ChartUtils        {            /// <summary>            /// 增加数据筛选            /// </summary>            /// <param name="SeriesBase">Series</param>            /// <param name="columnName">列名称</param>            /// <param name="value">列名称对应的筛选数值</param>            /// <param name="dataFilterCondition">DataFilterCondition枚举</param>            public static void AddDataFilter(this SeriesBase series, string columnName, object value, DataFilterCondition dataFilterCondition)            {                series.DataFilters.Add(new DataFilter(columnName, value.GetType().FullName, dataFilterCondition, value));            }           /// <summary>            /// 设置X轴Lable角度            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="angle">角度</param>            public static void SetXLableAngle(this ChartControl chart, int angle)            {                XYDiagram \_xyDiagram = (XYDiagram)chart.Diagram;                if (\_xyDiagram != null)                    \_xyDiagram.AxisX.Label.Angle = angle;            }            /// <summary>            ///  设置Y轴Lable角度            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="angle">角度</param>            public static void SetYLableAngle(this ChartControl chart, int angle)            {                XYDiagram \_xyDiagram = (XYDiagram)chart.Diagram;                \_xyDiagram.AxisY.Label.Angle = angle;            }            /// <summary>            /// 设置ColorEach            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="colorEach">是否设置成ColorEach</param>            public static void SetColorEach(this Series series, bool colorEach)            {                SeriesViewColorEachSupportBase colorEachView = (SeriesViewColorEachSupportBase)series.View;                if (colorEachView != null)                {                    colorEachView.ColorEach = colorEach;                }            }            /// <summary>            /// 设置是否显示十字标线            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="crosshair">是否显示十字标线</param>            public static void SetCrosshair(this ChartControl chart, bool crosshair)            {                chart.CrosshairEnabled = crosshair ? DefaultBoolean.True : DefaultBoolean.False;                chart.CrosshairOptions.ShowArgumentLabels = crosshair;                chart.CrosshairOptions.ShowArgumentLine = crosshair;                chart.CrosshairOptions.ShowValueLabels = crosshair;                chart.CrosshairOptions.ShowValueLine = crosshair;            }            /// <summary>            /// 新增ChartControl的Title文字            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="title">Title文字</param>            public static void AddTitle(this ChartControl chart, string title)            {                ChartTitle \_title = new ChartTitle();                \_title.Text = title;                chart.Titles.Add(\_title);            }            /// <summary>            /// 饼状Series设置成百分比显示            /// </summary>            /// <param name="series">Series</param>            public static void SetPiePercentage(this Series series)            {                if (series.View is PieSeriesView)                {                    ((PiePointOptions)series.PointOptions).PercentOptions.ValueAsPercent = true;                    ((PiePointOptions)series.PointOptions).ValueNumericOptions.Format = NumericFormat.Percent;                    ((PiePointOptions)series.PointOptions).ValueNumericOptions.Precision = 0;                }            }            /// <summary>            /// 将X轴格式化成时间轴            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="dateTimeMeasureUnit">X轴刻度单位</param>            /// <param name="dateTimeGridAlignment">X轴刻度间距的单位</param>            public static void SetTimeAxisX(this ChartControl chart, DateTimeMeasurementUnit dateTimeMeasureUnit, DateTimeMeasurementUnit dateTimeGridAlignment)            {                XYDiagram \_diagram = (XYDiagram)chart.Diagram;                if (\_diagram != null)                {                    \_diagram.AxisX.DateTimeMeasureUnit = DateTimeMeasurementUnit.Month;//X轴刻度单位                    \_diagram.AxisX.DateTimeGridAlignment = DateTimeMeasurementUnit.Month;//X轴刻度间距                }            }            /// <summary>            /// 将X轴格式化成时间轴            /// </summary>            /// <param name="chart">ChartControl</param>            /// <param name="dateTimeMeasureUnit">X轴刻度单位</param>            /// <param name="dateTimeGridAlignment">X轴刻度间距的单位</param>            /// <param name="formatString">时间格式；eg:yyyy-MM</param>            public static void SetTimeAxisX(this ChartControl chart, DateTimeMeasurementUnit dateTimeMeasureUnit, DateTimeMeasurementUnit dateTimeGridAlignment, string formatString)            {                XYDiagram \_diagram = (XYDiagram)chart.Diagram;                if (\_diagram != null)                {                    \_diagram.AxisX.DateTimeMeasureUnit = DateTimeMeasurementUnit.Month;//X轴刻度单位                    \_diagram.AxisX.DateTimeGridAlignment = DateTimeMeasurementUnit.Month;//X轴刻度间距                    \_diagram.AxisX.DateTimeOptions.Format = DateTimeFormat.Custom;                    \_diagram.AxisX.DateTimeOptions.FormatString = formatString;                }            }        }    } |

 运行效果如下图所示：

DevExpress ChartControl加载大数据量数据时的性能优化方法有哪些？

关于图表优化，可从以下几个方面解决：

1.关闭不需要的可视化的元素(如LineMarkers, Labels等): Series.View.LineMarkerOptions.Visible =false.

2. 关闭图表的滚动与缩放功能，手动调整范围，这样将大大减少所需计算的个数。

3. 将 ChartControl.RefreshDataOnRepaint属性设为false

4. 将 ChartControl.CacheToMemory属性设为true，这样将通过缓存提供图表性能及内存消耗。

5. 将ChartControl.RuntimeHitTesting属性设为false。这样将禁用聚焦检查及测试，提供图表重绘速度.

\_\_\_\_

http://www.devexpresscn.com/DevExpress-Resources/Documentation-17.html