# Introduction

MSChart Extension is an extension package released under MIT license, created for Microsoft Chart Controls. The package is first published in July 2012 (See: [MSChart Extension - Zoom and Pan Control](http://www.codearteng.com/2012/05/mschart-extension-zoom-and-pan-controls.html)) to overcome limitation of the original chart component and provide additional features.

The source code and the release package is available for download at:

* [Download from GitHub](https://github.com/Code-Artist/MSChartExtension)
* [Download from NuGet](https://www.nuget.org/packages/MSChartExtension/)

# Background

Microsoft chart is a great data visualization component for displaying data with rich features for data analysis and annotations. It come with lots of features. Unfortunately, most of the interactive features are not ready accessible without deep knowledge on this component. MSChart Extension is created with aimed to unleash its power. Development of MSChart Extension mainly focused on using X-Y chart series for data and waveform analysis.

# Features List

* Fast Clear Data
* Zoom (Windowed, X-Axis, Y-Axis, Mouse-Wheel)
* Pan
* Show / Hide Series
* Annotations
* Snap to Nearest Data
* Cursors
* Snap to Nearest Data Point
* Down-sampling large data
* Configurable cursor and series color.
* Theme.

# Using MSChart Extension

To use this extension package, add **MSChartExtension by Code Art Engineering** to your project from NuGet Package Manager, then enable additional features and built-in context menu by calling the following method:

ChartOption ChartOptions = new ChartOption();

MyChart.EnableZoomAndPanControls( ChartCursorSelected, ChartCursorMoved, ZoomChanged, ChartOptions)  
  
private void ChartCursorSelected(Chart sender, ChartCursor e)  
{

TxtPos.Text = e.XFormattedString + ", " + e.YFormattedString;

}

## Callback Functions

ChartCursorSelected and ChartCursorMoved are two callback to notify parent when chart coordinate changed. ChartCursorSelected will be called when a point in chart area is selected whereas ChartCursorMoved will trigger when mouse is moving in any of the chart area. Details of the event is returned in ChartCursor object which contains the X, Y values as well as the respective ChartArea which triggered these callbacks.

It’s recommended to use XFormattedString and YFormattedString for displaying the value in predefined formatted string where prefix, postfix and string format can be configure in ChartOption object under properties prefix with CursorLabel (CursorLabelPrefixX1, CusorLabelStringFormatX1 and etc…)