Pivot Charting

Overview

WebGrid.NET Enterprise® 6.0 redefines data visualization concept by introducing a new breakthrough *Pivot Charting* feature. The Pivot Charting is offered as built-in feature, including the automatic data processing and data aggregation. With Pivot Charting feature, transforming raw data from the Grid View into a stunning Chart View is only a matter of single button click.

This section provides the overview and fundamental information of Pivot Charting so that you can quickly getting started with this new powerful feature.

What is Pivot Charting

Definition

Pivot Charting is a powerful data analysis and visualization tool that enables you to present your dataset in multiple viewpoints. Pivot Charting allows you to look at data in several dimensions; for example, sales by region, sales by sales rep, sales by product category, sales by month, etc. Such capability is provided in numerous decision support applications under various function names. The Pivot Charting terminology is first introduced in Microsoft Excel XP and Microsoft Access XP.

The following illustration describes the concept of Pivot Charting in general.

The ability to quickly switch between one slice of data and another allows users to analyze their information in small palatable chunks instead of a giant report that is confusing.

How it works

A Pivot Chart has four essential elements:

- Data fields: The Y-axis of the data space. This data field displays one of the aggregate functions (sum, count, max, min ...) of a data series.
- Series fields: The Z-axis of the data space. This data field represents a series of unique data available in the data source which appear in each data category.
- Category fields: The X-axis of the data space. The data must be categorical.
- Filter fields: The fields used to filter the data source from global level.

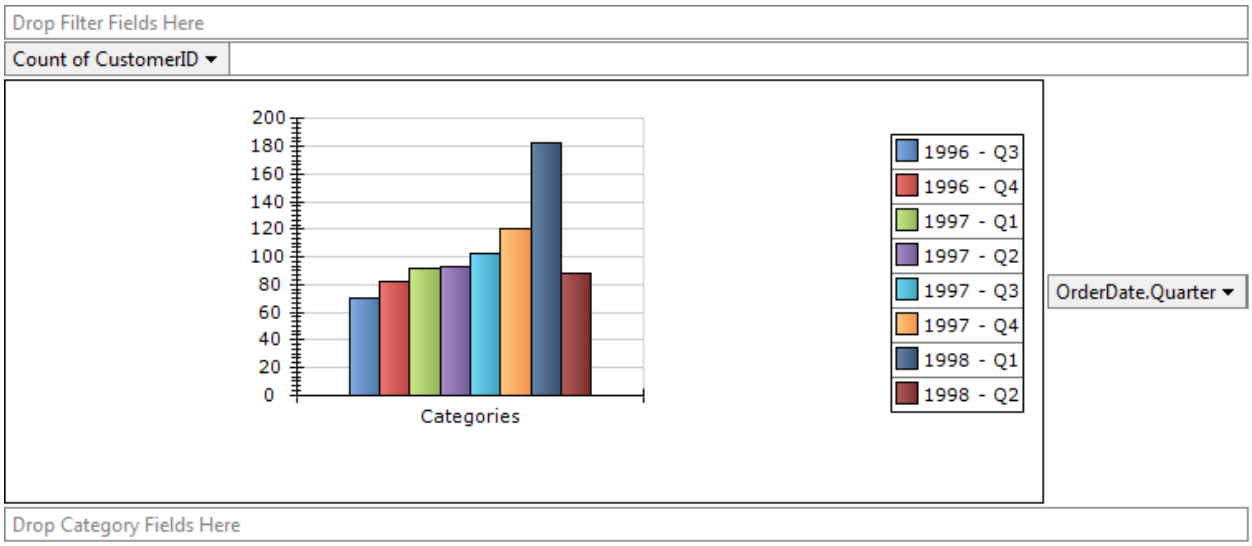
You can assign multiple fields to the Data fields, Series fields, Category fields and Filter fields.

For example, you can assign "Year" as the major category; "Country" as the minor category; and divide the GDP into distinct sectors of the industry. Or you may drag "Country" from the category field to the page field if you find your table too crowded.

You can choose one of the built-in aggregation methods for each *Data field*. The common aggregation methods are:

- Sum
- Count
- Min
- Max
- Average
- Standard Deviation
- Variance

Pivot Charting is an exact visualization of *Pivot Table* in Graph format instead of text. The Pivot Table itself has the same concept with Pivot Charting as mentioned above. The following illustration describes the sales data by order date in quarterly time interval.



A graphical data presentation with built-in aggregation, instead of *raw flat data*.

While in the Pivot Charting view, WebGrid takes the same datasource used by the Grid view and perform data pivoting and aggregation based on the specified root data member. The *Chart Field List* is retrieved from the available columns in the Grid view.

By default, the Pivot Charting view will show an empty Chart for the first time unless the initial series, data and category field is specified. You can start to analyze your data by dragging one of the fields in the *Chart Field List* to series, data and category zone.

Pivot Charting in WebGrid.NET Enterprise 6.0

WebGrid.NET Enterprise® 6.0 takes data presentation and visualization component to a new level by providing "out-of-the-box" Pivot Charting functionalities such as those provided in Microsoft Access® 2007.

The core functions of Pivot Charting included in WebGrid.NET Enterprise 6.0 are:

- **Ability to specify multiple fields to Data, Series, Categories and Filter**. This function allows you to create multi-dimensional viewpoint of your data.
- **Ability to specify one of the aggregation methods for each Data field**. The default aggregation method is Count.
- **Ability to perform auto filtering**. You can easily slice your data by excluding the category or series that you don't want to see in the visualized Chart.
- **Ability to sort Category and Series field automatically in ascending or descending way**. This function enables you to quickly display sorted data in the visualized Chart.
- **Ability to switch between Category and Series fields in one button click**. This function enables you to easily inspect your data in different perspective.
- **Ability to break down DateTime field into five time intervals automatically**. This powerful function automatically breaks down DateTime field into the following time intervals: Year,

Quarter, Month, Week, Date. In the fraction of seconds, you can now visualize a series of data grouped by any time interval you preferred. For instance, order date in quarterly interval.

In addition to the comprehensive core pivoting functions above, WebGrid.NET Enterprise® 6.0 is bundled with the *best-of-breed Charting Engine* and *industry's most popular User Interface*. The combination of both delivers the most sophisticated and the most advanced data visualization functions for enterprise ASP.NET Web development. The addition with the hundreds of existing features delivers perfect values that enable you to build just anything – from basic flat table presentation, hierarchical view, columnset presentation to a Pivot Charting-enabled presentation.

The Charting Engine included in WebGrid.NET Enterprise® 6.0 is a royalty-free redistributable runtime Charting Engine which exclusively licensed from world's leading Charting provider, Nevron LLC. Nevron LLC is also the technology partner of Intersoft Solutions Corp. The included runtime Charting Engine is designed only for WebGrid to perform call to the visualization functions – and doesn't include the license for development environment.

The Charting Engine Key Features:

- Over 50 chart types that you can customize at runtime.
- High performance visualization. Ability to process large data points in fraction of seconds.
- 3D predefined projection.
- 3D predefined lighting and multiple light sources.
- Visual effects, such as bevel, emboss, blur, transparency and more.
- Styles and appearance. Ability to customize the styles of lines, titles, legend, wallpaper, docking, color series, markers and much more.

The User Interface Key Features:

- **Drag and Drop**. Conveniently drag a field from Chart Field List to any of four available drop zones. You can also drag any field element from any drop zone to another in the same way and manner as in Microsoft Access® 2007.
- **Ribbon UI**. The UI adopts Microsoft's latest User Interface innovation used in Microsoft Office System 2007. The built-in Ribbon UI allows end users to easily understand the commands provided by the Pivot Charting.
- **Context Menu**. Easily toggle the visibility of user interface elements such as the Ribbon, Legend, Chart Field List and so on. This allows end users to hide one or more user interface elements for maximum screen real estate.
- **AJAX Chart Image Generation**. The Chart image is generated in near real-time by using Intersoft's high performance built-in AJAX technology, called FlyPostBack™

For detailed User Interface and Interactivity features, please refer to Pivot Charting User Interface

Pivot Charting and Business Intelligence

WebGrid.NET Enterprise® 6.0 includes built-in data analytical and data processing functionality which automatically transforms raw data into pivot data view for the Pivot Charting rendering. This feature works in conjunction with the Grid View's datasource, which needs to show the raw flat data.

With the built-in data analytical and processing functions, this means you are not required to build complex data warehousing or separate OLAPS server. You can simply retrieve the datasource as usual and bind it to the Grid instance by either using DataSource Control or custom datasource object.

The pivot data analysis and processing offered in WebGrid.NET Enterprise® 6.0 is a simplified approach to build *Business Intelligence* application. Business Intelligence (BI) is a far more complex system which takes advantage of multiple technologies such as *data warehousing*, *data dimensioning*, *cubes* and *data mining*. One of the most popular Business Intelligence tool today is Microsoft SQL Server 2005 Analysis Services.

The built-in data analytical function in WebGrid.NET Enterprise® 6.0 is a scale-down version of Business Intelligence which primarily focused on single datasource and Charting projection of the pivot data. The concept adopted in WebGrid.NET Enterprise® 6.0 is more similar to those in Microsoft Access 2007® rather than SQL Server 2005 Analysis Services.

Although you can't connect directly to SQL Server 2005 Analysis Services data sources such as its cubes, dimensions etc – you can build a comprehensive query in your OLTP relational database to get a combined result to simulate the cubes in Analysis Services. When bound to WebGrid, the available fields can be specified as the facts and measures. You can have more than one facts and measures in the current implementation of Pivot Charting in WebGrid.NET Enterprise® 6.0, which allows you to build a simplified Business Intelligence application.

In the future releases, WebGrid.NET Enterprise® will include more comprehensive functions to process larger Business Intelligence application by offering out-of-the-box integration with MDX technology and Analysis Services. This is made possible with the robust architecture of the current release which is extensible to the path of larger BI presentation implementation.

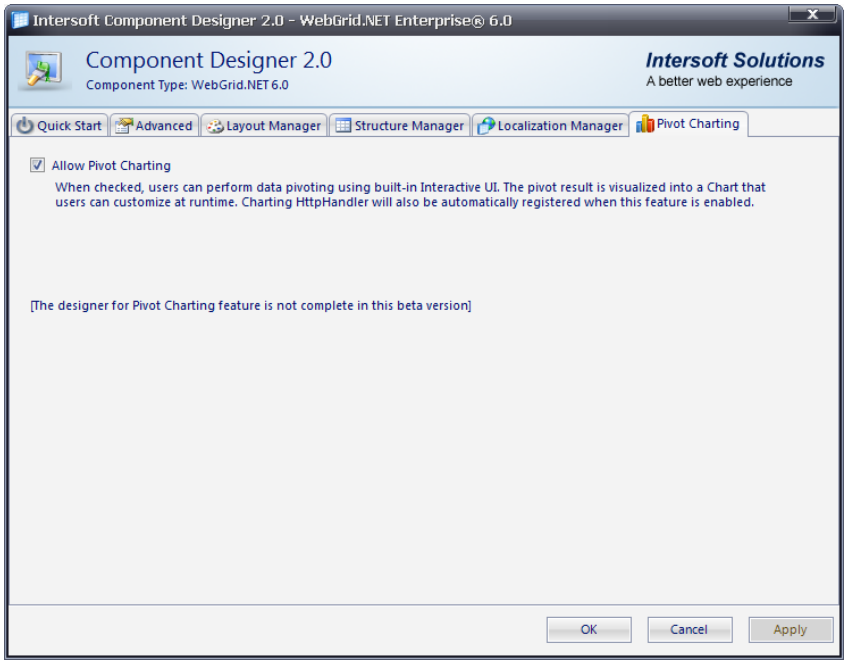
Getting Started

In addition to the wealth of Pivot Charting functions and comprehensive object model, WebGrid.NET Enterprise 6.0 is designed to let you take advantage of these advanced features quickly and easily.

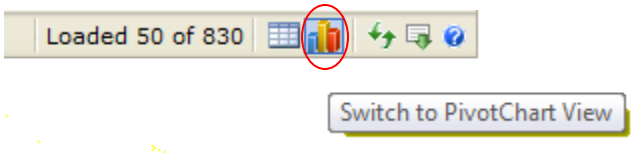
The Pivot Charting feature can be enabled in as easy as three steps:

1. Launch **WebGrid.NET Designer** for the WebGrid instance.
2. Select Pivot Charting tab, then check “Enable Pivot Charting”.
3. Click OK.

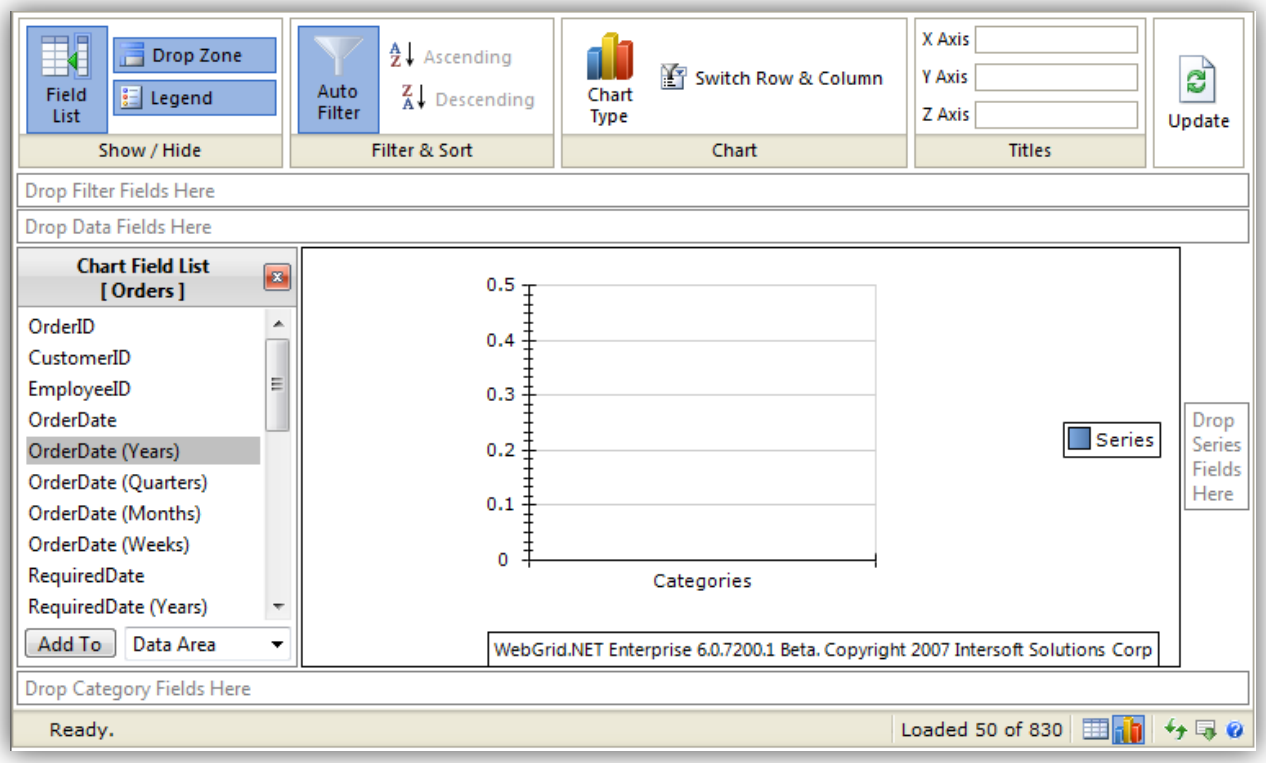
The following screenshot shows the Pivot Charting designer in WebGrid.NET Designer:



View your Webform in the browser. To see the data in Pivot Charting view, simply click the *Switch to Pivot Chart View* that appear in the status bar of the WebGrid.



Immediately after you clicked on the PivotChart View button, the *Pivot Charting Interactive UI™* will be delivered to your browser similar to the following image.



With the Charting UI ready, the data aggregation and visualization is now at your fingertips. You can start by dragging *OrderID* field to the *Data Fields* area, then *OrderDate (Years)* to the *Series Fields* area. If you have been familiar with Multi-dimensional Data concept, you can try to drop several fields to Series and Category for data analytical, as well as to change the aggregation method of the Data Fields to get the right information you wanted.

Using Designer

[This topic will be available in v6.0 RTM]

Deployment

Two additional assemblies are required to be copied to the private bin folder of your Web application in the production server. The assemblies required for Pivot Charting feature are:

- **ISNet.WebUI.WebGrid.ChartEngine.dll**. The built-in charting and visualization implementation based on aggregated data in WebGrid and ChartSettings object.
- **ISNet.Nevron.Charting.dll**. The runtime assembly of Core Charting developed by Nevron LLC which exclusively licensed for Intersoft Solutions Corporation for the use in WebGrid.NET Enterprise® 6.0.

The complete deployment guide can be found in [LINKTO: Deployment Guide] topic.

Key Concepts

WebGrid.NET Enterprise's Pivot Charting is built on the top of rock solid architecture that allows great level of extensibility, customizability and scalability. This section provides more detailed explanation of the Pivot Charting's key concepts.

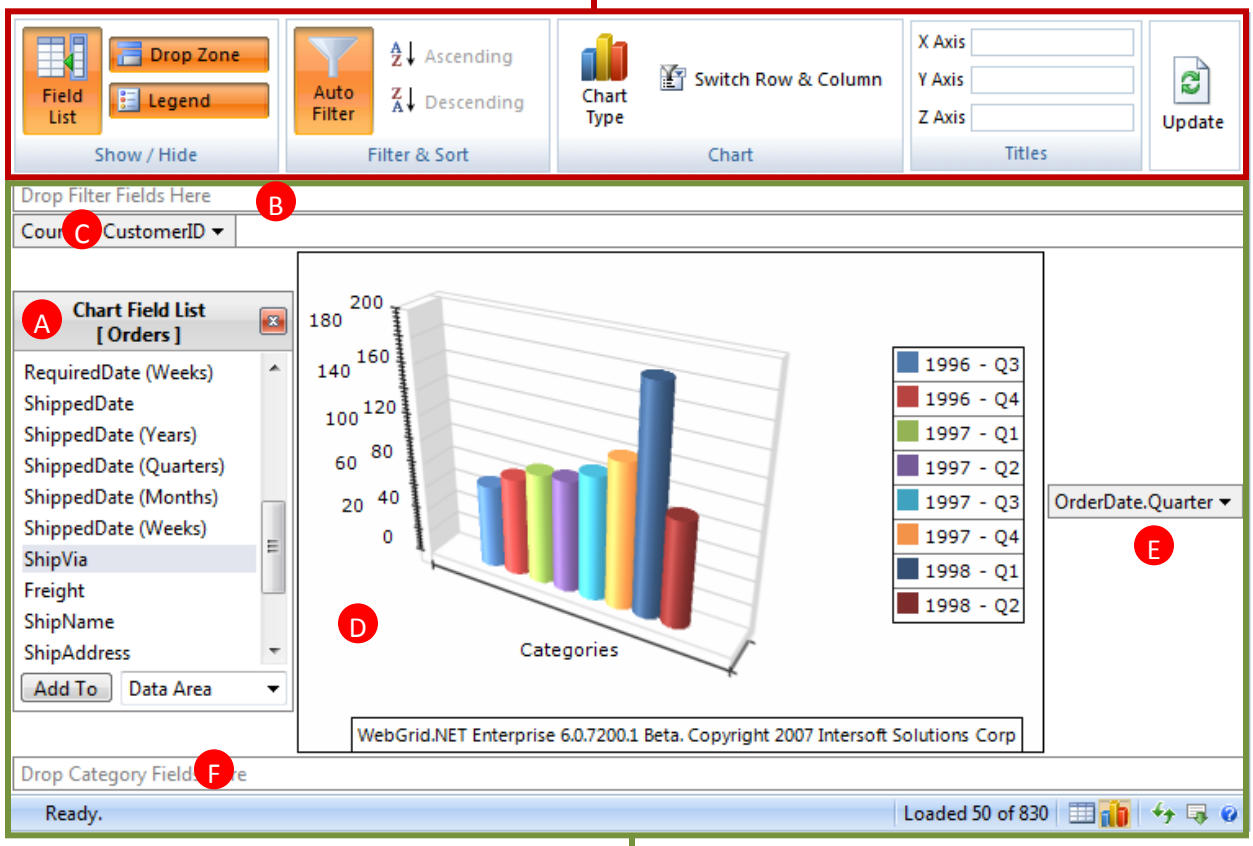
Pivot Charting User Interface

The Pivot Charting is bundled with intuitive user interface to ensure high efficiency and ease-to-use for end users. The Pivot Charting User Interface adopted Ribbon UI with two built-in themes: *Standard* and *Elegant*. The theme is determined with the Default Style mode value applied to the WebGrid control. The Default Style was firstly introduced in previous version. To learn more, please refer to Default Style.

The following screenshot describes the main User Interface of Pivot Charting.

Ribbon UI.

Contains pivot related commands.



Pivot Chart Surface.

Contains pivot related commands.

As seen in above illustration, the main Pivot Charting's User Interface consists of two main parts:

1. **Ribbon UI.**
   The Ribbon UI consists of five group commands:
   a. Show/Hide (Toggle) the visibility of interface elements. You can toggle the visibility of three main essential interface elements:
      - Field List. The container that shows a list of available fields for pivot charting.
      - Drop Zone. The drop areas for data, series and category fields.
      - Legend. The legend of data series.
   b. Filter & Sort. This group contains commands to perform auto filtering, as well as data sorting for Series and Category fields.
   c. Chart group contains commands for customizing chart types. A list of supported chart types will be shown when you clicked on *Chart Type* command.
   d. Titles group allows you to enter custom axis labels that visualized along with the Chart image.
   e. Update command. When you need to refresh the Chart data, you can easily access the Update button.

2. **Pivot Surface Area.**
   The main pivot surface area consists of six essential user interface elements such as described in the following:
   a. **Chart Field List**. Represents the fields available for pivot charting. The available fields are WebGridColumn objects which **ShowInChartFieldList** property are set to True.
   b. **Filter Drop Zone**. Represents the filtered fields that applied to the global data view.
   c. **Data Drop Zone**. Represents the data fields used by the aggregate method.
   d. **Chart Image Surface**. Represents the chart image produced by the Charting engine based on the Pivot Chart Settings. The chart image is produced in the real time by using Intersoft's proprietary AJAX technology (FlyPostBack™)
   e. **Series Zone**. Represents the data fields displayed as series.
   f. **Category Zone**. Represents the data fields displayed as category.

How-to: Hide Chart Ribbon Initially

How-to: Customize Chart Ribbon

How-to: Customize Available Chart Types

How-to: Customize Pivoting Rules

Pivot Charting View Mode

Pivot Charting has several view modes that you can use.

Allow Pivot Charting (Initial View: Grid view)

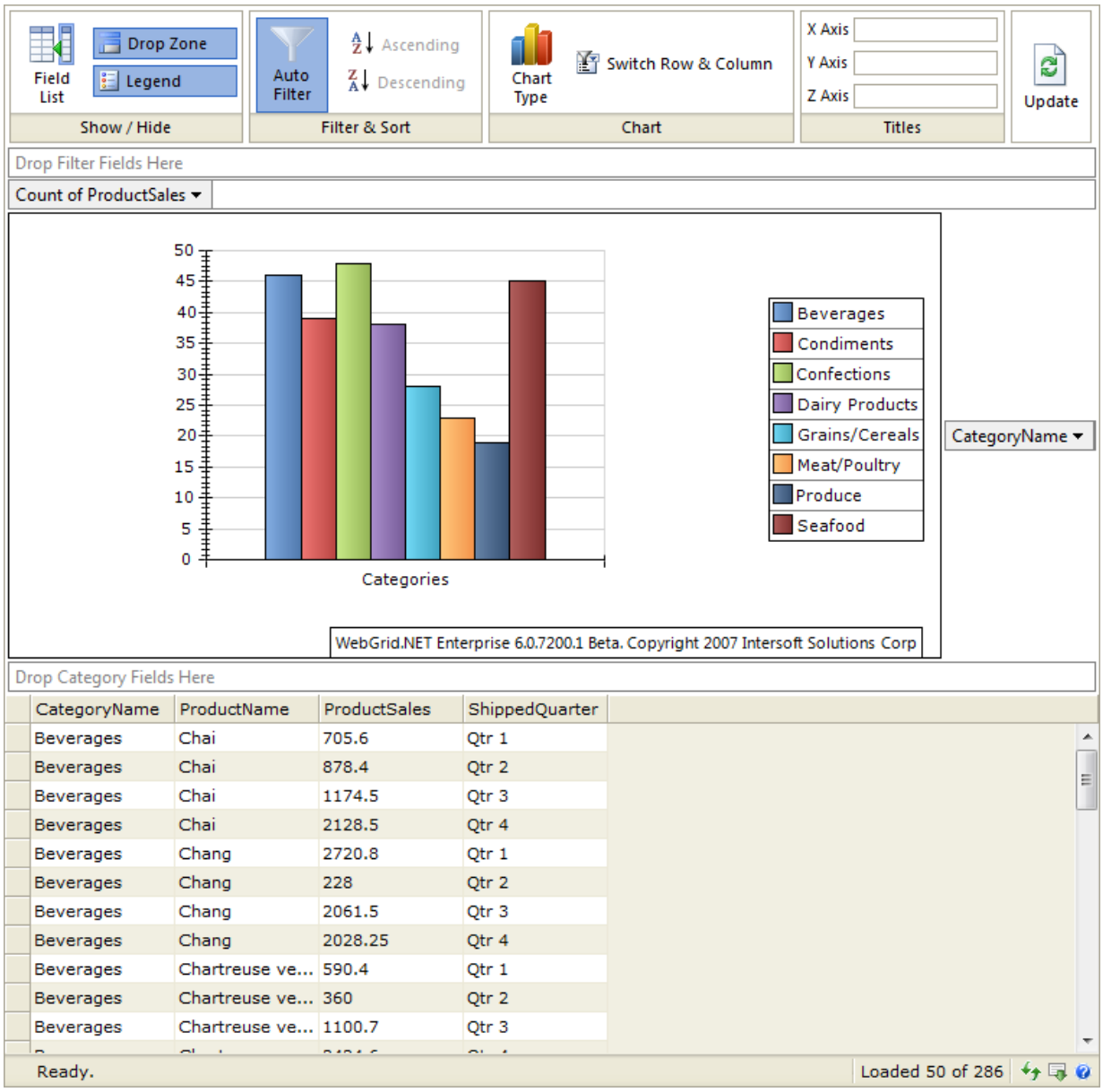
This mode show Grid view initially but the user can switch to Chart view from the PivotChart View button.

Allow Pivot Charting (Initial View: ChartView)

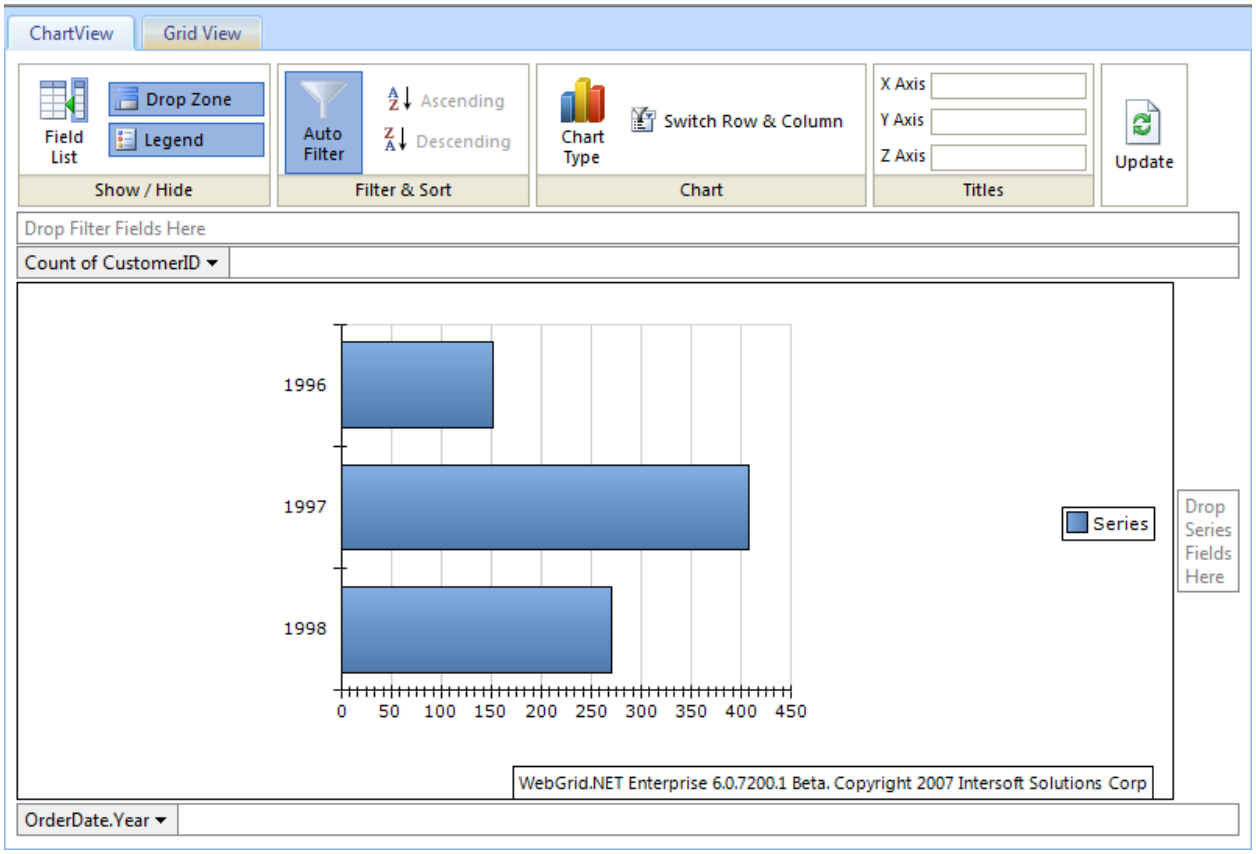
This mode is the opposite approach than the previous one. The Chart view is shown initially but the user later can switch to Grid view by clicking the Grid View button

Both View

This mode shows both Grid and Chart view. You can switch the position to Top / Bottom / Custom.



To place the chart view in custom container you need to specify the ChartCustomContainerId (e.g. ChartCustomContainerId = “ChartImage”)

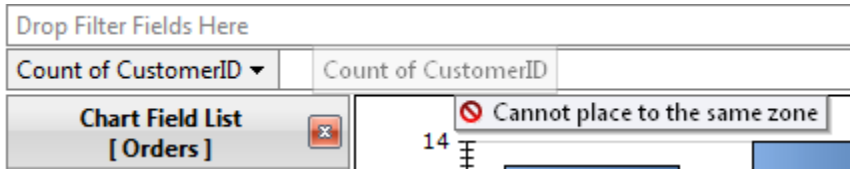


In this figure you can see that the Chart view is place under ChartView tab while the Grid view is placed under GridView tab.

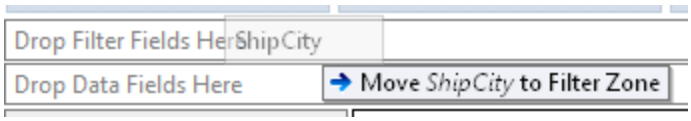
Drag and Drop

Drag and Drop is one of the Key Feature in Pivot Charting User Interface, you can conveniently drag a field from Chart Field List to any of four available drop zones. You can also drag any field element from any drop zone to another in the same way and manner as in Microsoft Access® 2007.

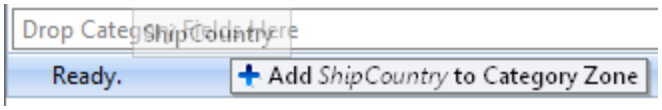
Furthermore when you drag a field element there will be an indicator whether you can drag to that particular place or not.



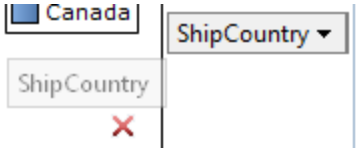
Cannot place to the same zone



Move to another zone



Add to a zone



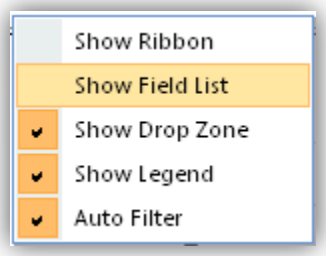
Delete a field element

Context Menu

The context menu for the Pivot Charting can be accessed by right click on anywhere in the Pivot Charting UI except the Chart Image Surface.

The context menu contains commands to toggle the visibility of user interface elements. The context menu is useful when the Ribbon UI is currently in *Hidden* state and you need to toggle the visibility of other interface elements.

The following screenshot shows the context menu of Pivot Charting.

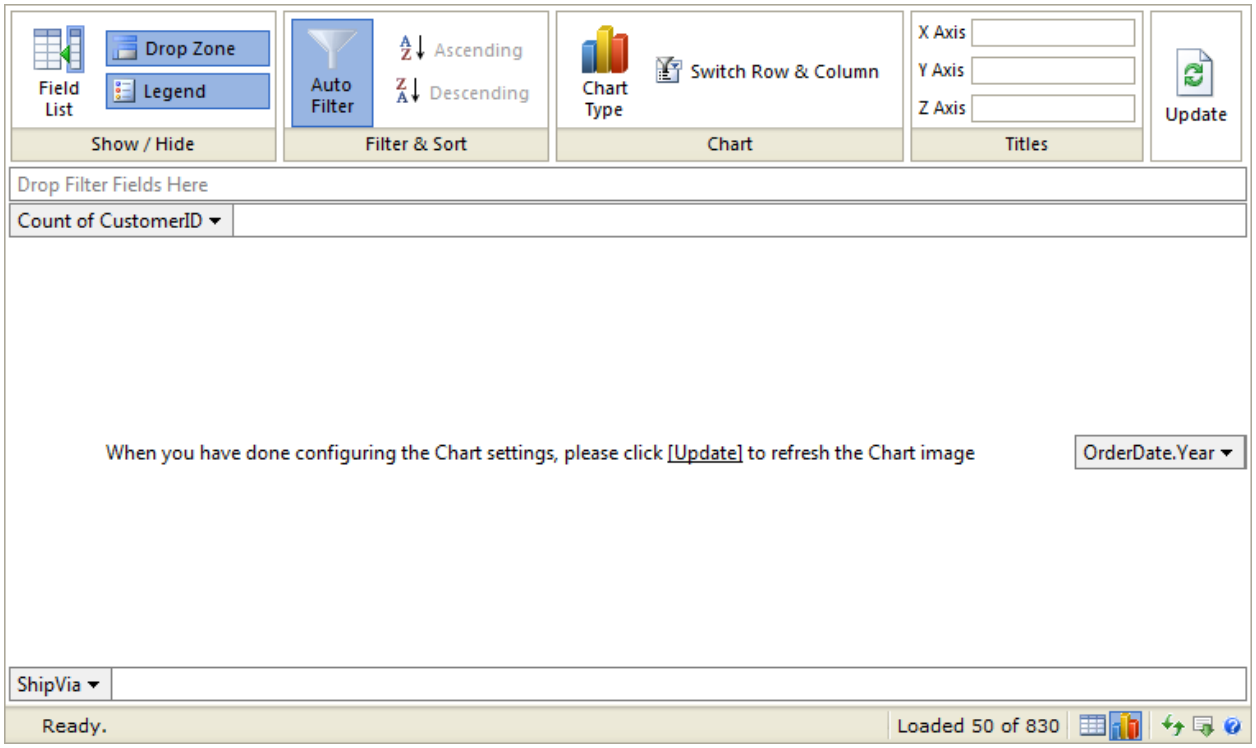


The commands in the context menu have equal function to those in Ribbon UI.

Automatic Refresh vs Batch Refresh

To show more interactivity by default when the user perform an action at Pivot Charting interface (e.g. add field element to a drop zone, show/hide legend, show/hide table field list), the Chart Image will be refreshed. So the user will get the latest chart image based on their last action.

If you prefer not to refresh the Chart image after every action, you can turn off the *ChartAutoRefresh* property. With this setting, when there's an action the ChartImage container will be changed into a message notifying the user to click update to get the ChartImage such as shown in the following screenshot.

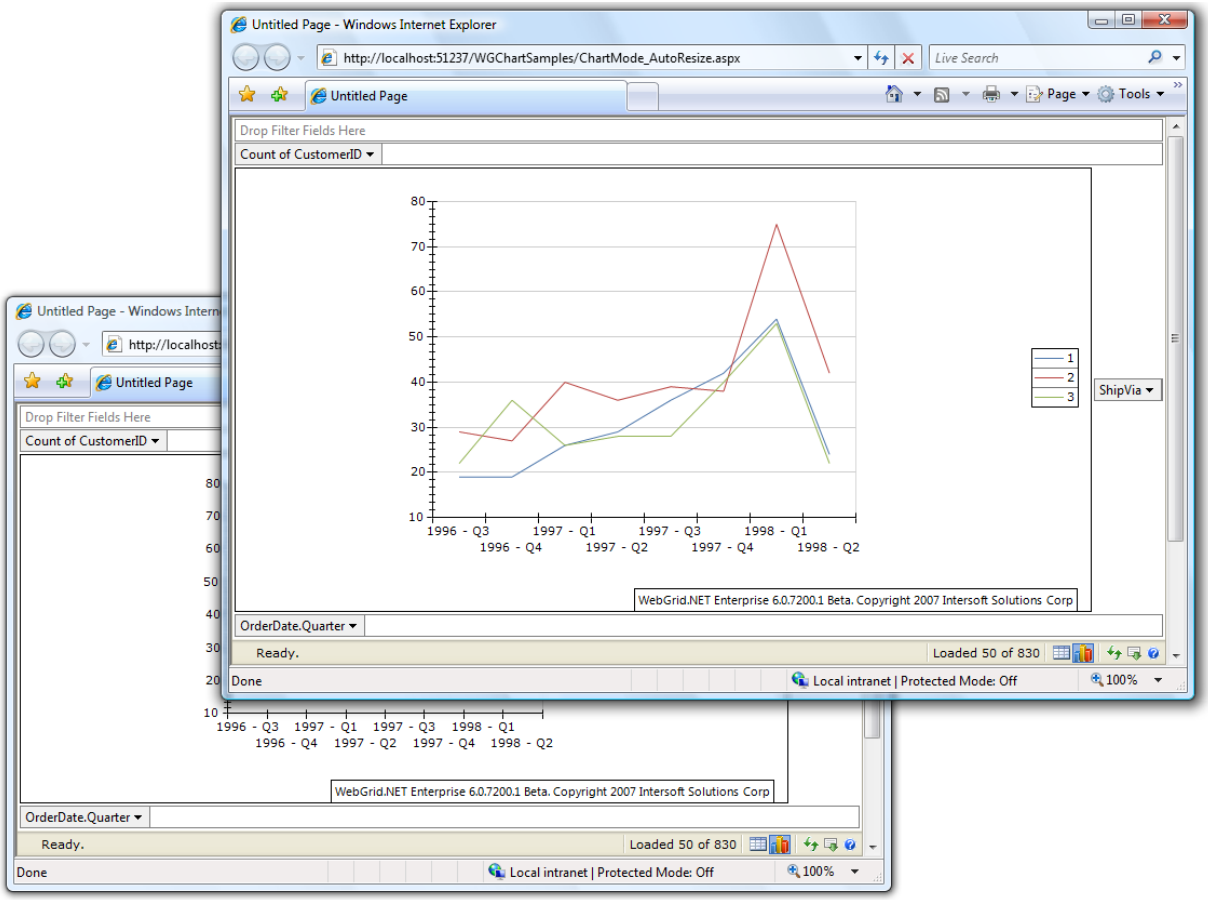


When ChartAutoRefresh is off, the changes will not be updated immediately. This enables you to perform multiple changes and update it at once.

Auto Resize Chart Image

Another interesting feature embedded in Pivot Chart User Interface is the capability to auto-resize the chart image based on browser height / width. You can do this by set the width / height of the WebGrid to 100%.

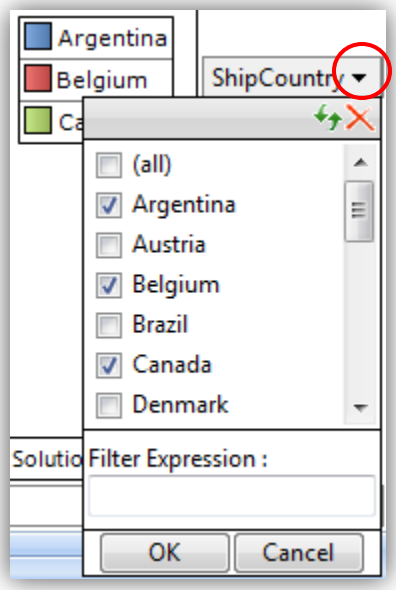
With this feature, the user who has larger screen size can benefit from it because they can see larger data visualization. On the other hand the user who has smaller screen will also benefiting cause they don't need to scroll the browser since the image will be resize to fit their browser's size.



Pivot Field Data Filtering

You can choose specific data of series or category to be included in the visualized chart image. The Pivot Charting includes built-in automatic filtering feature which can be easily accessed by clicking on the dropdown arrow of the category or series field.

The following screenshot describes the user interface for pivot filtering.



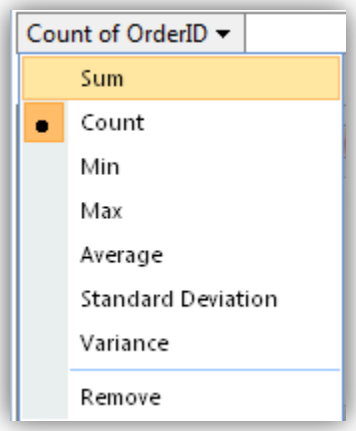
The filtering user interface allows you to check/uncheck specific items to be filtered.

Instead of checking / un-checking the data user can also specify a filter expression. When filter expression is used the checked / un-checked data will be ignored.

Note that you can easily switch off the filtering for temporary by clicking the **Auto Filter** command in the Ribbon UI. When clicked once again, it will turn on the last specified filtering settings.

Pivot Field Data Aggregation

You can customize the aggregation method of each Data Field by clicking on its dropdown arrow. The Chart will be refreshed automatically when a new selection is made.

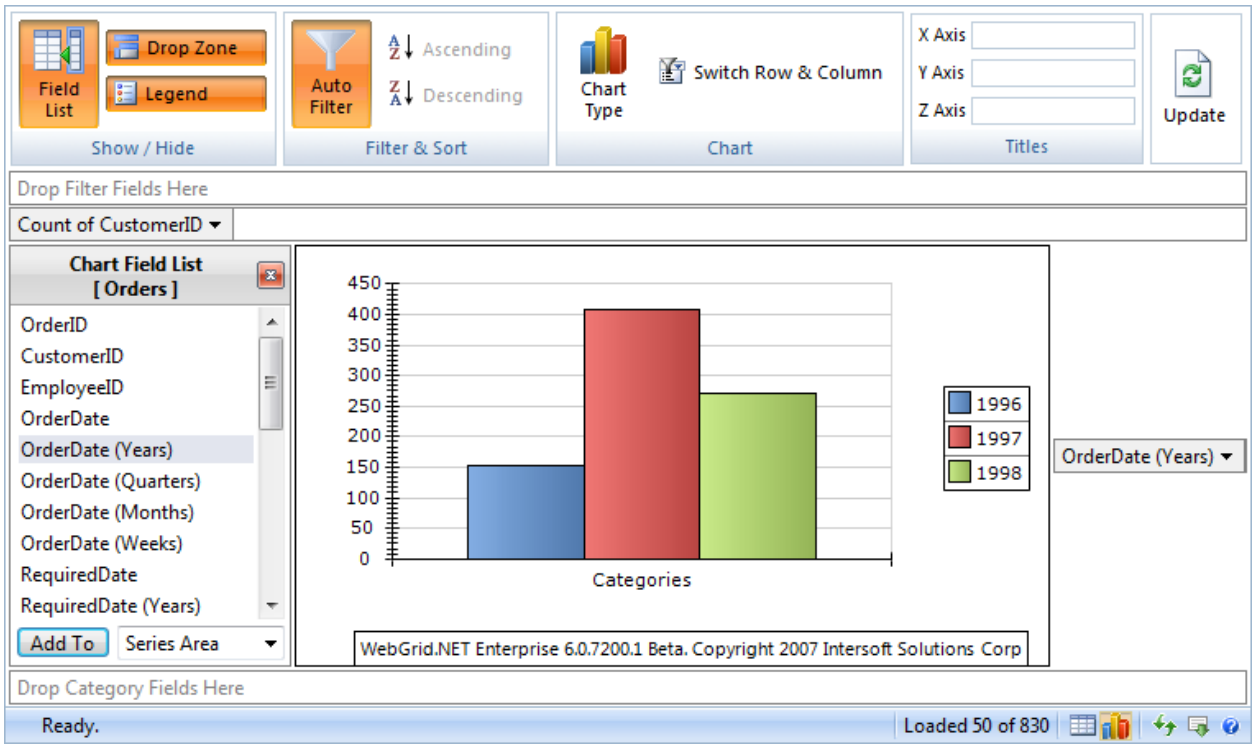


Pivot Data Processing

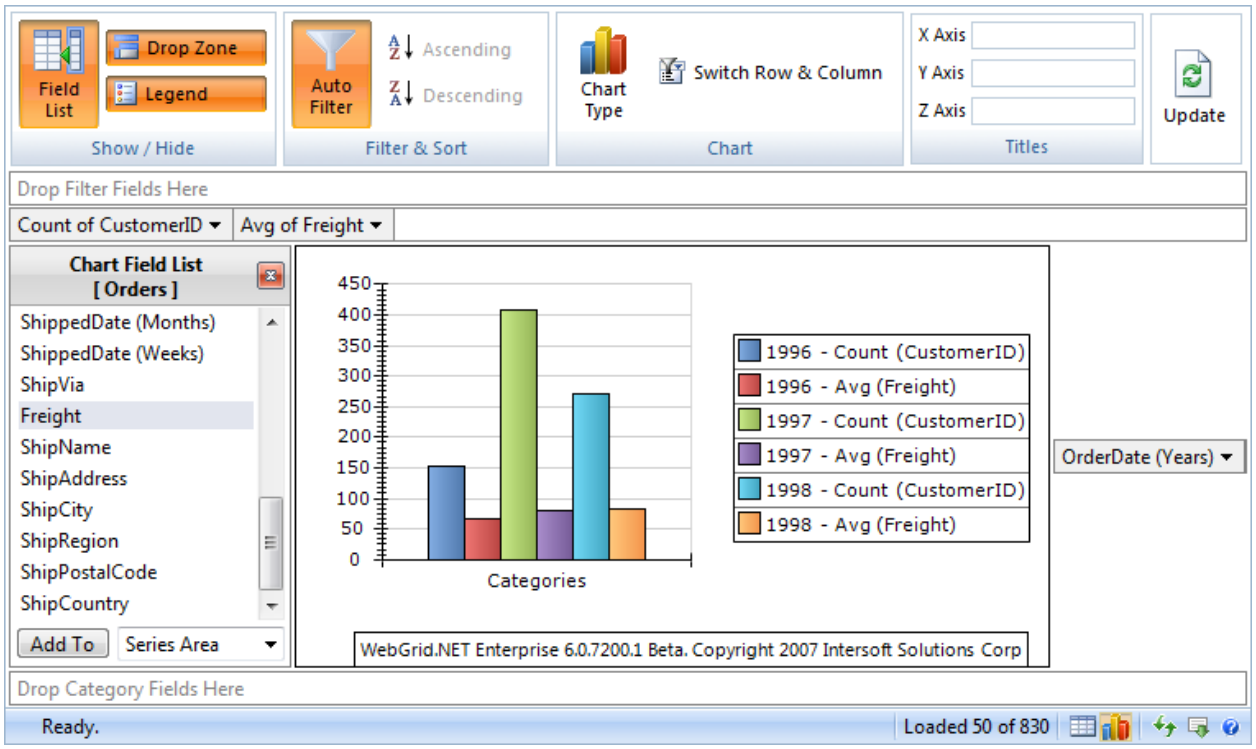
The pivot data visualization is generated by processing the collection of series fields, category fields, filter fields and data fields.

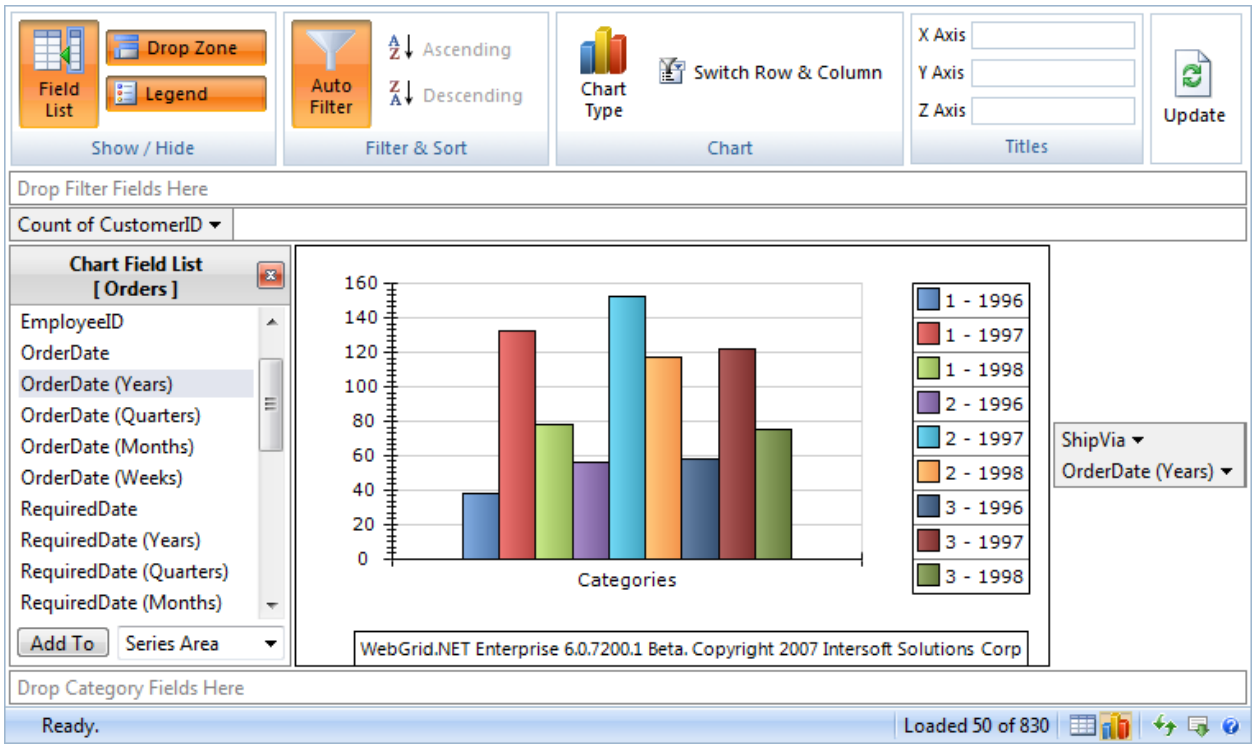
- Data fields: The Y-axis of the data space. This data field displays one of the aggregate functions (sum, count, max, min ...) of a data series.
- Series fields: The Z-axis of the data space. This data field represents a series of unique data available in the data source which appear in each data category.
- Category fields: The X-axis of the data space. The data must be categorical.
- Filter fields: The fields used to filter the data source from global level.

Data fields and series fields play parts in determining the series data that appear in each data category. If you only have one object in data fields' collection and also one object in series fields' collection, the chart engine will use the data in the series field's collection as the series data collection. (This series data collection will appear in legend if the legend is visible).

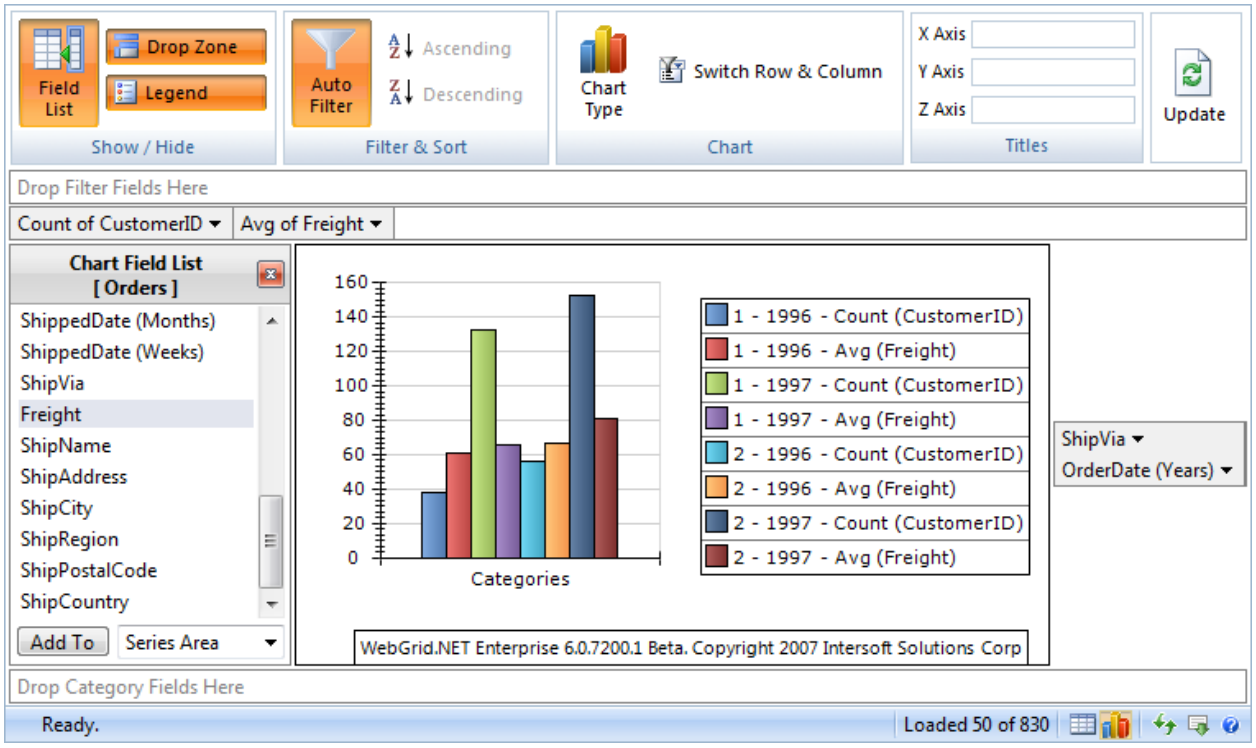


If you have more than one object in data fields' collection and/or series field's collection, the chart engine will perform a combination from both object in data fields' collection and series fields' collection for the series data. The following screenshot shows a Pivot Chart with two data fields.

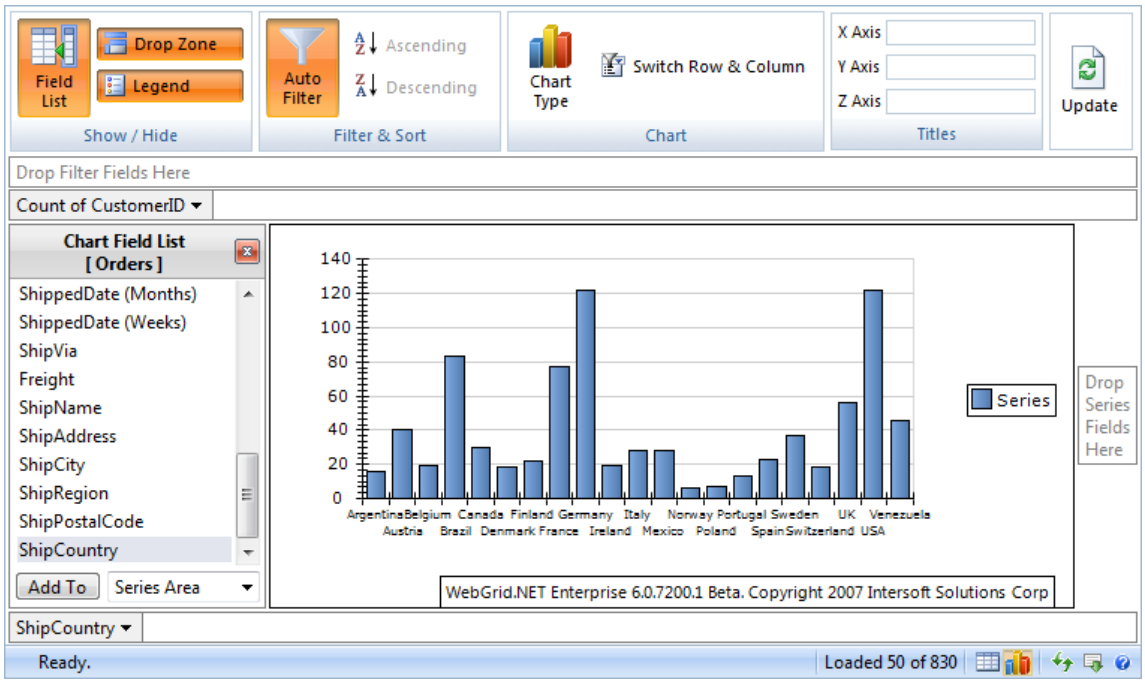




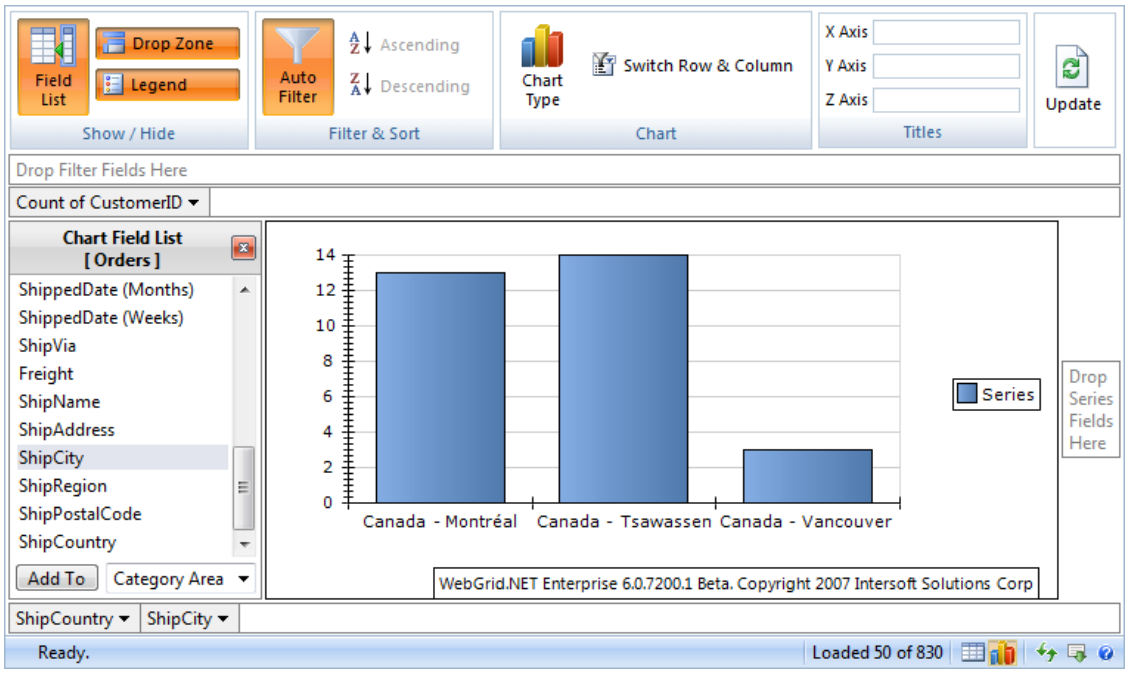
The above screenshot shows a pivot chart with two series fields, while the below one show the pivot with two series fields and two data fields.



Data category fields play parts in categorizing the series data. If you have one object in category fields' collection the chart engine will categorize the data based on that object (e.g. categorize the sales per country).



If you have more than one object in category fields' collection, the chart engine will categorize the data based on the last object. For example if we have country – city in the category collection, the data will be categorized based on city. However the label will still use country-city format. The following screenshot shows a pivot chart with two category fields.



Automatic Image Handlers

Chart Image request handler uses *HttpHandler* that need to be defined in web.config as:

```
<httpHandlers>
    <add path="ISChart.axd" verb="GET"
        type="ISNet.WebUI.WebGrid.Chart.ChartRequestHandler,
        ISNet.WebUI.WebGrid" validate="true"/>
</httpHandlers>
```

The *HttpHandler* definition above is automatically registered in your Web application when you enable Pivot Charting in WebGrid.NET Designer. To make sure you have enabled Pivot Charting in proper way, please see Getting Started with Pivot Charting.

When using the *HttpHandler* the image will be rendered directly to the client's browser during AJAX call. Note that you might want to change the request handler mode to *Centralized File Server* mode if you're using Multiple Server / Web Farm environment for better scalability.

The automatic image handler is a unique innovation in WebGrid's Charting feature which has more advantages over the traditional temporary images approach. Some of the key advantages are:

- Hassles-free development and deployment. You don't need to configure IIS virtual directory or any other folders. Just a simple XCOPY of your Web application will do.
- Better performance. Since there is no need to produce temporary images, it reduces the overhead of I/O operation and hence increasing performance.
- Better security measurement. Since there are no temporary images generated in the IIS-level directory, your Web application is more secure than ever.

Centralized Image File Server

When using centralized File Server the image will be cached in file server specified in *ImageFileServer* settings and the client will then request the image from the file server. (E.g. ImageFileServer: path=C:\Cache)

Some of benefits using File Server cache are:

- Highly scalable. You can choose a network drive to store the caches which enable multiple web servers to access the state.
- Decent performance. With more than dozen of high performance hard drives, you can choose a near memory speed with high random access and sequential read write hard drive to store the caches.
- Easily extensible. Unlike memory (RAM), you can easily extend hard drives with larger one when it becomes insufficient.

Features of File Server:

- Automatic cleaning. When the cache has expired, the caches will be automatically deleted.
- Allow customizations on cache policy and expiration duration.

To learn more about File Server cache please refers to Intersoft WebUI.NET Framework 2007

Chart Engine Interface (IChartEngine)

WebGrid.NET PivotCharting architecture provides an extensible Chart engine interface called ***IChartEngine***.

```
public interface IChartEngine
{
    Bitmap GenerateChartImage(ChartConfig chartConfig, DataTable data);
}
```

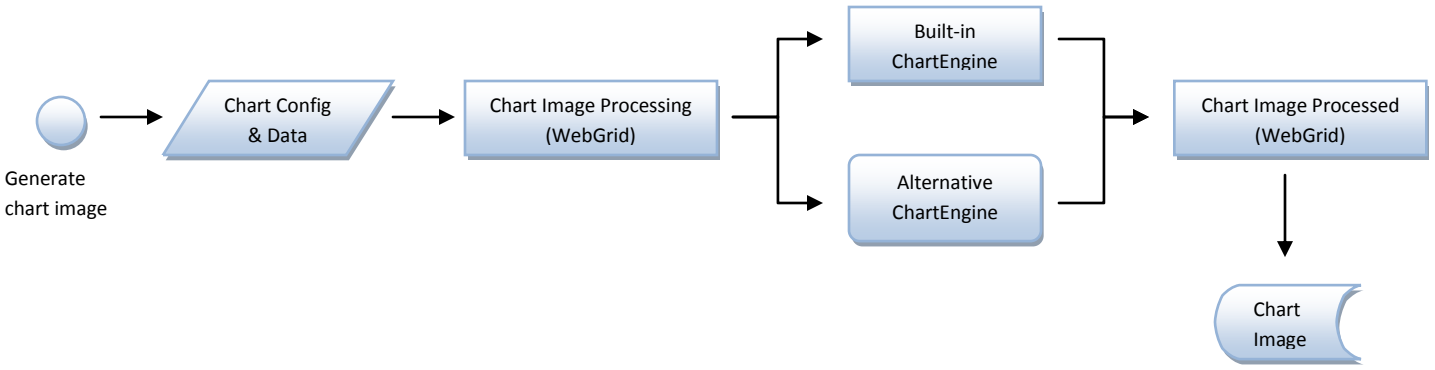
You can implement this interface and generate the chart image using your own ChartEngine. Note that the return value must be a ***Bitmap type***.

After you have implemented the interface you can attach the engine into e.AlternativeChartEngine at ChartImageProcessing event.

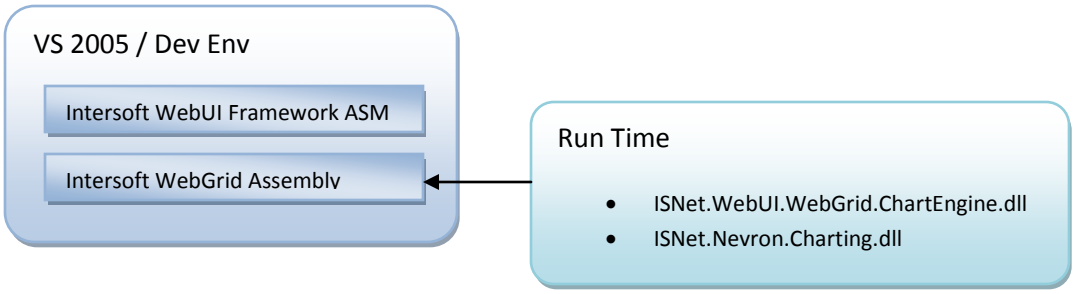
```
protected void WebGrid1_ChartImageProcessing(object sender,
    ISNet.WebUI.WebGrid.Chart.WebGridChartEventArgs e)
{
    e.AlternativeChartEngine = new AlternativeChartEngine();
}
```

By implementing this approach, WebGrid will use the specified Chart Engine to perform the chart image generation, instead of using the built in Chart Engine. Just make sure that *GenerateChartImage* method of your *AlternativeChartEngine* produce the image in *Bitmap* type.

Generating Chart image process sequence



Assembly Relations



The built-in chart engine is bundled in ISNet.WebUI.WebGrid.ChartEngine.dll. You'll need this assembly and ISNet.Nevron.Charting.dll when you use WebGrid Pivot Charting. If you decide to use your own ChartEngine (by implementing IChartEngine) you don't need to include these assemblies.

Language Localization

All textual items in Pivot Charting User Interface can be localized from WebGrid's localization settings. You can add the following tag in your localization file (which can be found in C:\Program Files\Intersoft Solutions\CommonLibrary\WebGrid\V6_0_7200\Localization) to customize the Pivot Charting UI text items based on your preference.

```
  <Charting>
    <Loading>Loading Chart...</Loading>
    <Update>When you have done configuring the Chart settings, please click
{UpdateLink}  to refresh the Chart image</Update>
    <UpdateLinkText>[Update]</UpdateLinkText>
    <DropFilterField>Drop Filter Fields Here</DropFilterField>
    <DropDataField>Drop Data Fields Here</DropDataField>
    <DropSeriesField>Drop Series Fields Here</DropSeriesField>
    <DropCategoryField>Drop Category Fields Here</DropCategoryField>
    <DropXData>Drop X Data Fields Here</DropXData>
    <DropYData>Drop Y Data Fields Here</DropYData>
    <DropBubbleData>Drop Bubble Data Fields Here</DropBubbleData>
    <DropHighData>Drop High Data Fields Here</DropHighData>
    <DropLowData>Drop Low Data Fields Here</DropLowData>
    <DropOpenData>Drop Open Data Fields Here</DropOpenData>
    <DropCloseData>Drop Close Data Fields Here</DropCloseData>
    <DropThetaData>Drop Theta Data Fields Here</DropThetaData>
    <DropRadiusData>Drop Radius Data Fields Here</DropRadiusData>
    <ShowHideRibbon>Show / Hide</ShowHideRibbon>
    <FilterSortRibbon><![CDATA[Filter & Sort]]></FilterSortRibbon>
    <ChartRibbon>Chart</ChartRibbon>
    <TitlesRibbon>Titles</TitlesRibbon>

    <FieldListButton><![CDATA[Field<br/>List]]></FieldListButton>
    <DropZoneButton>Drop Zone</DropZoneButton>
    <LegendButton>Legend</LegendButton>
    <AutoFilterButton><![CDATA[Auto<br/>Filter]]></AutoFilterButton>
    <AscendingButton>Ascending</AscendingButton>
```

```
    <DescendingButton>Descending</DescendingButton>
    <ChartTypeButton><![CDATA[Chart<br/>Type]]></ChartTypeButton>
    <SwitchRowColumnButton><![CDATA[Switch Row &
Column]]></SwitchRowColumnButton>
    <UpdateButton>Update</UpdateButton>

    <TitleXAxis>X Axis</TitleXAxis>
    <TitleYAxis>Y Axis</TitleYAxis>
    <TitleZAxis>Z Axis</TitleZAxis>

    <ErrorDuplicateData>Current PivotData already existed in PivotData
Collection</ErrorDuplicateData>
    <ErrorAddToData>Unable to add this field to Drop Data</ErrorAddToData>
    <ErrorAddToFilter>Unable to add this field to Drop
Filter</ErrorAddToFilter>
    <ErrorAddToCategory>Unable to add this field to Drop
Category</ErrorAddToCategory>
    <ErrorAddToSeries>Unable to add this field to Drop
Series</ErrorAddToSeries>
  </Charting>
```

Chart Features

WebGrid.NET Enterprise 6.0 ChartEngine which exclusively licensed from world's leading Charting provider, Nevron LLC comes with lots of fascinating features that can be utilize to produce a stunning data visualization.

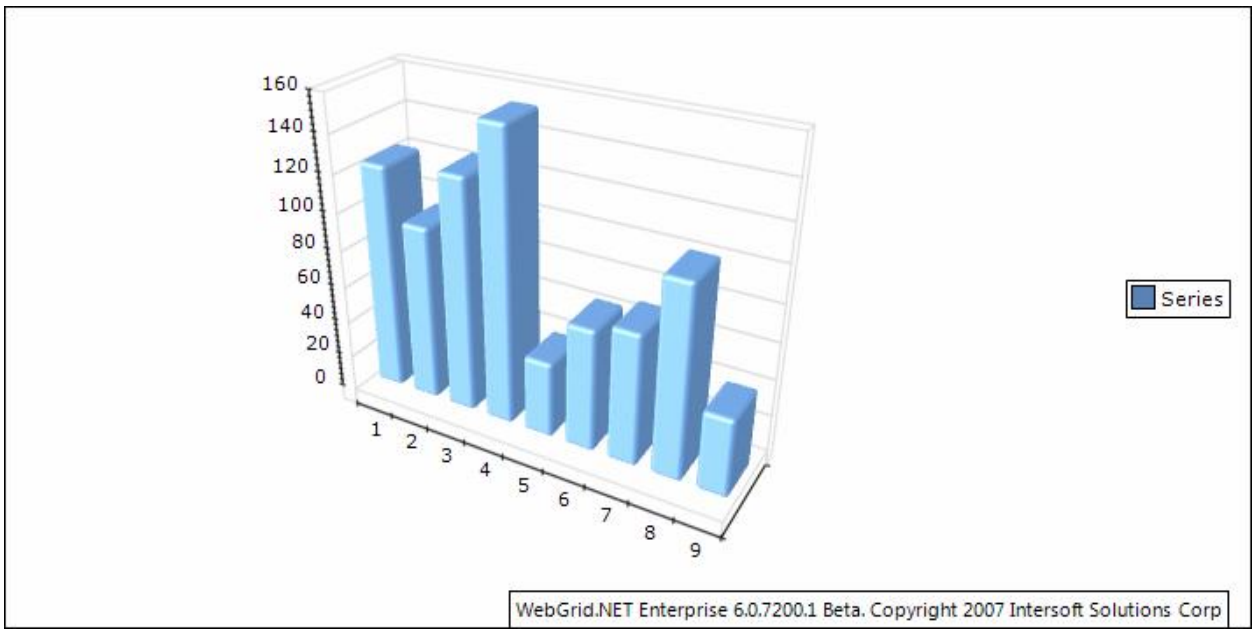
The features are categorized into several sections.

1. Series related features
2. General features
3. Lighting
4. Projection

Series related features

The features fall in this section are applied to the series object (the charting object itself). Each chart type has its own unique features that will be discussed in this section.

Series Column



Series Column has seven sub types, they are:

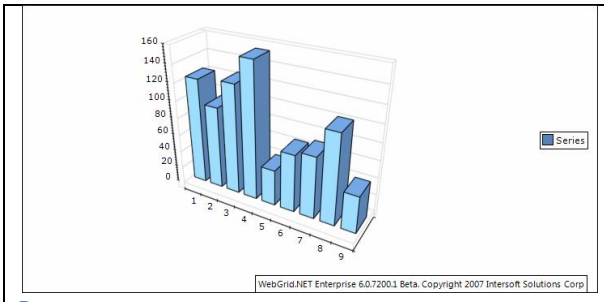
1. Clustered Column
2. Stacked Column
3. 100% Stacked Column
4. Basic 3D Column
5. Clustered 3D Column
6. 100% Stacked 3D Column

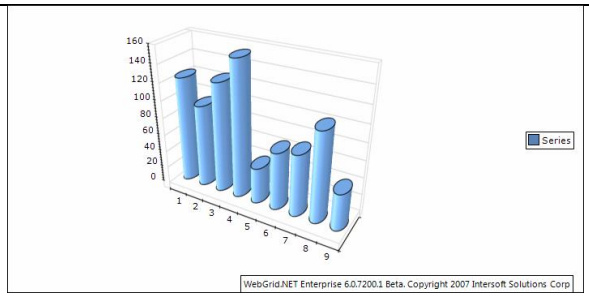
You can select which chart type to be used from:

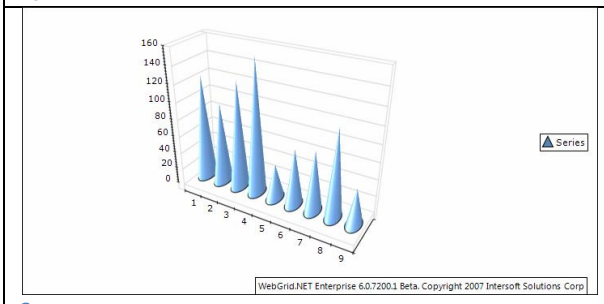
```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesColumnSettings.Type = SeriesColumnType.ClusteredColumn
```

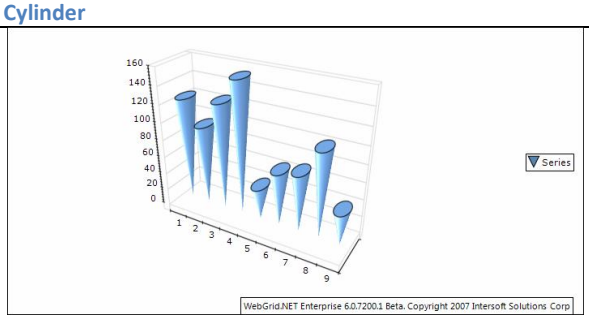
Bar Shape

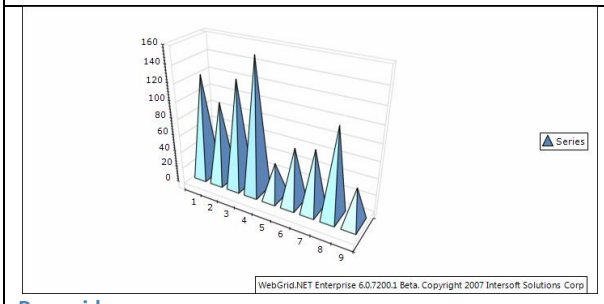
There are nine shape of bar that you can choose from.

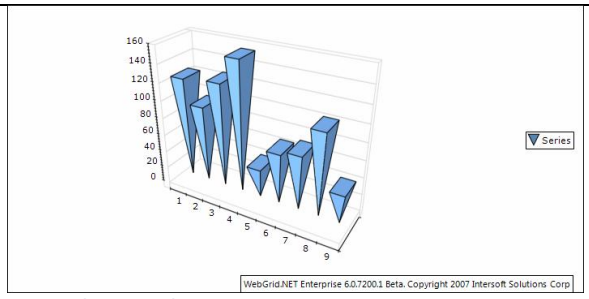
<table>
<tr>
<td> 

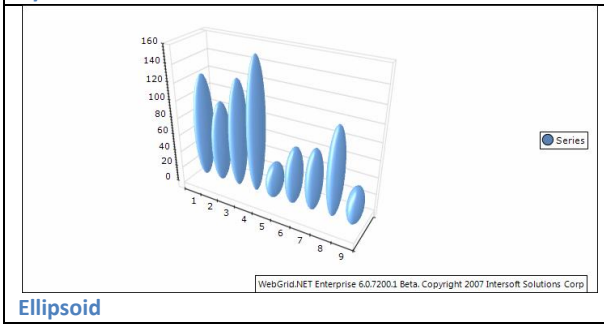
Bar</td>
<td> 

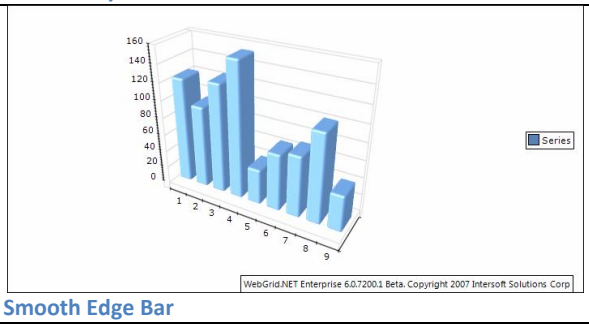
Cylinder</td>
</tr>
<tr>
<td> 

Cone</td>
<td> 

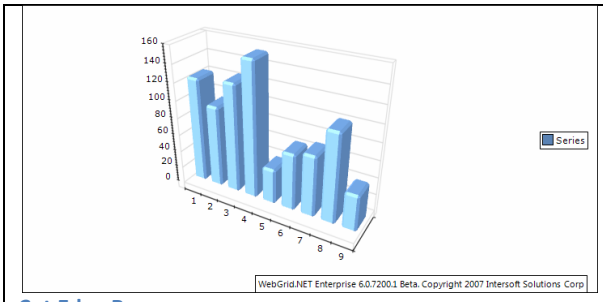
Inverted Cone</td>
</tr>
<tr>
<td> 

Pyramid</td>
<td> 

Inverted Pyramid</td>
</tr>
<tr>
<td> 

Ellipsoid</td>
<td> 

Smooth Edge Bar</td>
</tr>
</table>



Cut Edge Bar

The shape of the bars is controlled from BarShape property. It accepts values from BarShape enumeration.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesColumnSettings.Settings;
settings.BarShape = BarShape.SmoothEdgeBar;
```

When the bars are displayed as SmoothEdgeBar or CutEdgeBar you can control whether top and bottom edges of the bar are displayed in the respective manner with the help of the HasTopEdge and HasBottomEdge properties.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesColumnSettings.Settings;
settings.HasBottomEdge = true;
settings.HasTopEdge = true;
```

The size of the edge is controlled in percents of the smaller width or depth bar dimension. By default it is set to 15. The following code will make the smooth bar edges twice bigger.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesColumnSettings.Settings;
Settings.BarEdgePercent = 30;
```

Origin

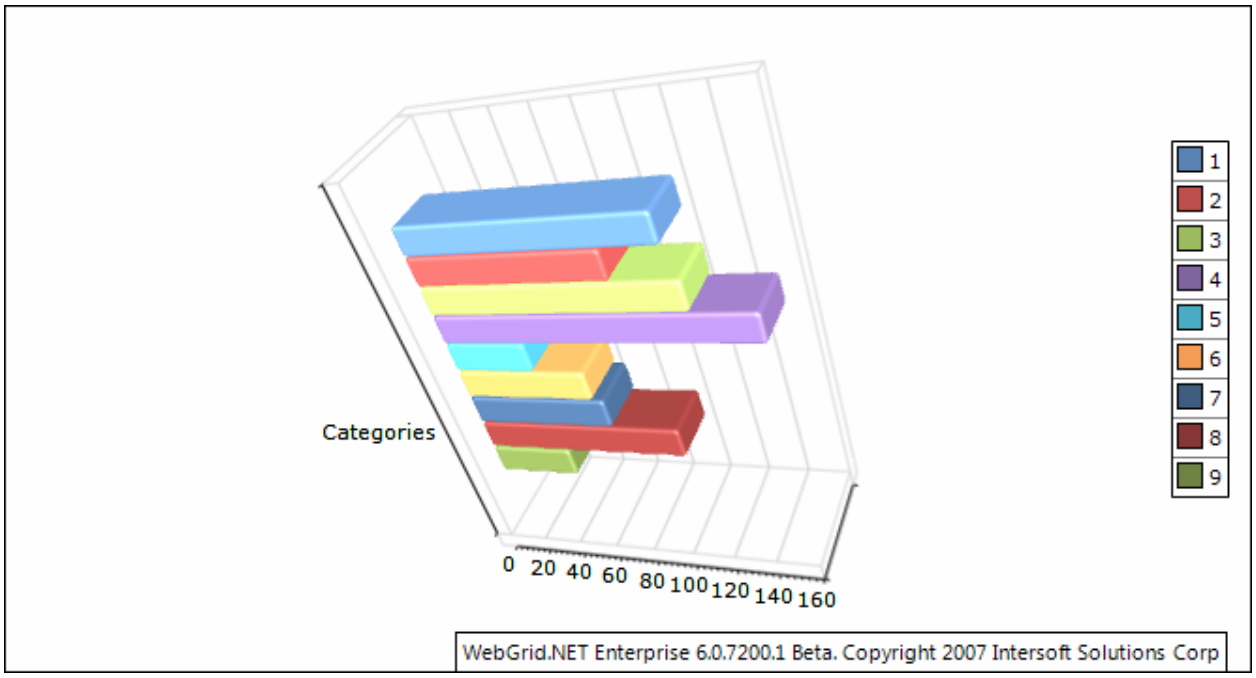
By default all bars are using 0 as origin value (all bars begin from 0). You can turn off this behavior by setting the UseOrigin property to false.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesColumnSettings.Settings;
settings.UseOrigin = false;
```

If the bars do not use an origin value they will all begin from the minimum bar value (the minimal bar will be displayed with zero height). In some cases you may want the bars to begin from a certain value – for example 15. The following code will achive the resilt:

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesColumnSettings.Settings;
settings.UseOrigin = false;
settings.Origin = 15;
```

Series Bar



Series Bar has seven sub types, they are:

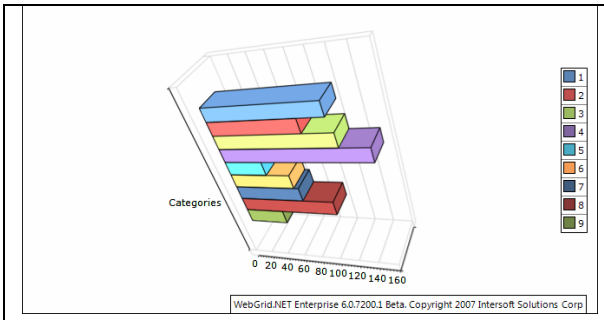
1. Clustered Bar
2. Stacked Bar
3. 100% Stacked Bar
4. Basic 3D Bar
5. Clustered 3D Bar
6. 100% Stacked 3D Bar

You can select which chart type to be used from:

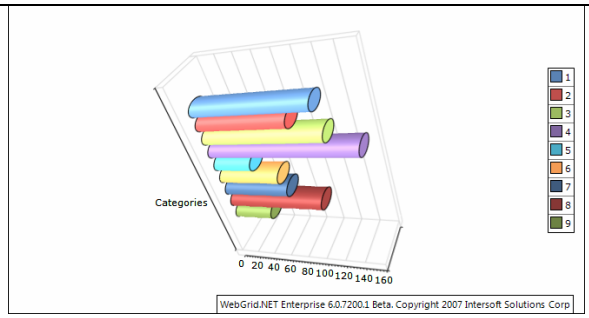
```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesBarSettings.Type = SeriesBarType.ClusteredBar;
```

Bar Shape

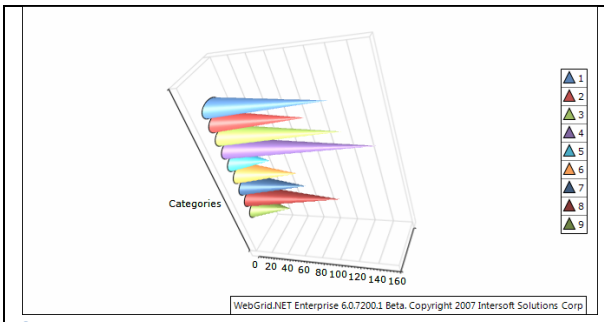
There are nine shape of bar that you can choose from.



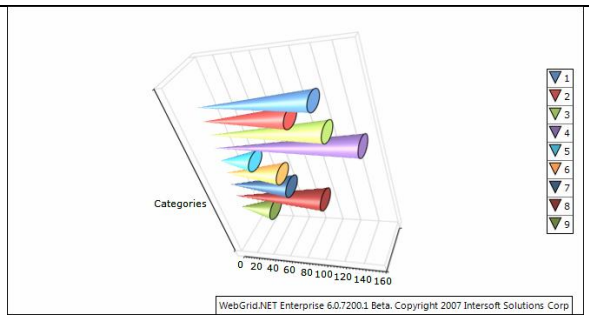
Bar



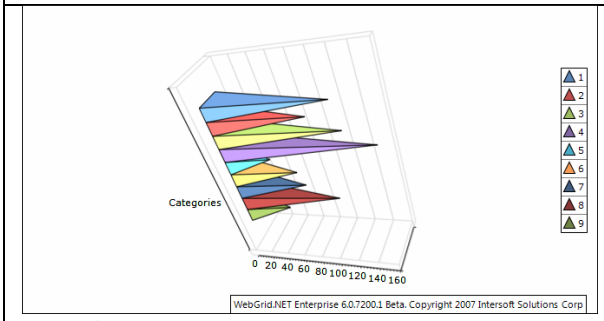
Cylinder



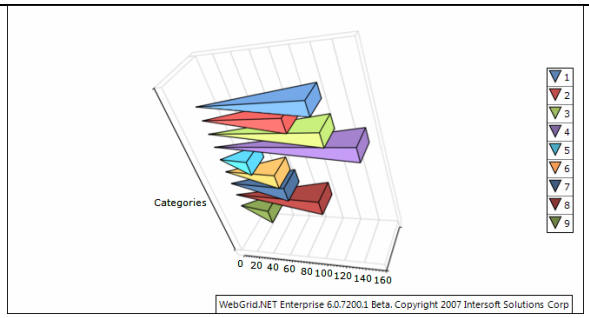
Cone



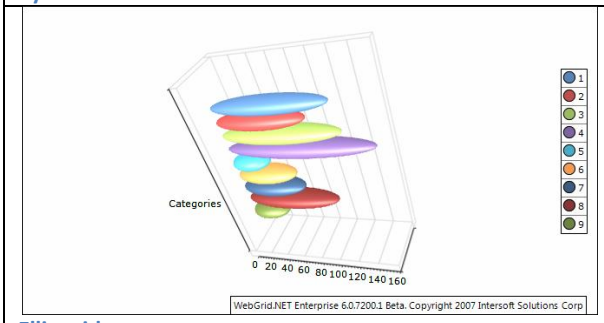
Inverted Cone



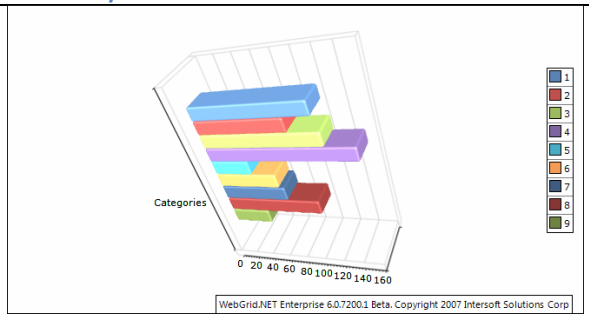
Pyramid



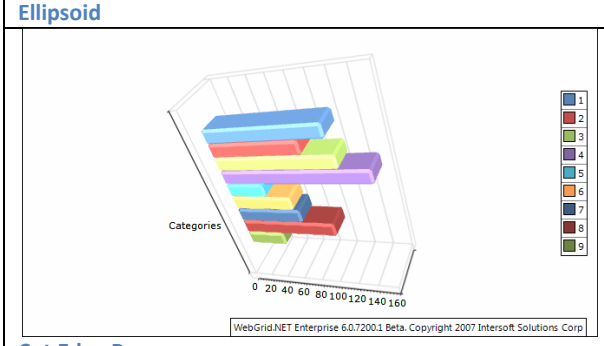
Inverted Pyramid



Ellipsoid



Smooth Edge Bar



Cut Edge Bar

The shape of the bars is controlled from BarShape property. It accepts values from BarShape enumeration.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesBarSettings.Settings;
settings.BarShape = BarShape.SmoothEdgeBar;
```

When the bars are displayed as SmoothEdgeBar or CutEdgeBar you can control whether top and bottom edges of the bar are displayed in the respective manner with the help of the HasTopEdge and HasBottomEdge properties.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesBarSettings.Settings;
settings.HasBottomEdge = true;
settings.HasTopEdge = true;
```

The size of the edge is controlled in percents of the smaller width or depth bar dimension. By default it is set to 15. The following code will make the smooth bar edges twice bigger.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesBarSettings.Settings;
Settings.BarEdgePercent = 30;
```

Origin

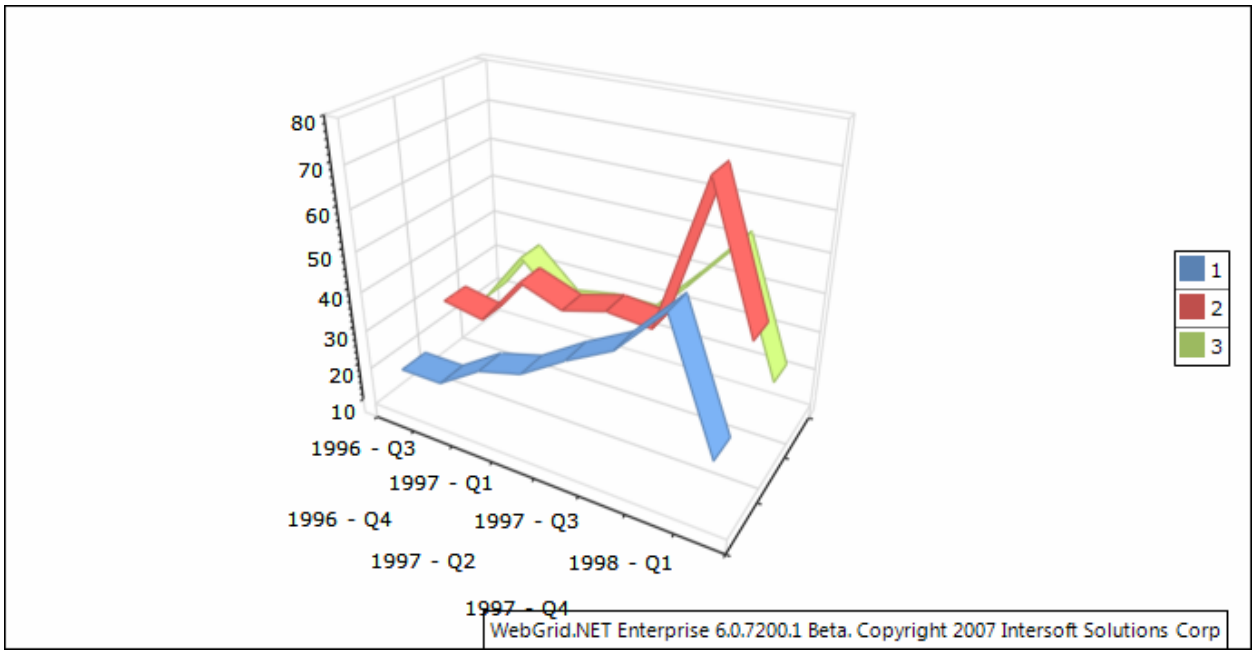
By default all bars are using 0 as origin value (all bars begin from 0). You can turn off this behavior by setting the UseOrigin property to false.

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesBarSettings.Settings;
settings.UseOrigin = false;
```

If the bars do not use an origin value they will all begin from the minimum bar value (the minimal bar will be displayed with zero height). In some cases you may want the bars to begin from a certain value – for example 15. The following code will achive the resilt:

```
SeriesBar settings = WebGrid1.ChartSettings.SeriesBarSettings.Settings;
settings.UseOrigin = false;
settings.Origin = 15;
```

Series Line



Series Line has fourteen sub types, they are:

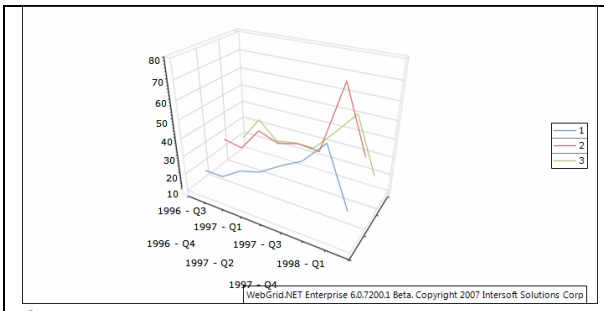
1. Overlapped Line
2. Overlapped Line With Markers
3. Stacked Line
4. Stacked Line With Markers
5. 100% Stacked Line
6. 100% Stacked Line With Markers
7. Basic 3D Line
8. Basic 3D Line With Markers
9. Overlapped 3D Line
10. Overlapped 3D Line With Markers
11. Stacked 3D Line
12. Stacked 3D Line With Markers
13. 100% Stacked 3D Line
14. 100% Stacked 3D Line With Markers

You can select which chart type to be used from:

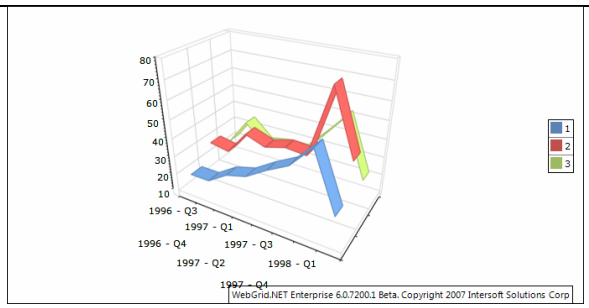
```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesLineSettings.Type = SeriesLineType.OverlappedLine;
```

Line Shape

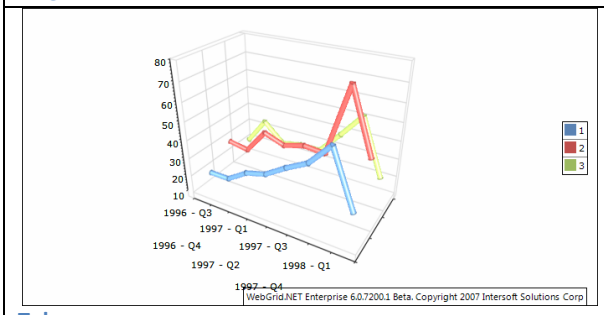
There are four types of shape that you can choose from



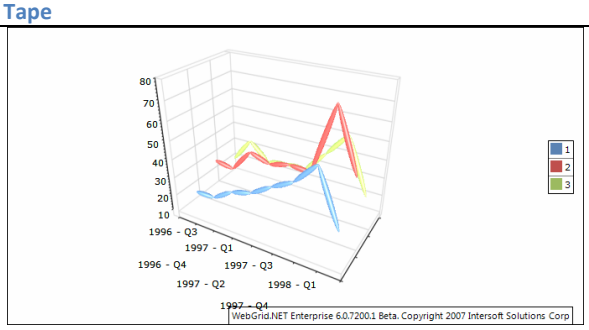
Line



Tape



Tube

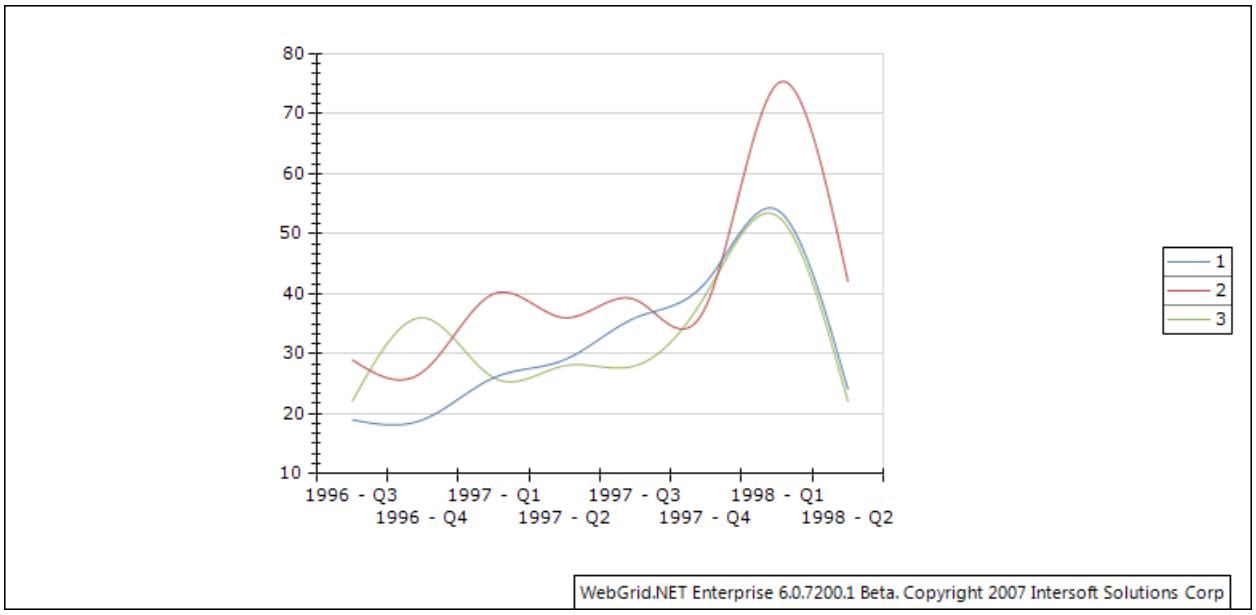


Ellipsoid

The line segments can be visualized with different shapes. You can control the shape with the help of the LineSegmentShape property which accepts values from the LineSegmentShape enumeration. By default the series is displayed as simple line. The following example will display the series as tape:

```
SeriesLine settings = WebGrid1.ChartSettings.SeriesLineSettings.Settings;
settings.LineSegmentShape = LineSegmentShape.Tape;
```

Series Smooth Line



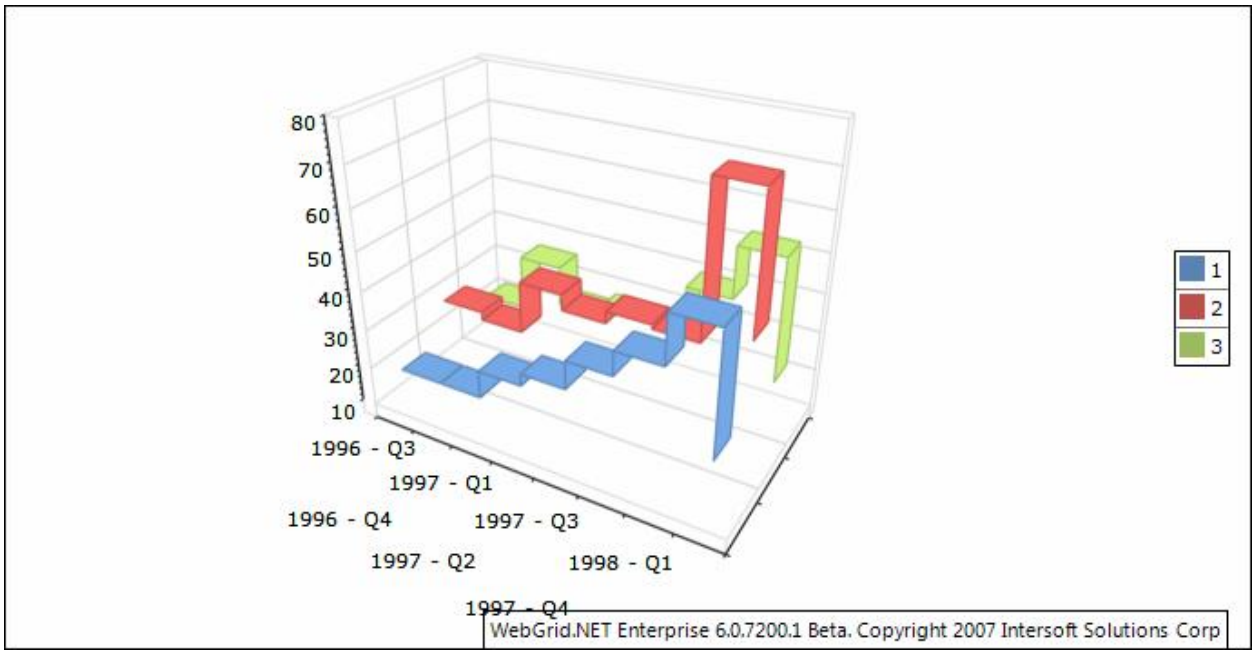
Series Line has two sub types, they are:

1. Overlapped Smooth Line
2. Overlapped Smooth Line With Markers

You can select which chart type to be used from:

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesSmoothLineSettings.Type                                    =
SeriesSmoothLineType.OverlappedSmoothLine;
```

Series Step Line



Series Step Line has four sub types, they are:

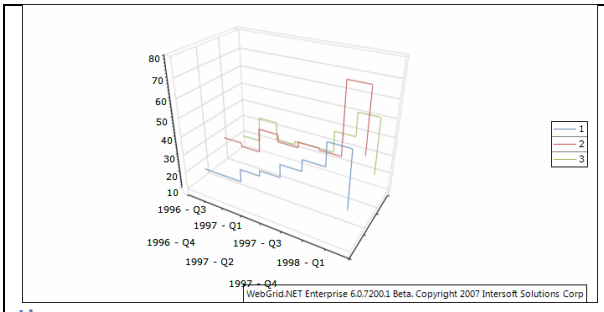
1. Overlapped Step Line
2. Overlapped Step Line With Markers
3. Basic 3D Step Line
4. Basic 3D Step Line With Markers

You can select which chart type to be used from:

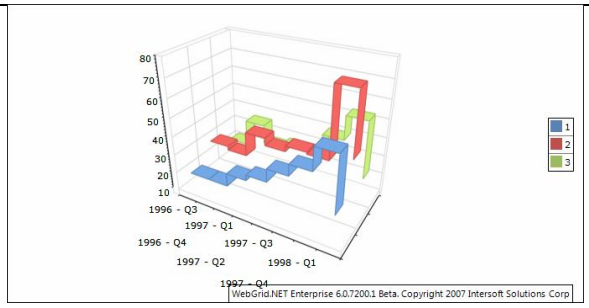
```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesStepLineSettings.Type                                    =
SeriesStepLineType.OverlappedStepLine;
```

Line Shape

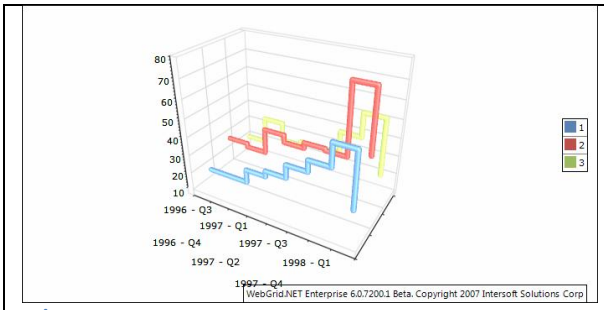
There are four types of shape that you can choose from



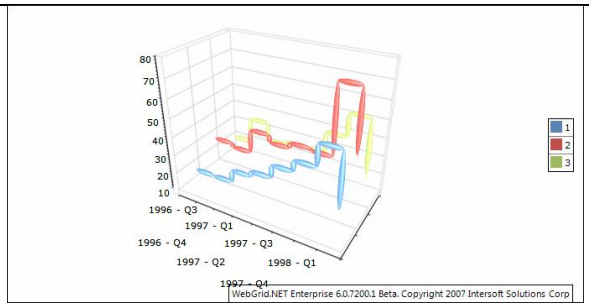
Line



Tape



Tube

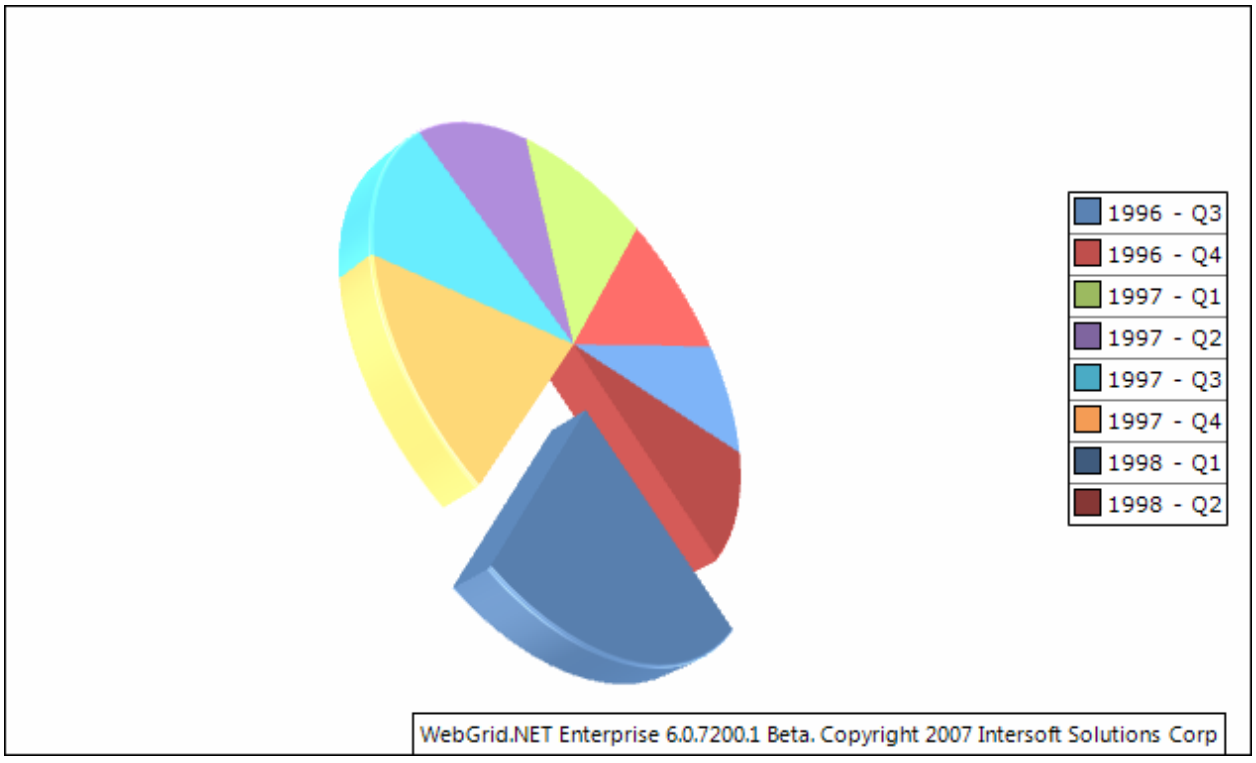


Ellipsoid

The line segments can be visualized with different shapes. You can control the shape with the help of the LineSegmentShape property which accepts values from the LineSegmentShape enumeration. By default the series is displayed as simple line. The following example will display the series as tape:

```
SeriesStepLine settings =
WebGrid1.ChartSettings.SeriesStepLineSettings.Settings;
settings.LineSegmentShape = LineSegmentShape.Tape;
```

Series Pie



Series Pie has four sub types:

1. Pie
2. Exploded Pie

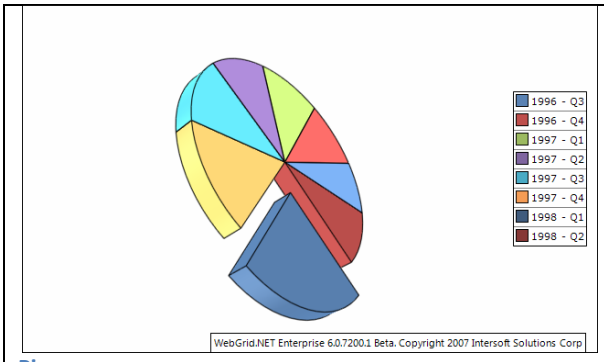
3. 3D Pie
4. 3D Exploded Pie

You can select which chart type to be used from:

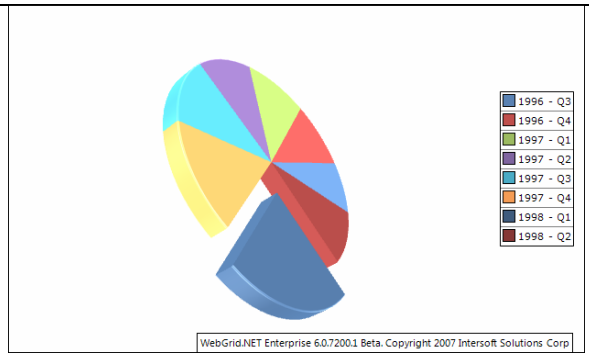
```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesPieSettings.Type = SeriesPieType.Pie;
```

Pie Style

There are two types of pie that you can choose from



Pie



Smooth Edge Pie

The shape of the pie segments can be controlled through the PieStyle property. It accepts values from the PieStyle enumeration. For example the following code will display a smooth edge pie chart:

```
settings.PieStyle = PieStyle.Pie;
```

Explosion

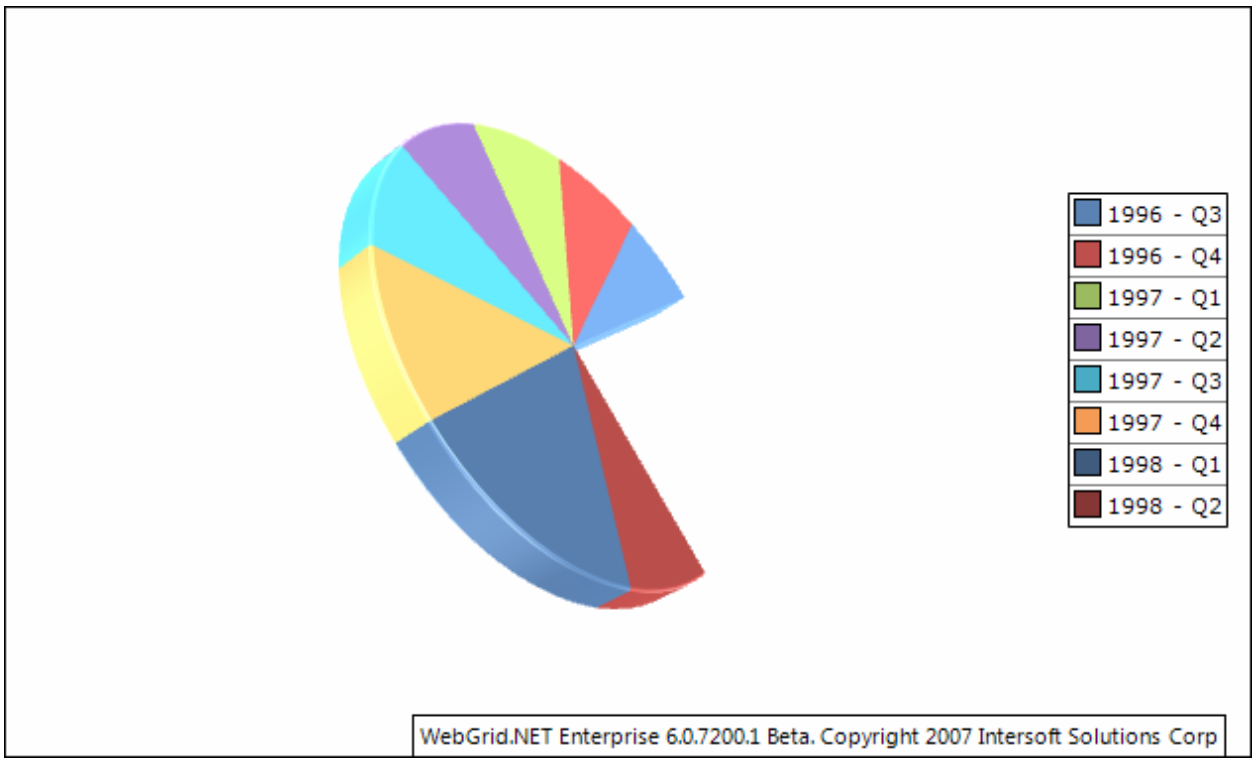
You can explode the data point of the pie chart using:

```
SeriesPie settings = WebGrid1.ChartSettings.SeriesPieSettings.Settings;
settings.ExplodeMode = PieExplodeMode.Biggest;
settings.Explosion = 8;
```

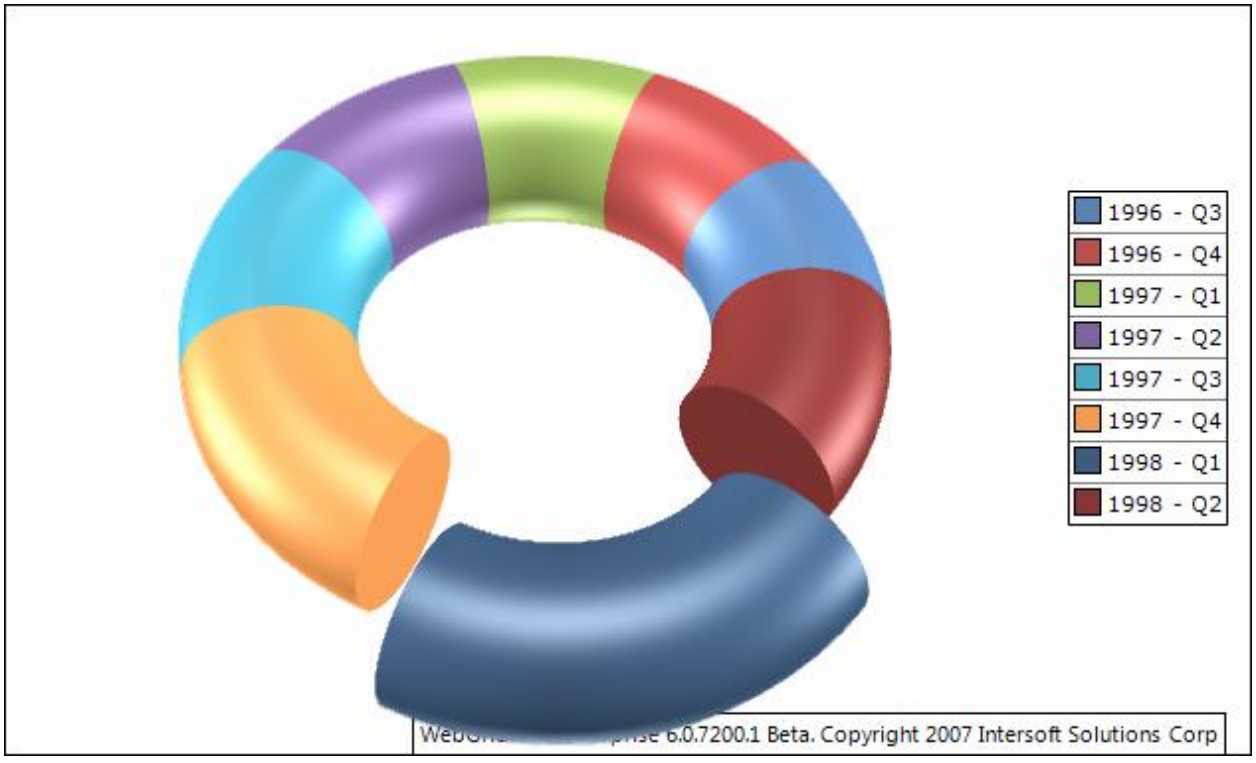
Miscellaneous

The BeginAngle property specifies the initial rotation of the first item. The TotalAngle property controls the total angle (in degrees) in which pies is displayed. By default this property is 360 degrees. The following code will change the begin angle of the pies to 30 degrees and set the total angle to 180 degrees.

```
SeriesPie settings = WebGrid1.ChartSettings.SeriesPieSettings.Settings;
settings.BeginAngle = 45;
settings.TotalAngle = 270;
```



Series Doughnut



Series Doughnut has four sub types:

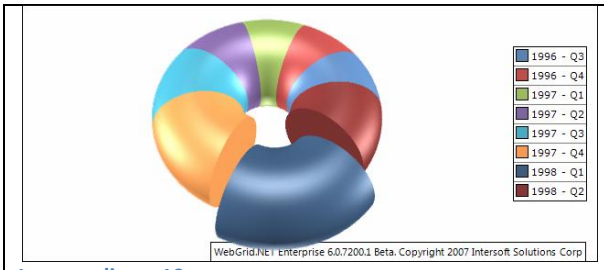
1. Doughnut
2. Exploded Doughnut
3. 3D Doughnut
4. Exploded 3D Doughnut

You can select which chart type to be used from:

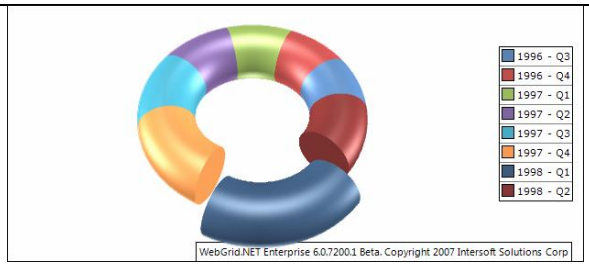
```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesDoughnutSettings.Type = SeriesDoughnutType.Doughnut;
```

InnerRadius

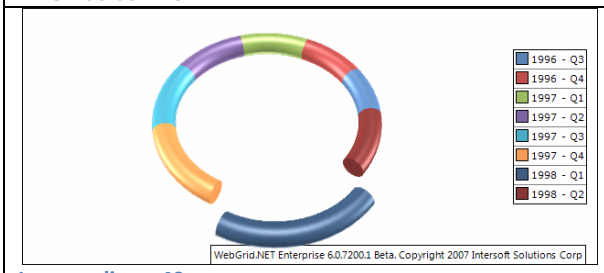
Inner radius controls the length of the inner radius (the doughnut hole), the larger the value the thinner the doughnut will be and vice versa. The maximum value is 50 which mean the doughnut is not visible.



Inner radius = 10



Inner radius = 25



Inner radius = 40

Explosion

You can explode the data point of the pie chart using:

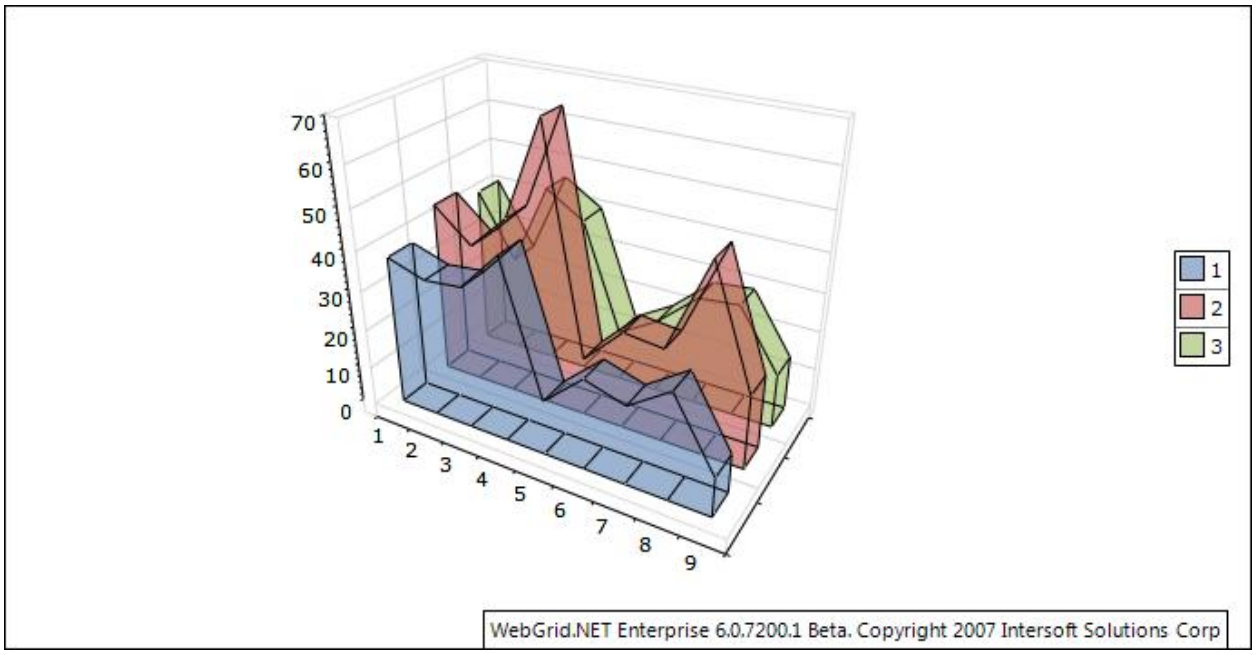
```
SeriesPie settings = WebGrid1.ChartSettings.SeriesDoughnutSettings.Settings;
settings.ExplodeMode = PieExplodeMode.Biggest;
settings.Explosion = 8;
```

Miscellaneous

The BeginAngle property specifies the initial rotation of the first item. The TotalAngle property controls the total angle (in degrees) in which pies is displayed. By default this property is 360 degrees. The following code will change the begin angle of the pies to 30 degrees and set the total angle to 180 degrees.

```
SeriesPie settings = WebGrid1.ChartSettings. SeriesDoughnutSettings.Settings;
settings.BeginAngle = 45;
settings.TotalAngle = 270;
```

Series Area



Series Area has five sub types:

1. Stacked Area
2. 100% Stacked Area
3. Basic 3D Area
4. Stacked 3D Area
5. 100% Stacked 3D Area

You can select which chart type to be used from:

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.SeriesAreaSettings.Type = SeriesAreaType.Area
```

Origin

By default the area uses 0 as origin (the area begines from 0). You can turn off this behavior by setting the UseOrigin property fo false.

```
SeriesArea settings = WebGrid1.ChartSettings.SeriesAreaSettings.Settings;
settings.UseOrigin = false;
```

If the area is not using an origin value it will begin from the minimum area value (the minimal area value will be displayed with zero height). In some cases you may want the area to begin from a certain value – for example 26. The following code demonstrate how to achive this result;

```
SeriesArea settings = WebGrid1.ChartSettings.SeriesAreaSettings.Settings;
settings.UseOrigin = false;
settings.Origin = 26;
```

Other Series related features

Except features that related to corresponding chart type there also features that applied generally to entire series, they are:

1. Series Legend
2. Series Data Point

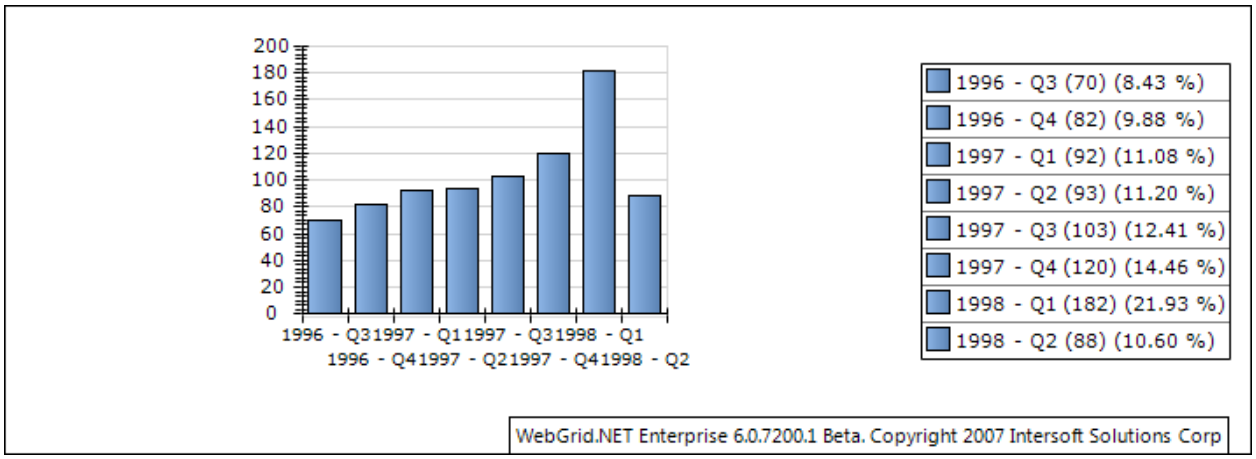
Series Legend

This feature basically used to control the formatting of the legend text format.

There are two types of legend mode that you can select from (Series or Data Points).

In Series mode the legend will be set from the Chart Engine, if set to Data Points the legend will be determine with the given format.

```
SeriesLegend seriesLegend = settings.SeriesLegend;
seriesLegend.Format = "<label> (<value>) (<percent>)";
```

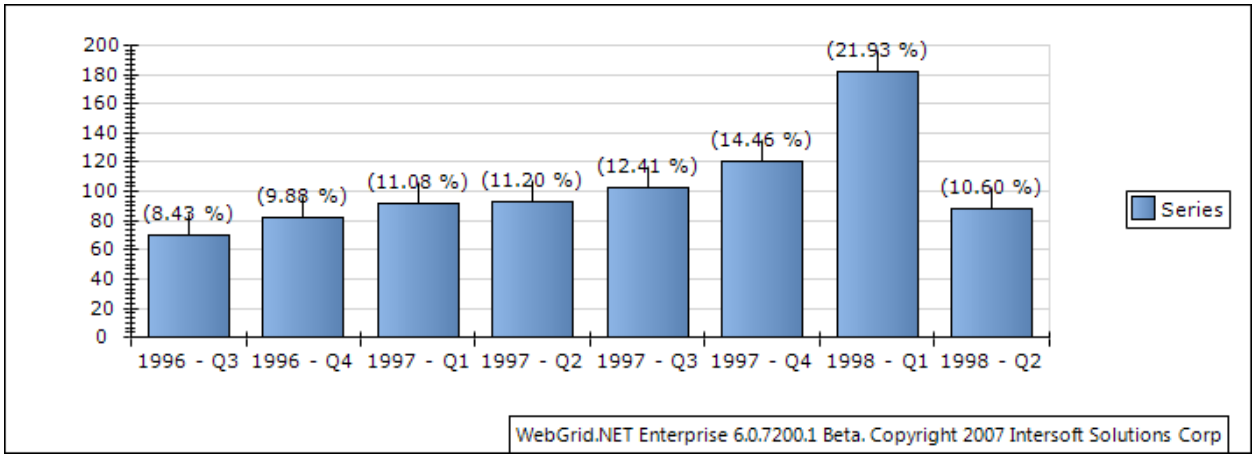


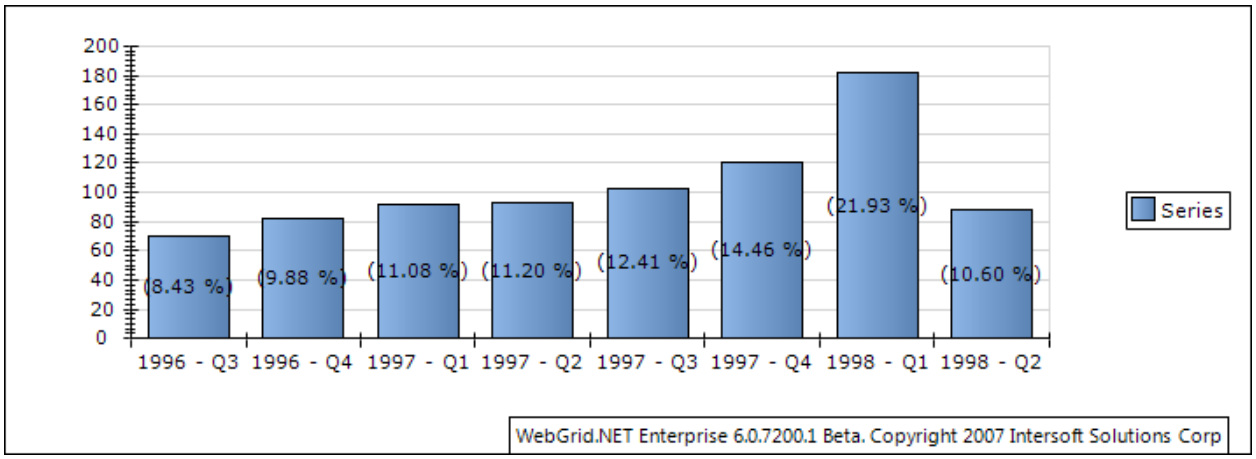
Series Data Point

Data point is a mark that is placed in each series object that represents the data. By default the data point is set to visible = false. If you want to show it you can simply.

```
ChartDataLabelStyle labelStyle = settings.DataLabelStyle;
labelStyle.Visible = true;
labelStyle.Format = "(<percent>)";
```

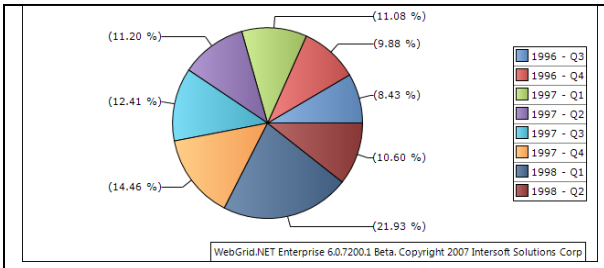
You can also specity the formatting text as in series legend. Further more you can also specify the position of the data points according to your preference.



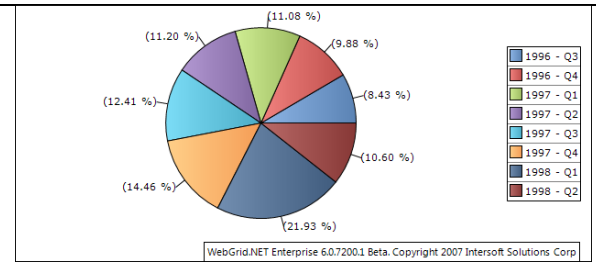


In pie and doughnut chart, the data points can be visualized in more modes:

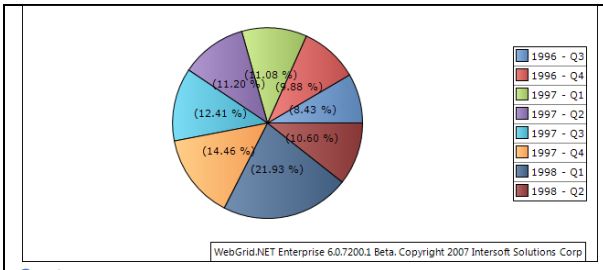
1. Center – the data labels are displayed in the center of the pie segments.
2. Rim the data labels are visualized at the pies rim.
3. Spider the data labels are visualized at aligned in columns on the left and the right side of the pie.



Spider



Rim



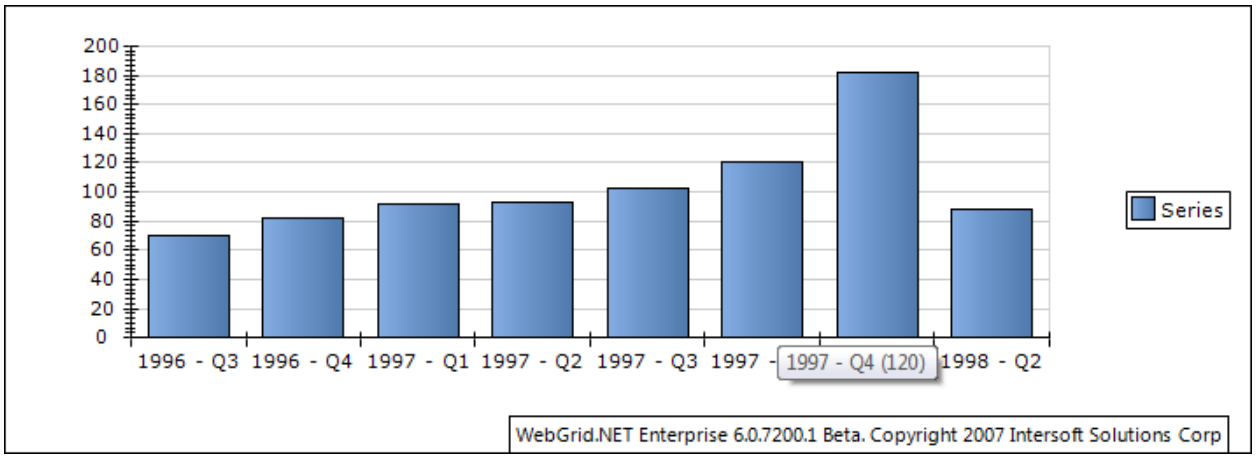
Center

Chart Tool Tip and Interactivity

You can show Chart tooltip and interactivity at chart object by activating ChartInteractiveStyle.

```
Series settings = WebGrid1.ChartSettings.GetSeries();
ChartInteractiveStyle interactive = settings.InteractiveStyle;
interactive.Enabled = true;
```

By doing this you can now show a tooltip at chart object and also attach mouse events (mouse over, mouse out, click, and double click).



You can control the tool tip format from

```
Series settings = WebGrid1.ChartSettings.GetSeries();
ChartInteractiveStyle interactive = settings.InteractiveStyle;
interactive.ToolTip = "<label> (<value>)";
```

Formating available are :

<label> showing the data label

<value> showing the data value

<index> showing the index of the data

<total> showing the total data value

<percent> showing the percentage of current value

Beside tool tip you can also attach mouse events as follow:

```
Series settings = GetSeries();
ChartInteractiveStyle interactive = settings.InteractiveStyle;
interactive.OnClick = "ChartEventClick";

function ChartEventClick(controlId, label, value, index, aggregate,
filterExpression)
{
    label = unescape(label);
    value = unescape(value);
    index = unescape(index);
    aggregate = unescape(aggregate);
    filterExpression = unescape(filterExpression);

    alert (
        "Label : " + label + "\n" +
        "Value : " + value + "\n" +
        "Index : " + index + "\n" +
        "Aggregate : " + aggregate + "\n" +
        "Filter Expression : " + filterExpression + "\n");
}
```

The information passed to client side event handler are : controlId, label, value, index, aggregate, filterExpression of corresponding chart object.

You might want to use the filterExpression as additional filterExpression later for your child query in drill in concept.

General features

The features fall in this section are applied to the charting globally, independent from the chart type used by the user. The features are:

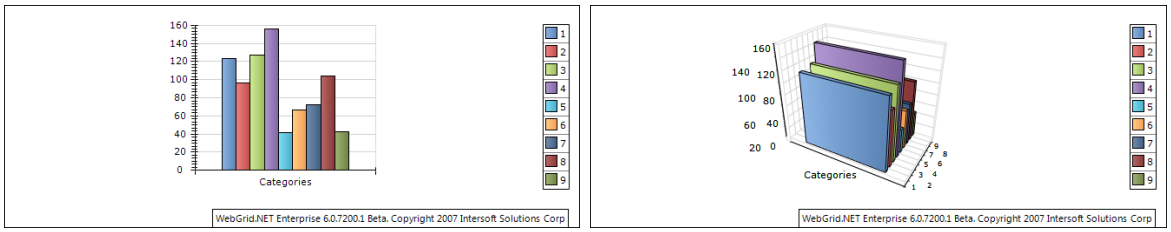
1. Bounds Mode
2. Background
3. Axis
4. Legend
5. Header
6. Image Format
7. Walls

Bounds Mode

Bounds mode determines how the chart image should fit the bounds defined by the layout manager. There are three types of bounds mode.

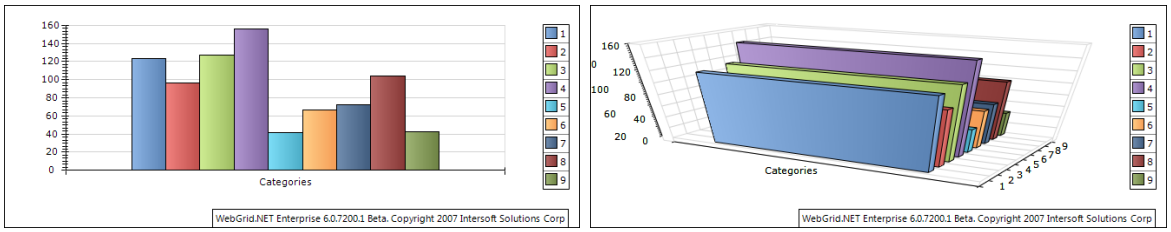
Fit

In this mode the chart image does not break outside the panel bounds by scaling without breaking the aspect ratio of the panel content. This mode is best for 3D Chart types.



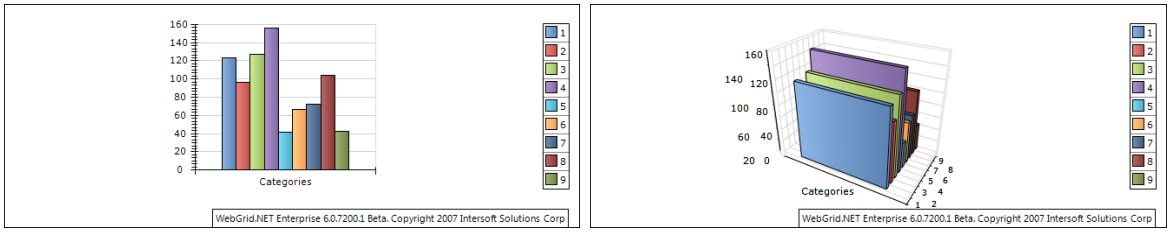
Stretch

In this mode the panel contents does not break outside the panel bounds by scaling along X and Y as much as needed to fit the entire bounds with the panel contents. This bounds mode does not preserve the aspect ratio of the original panel contents. This mode is best for 2D Chart types.



None

In this mode the panel does not try to fit in the bounds defined by the layout manager. It will only make sure that the center of the panel contents matches the center of the bounds.

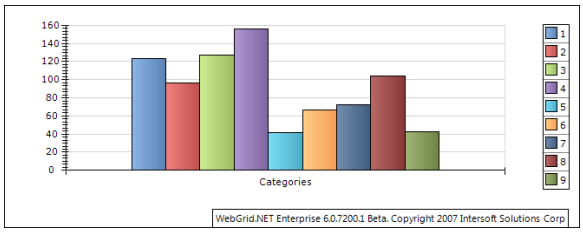


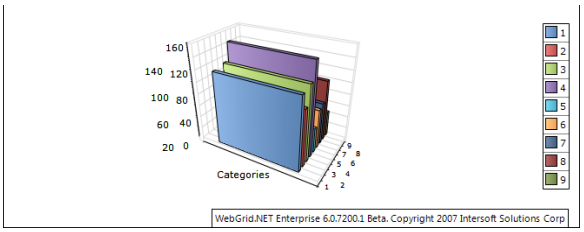
The following code modifies the bounds mode of the chart image.

```
WebGrid1.ChartSettings.BoundsMode = BoundsMode.Stretch;
```

Automatic

In this mode all 2D Chart except Pie and Doughnut will be using Stretch mode, the other chart type will use Fit Mode.

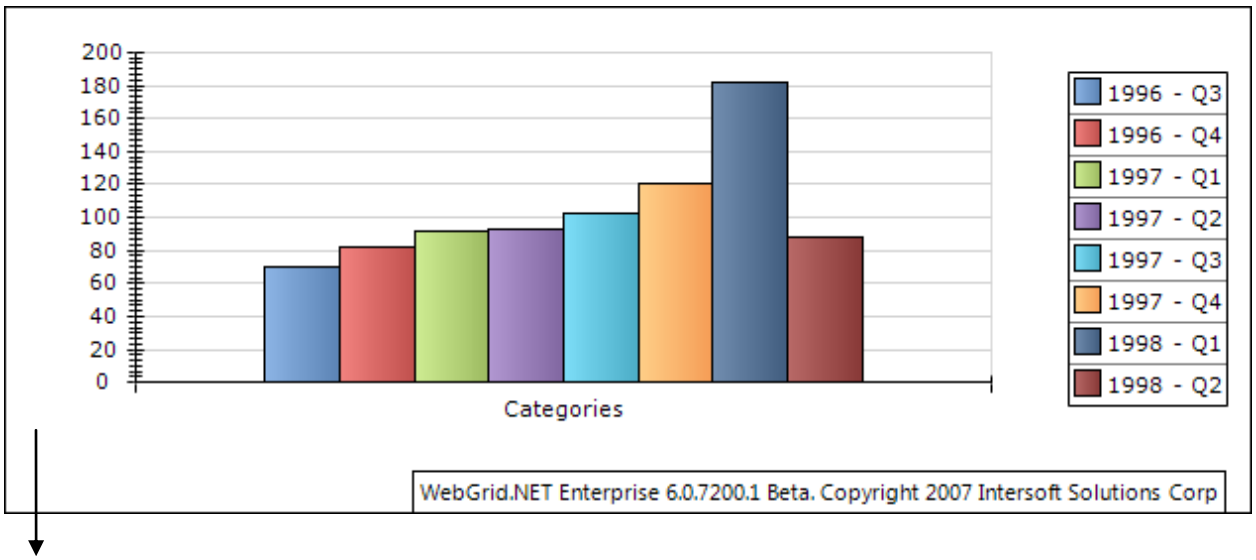




Note that if you modify the Zoom property of chart projection the chart may break outside the bounds.

Background

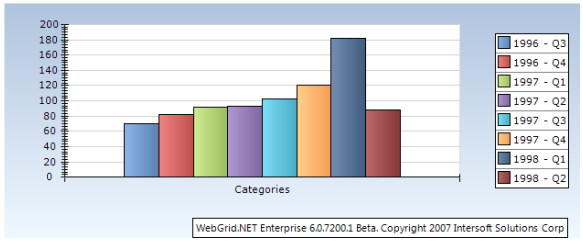
Background of the chart is the chart container's space where the chart image and other chart elements is placed into.



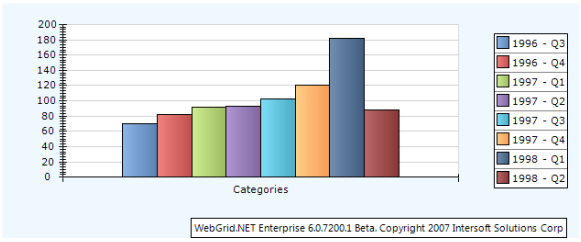
CHART'S BACKGROUND

Various Fill Style

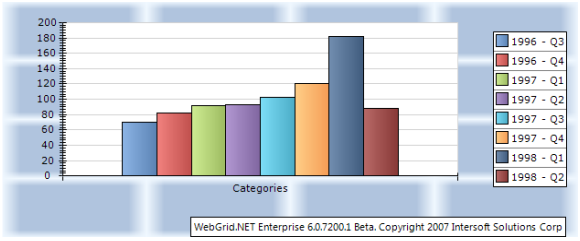
There are four types of fill style (Gradient Fill, Color Fill, Hatch Fill, Image Fill). Each of them has their own attributes. Below are some samples of background fill.



Gradient Fill



Color Fill



Hatch Fill

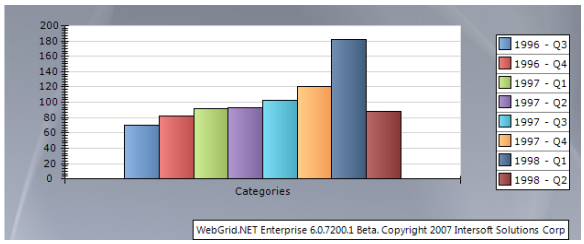


Image Fill

Axis

Follows are things that can be modified for each chart's axis (X, Y, and Z Axis):

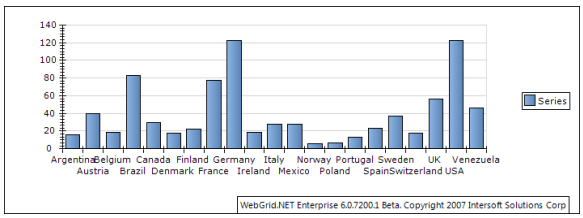
1. Label Orientation Mode
2. Invert Axis
3. Grid Line
4. Strip Style

Label Orientation Mode

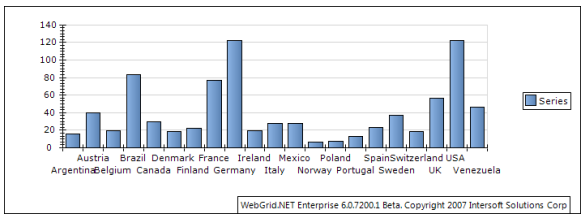
You can modify the label orientation from:

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.XAxis.LabelGenerationMode = LabelGenerationMode.Rotate90;
```

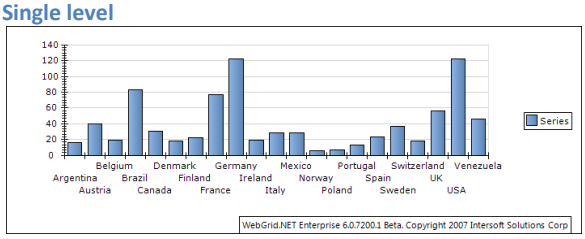
There are seven types of label orientation mode (SingleLevel, Stagger2, Stagger3, Rotate30, Rotate45, Rotate90 and AutoScale). Except SingleLevel, Stagger2, Stagger3 modes, the other entire modes will be applied if the number of data point is quite large. In SingleLevel mode, the chart engine has the capability to determine the size and the stagger level. If you prefer to have only one line at all costs try the AutoScale mode.



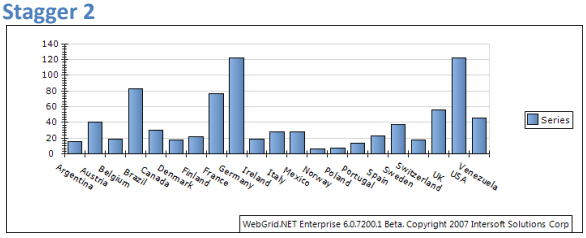
Single level



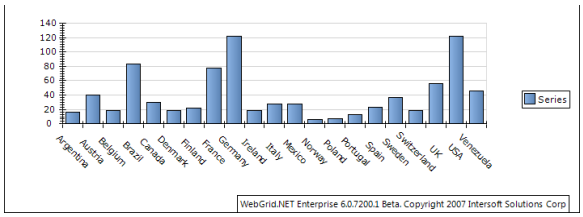
Stagger 2



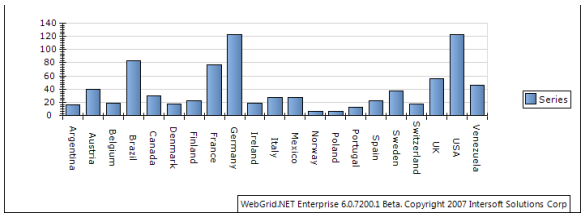
Stagger 3



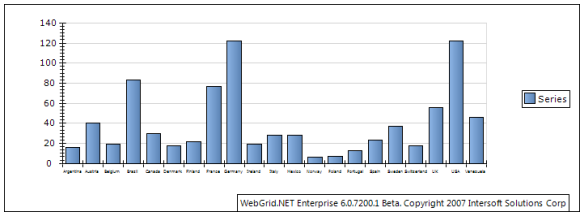
Rotate 30



Rotate 45



Rotate 90



Auto Scale

Invert Axis

You can invert the axis from:

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.YAxis.Invert = true;
```

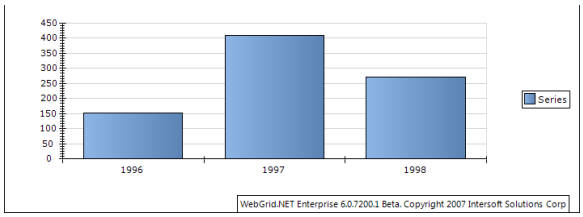
By inverting the axis you'll get different perspective of data visualization as follow:

Legend

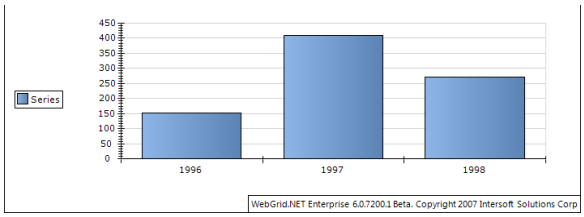
Chart's data legend visibility is controlled from:

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.LegendSettings.Mode = LegendMode.Automatic;
```

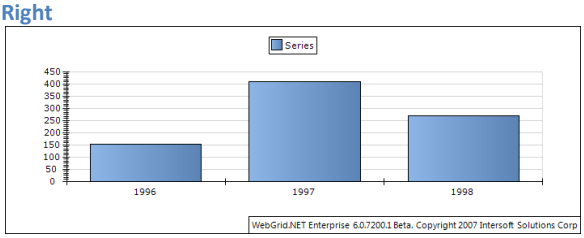
When legend is visible you can controls its appearance. Firstly you can dock the Legend to a specified position (Top, Left, Right, or Bottom).



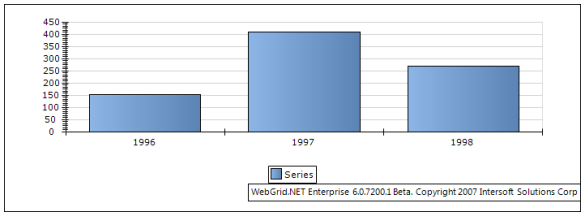
Right



Left

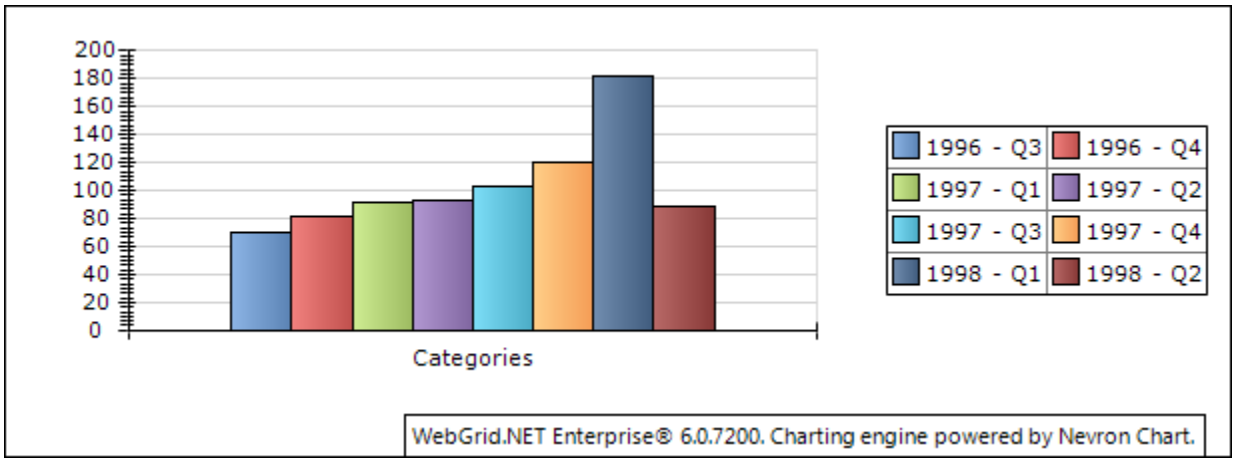


Top

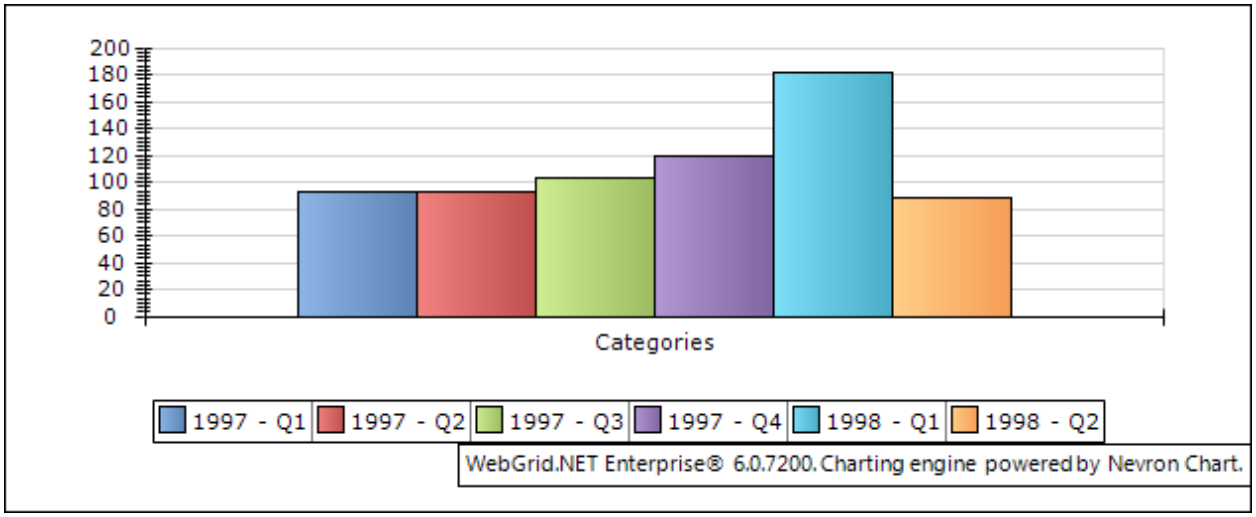


Bottom

Furthermore you can also control the layout of the legend. For example instead of showing the legend data per row (RowsOnly) you can set it per column (ColsOnly)



Alternatively you can also set to ColsFixed or RowsFixed so that the legend will have table format with specified cols / rows that is fixed.



Header

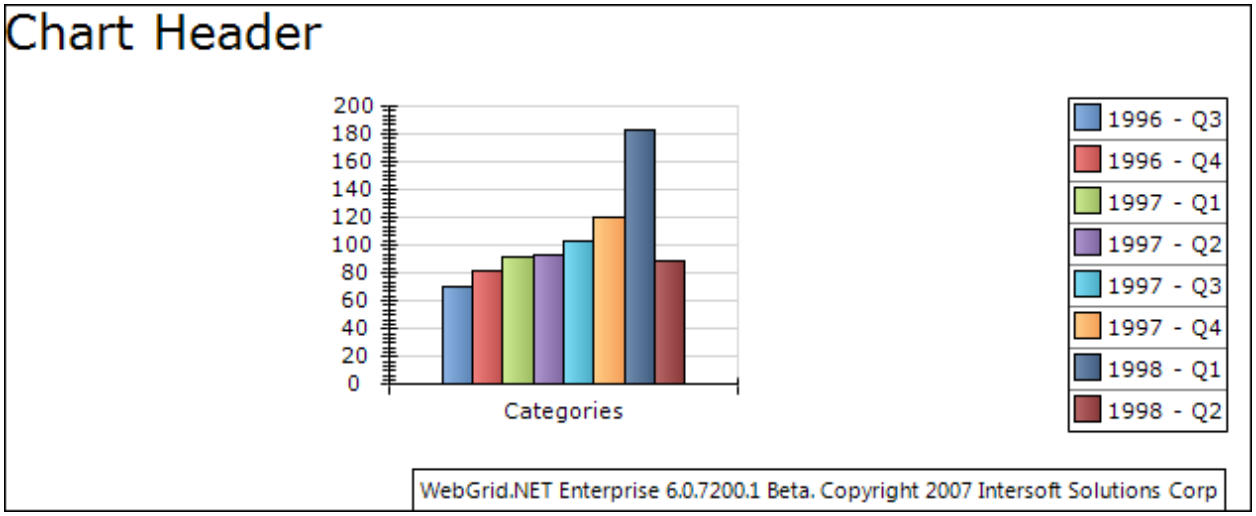
You can add a header / title text to Chart Image from:

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.HeaderSettings.Text = "ChartHeader";
```

By default the position of this Chart Header is at Top Left of the chart image, you can move the position using :

```
ChartSettings chartSettings = WebGrid1.ChartSettings;
chartSettings.HeaderSettings.Alignment = HeaderAlignment.TopRight;
```

Note: you can move the chart header to six positions (TopLeft, TopCenter, TopRIght, BottomLeft, BottomCenter, BottomRight);



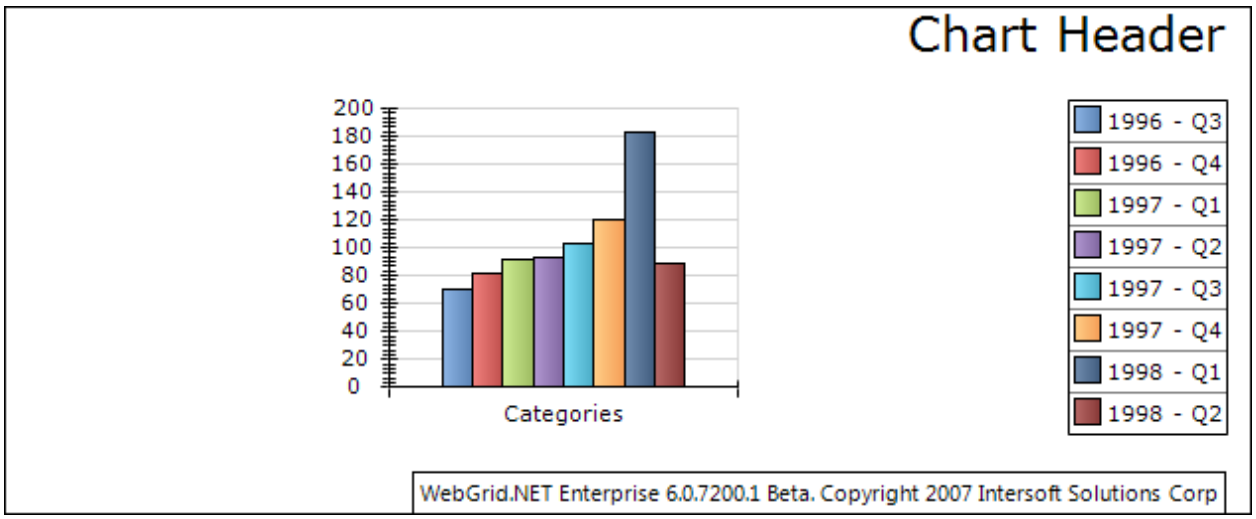
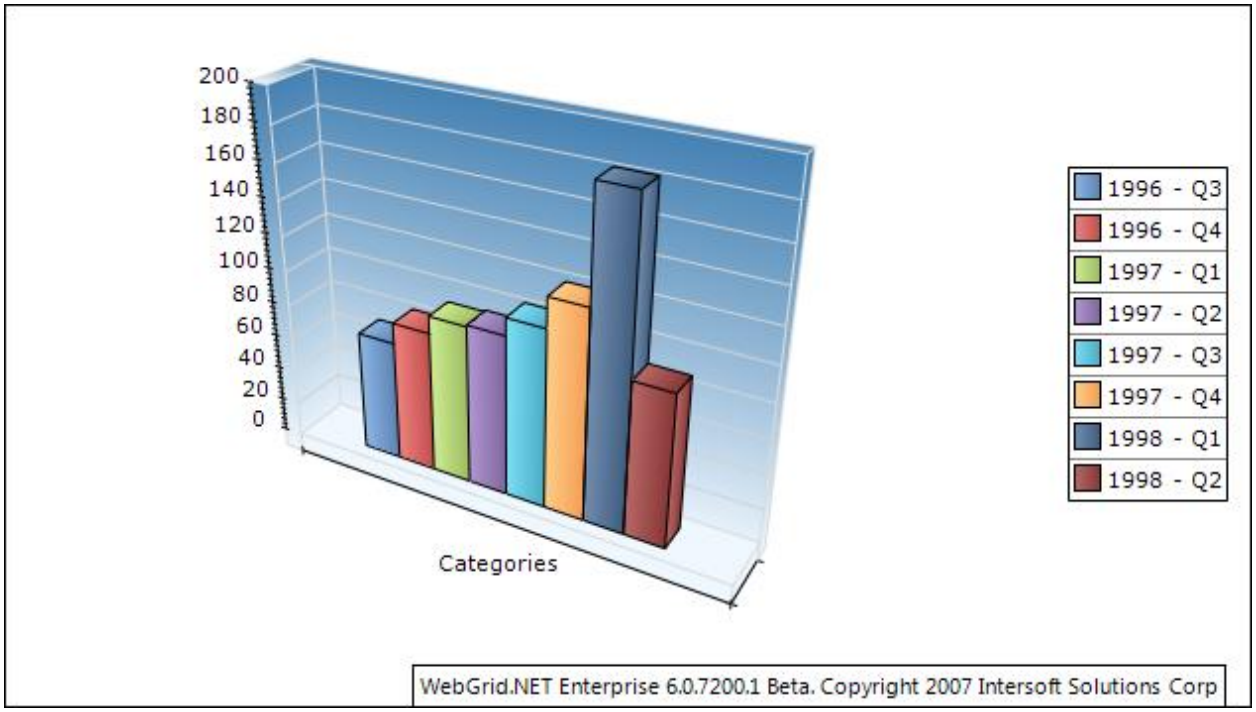


Image Formats

The chart image can be generated into four image format (Gif, Jpeg, Png and Bitmap). The default value is PNG.

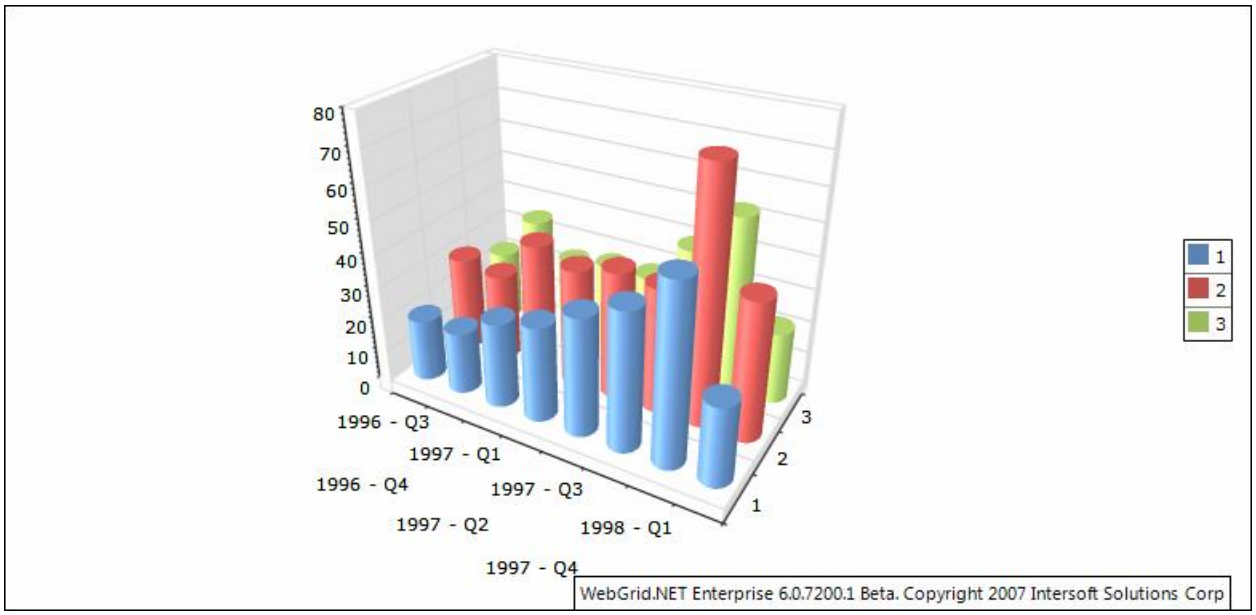
Chart Walls

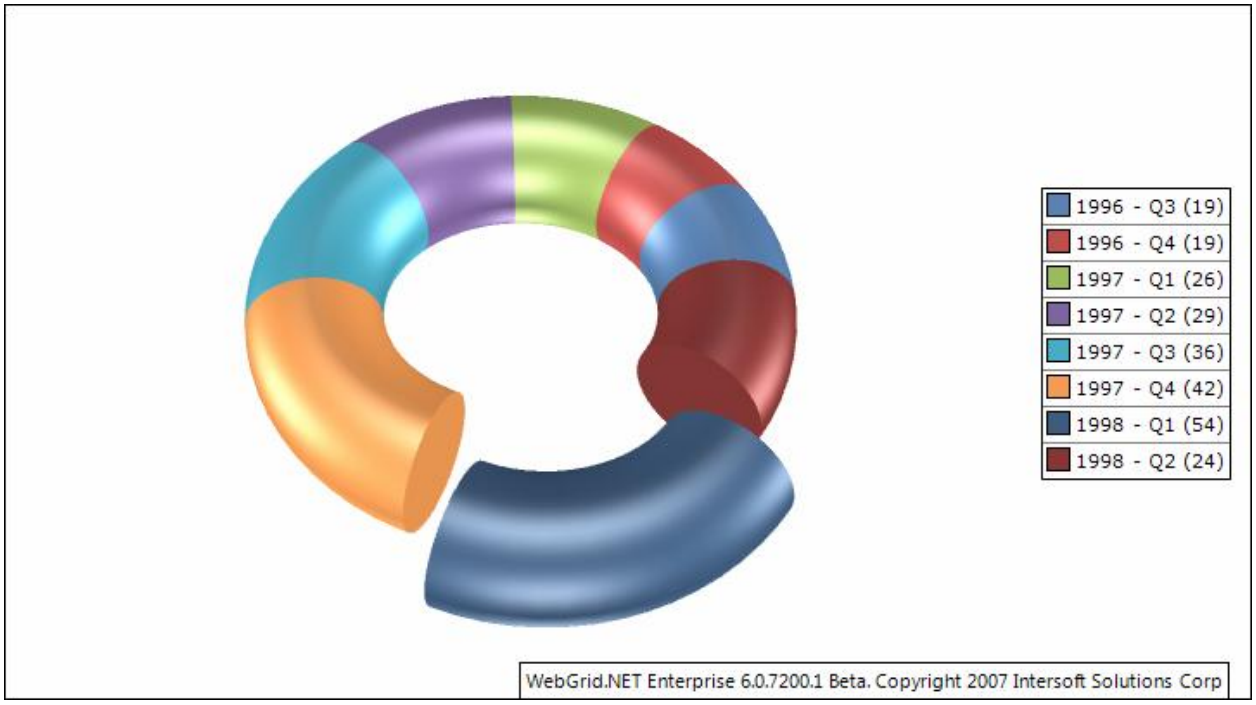
Chart walls are visible when 3D Chart types is used. All the axis settings are also applied to chart walls. Furthermore chart walls also have their own styles properties (BorderStyle, FillStyle and ShadowStyle), you can modified this properties to meet up with your preferences.



Lighting

Lighting in 3D Chart type is one of the main features in WebGrid.NET Enterprise 6.0 Charting feature. When you use any of 3D Chart types you can specify light sources to create stunning chart 3D images.





You can specify the lighting from:

```
<ChartSettings>
```

```
            <LightModelSettings GlobalAmbientLight="Gainsboro"
LightModel="NorthernLights" EnableLighting="true">
                <LightSources>
                    <ISWebGrid:ChartLightSourceSettings Ambient="Black"
Difuse="DimGray" Specular="DimGray"
                        LightSourceType="Directional">
                        <Position X="-1" Z="1" />
                    </ISWebGrid:ChartLightSourceSettings>
                    <ISWebGrid:ChartLightSourceSettings Ambient="Black"
Difuse="DimGray" Specular="DimGray"
                        LightSourceType="Directional">
                        <Position Y="1" Z="1" />
                    </ISWebGrid:ChartLightSourceSettings>
                </LightSources>
            </LightModelSettings>
            <VisualEffectSettings EnableJittering="True" JitteringSteps="4"
RenderSurface="Window"
                ShapeRenderingMode="HighQuality" />
        </ChartSettings>
```

The LightModelSettings property control the lighting effects to all chart types. You might want to use predefined lighting provided to fasten the process declaring the light sources.

However if you prefer to specify your own lighting settings, you can manipulate the following properties.

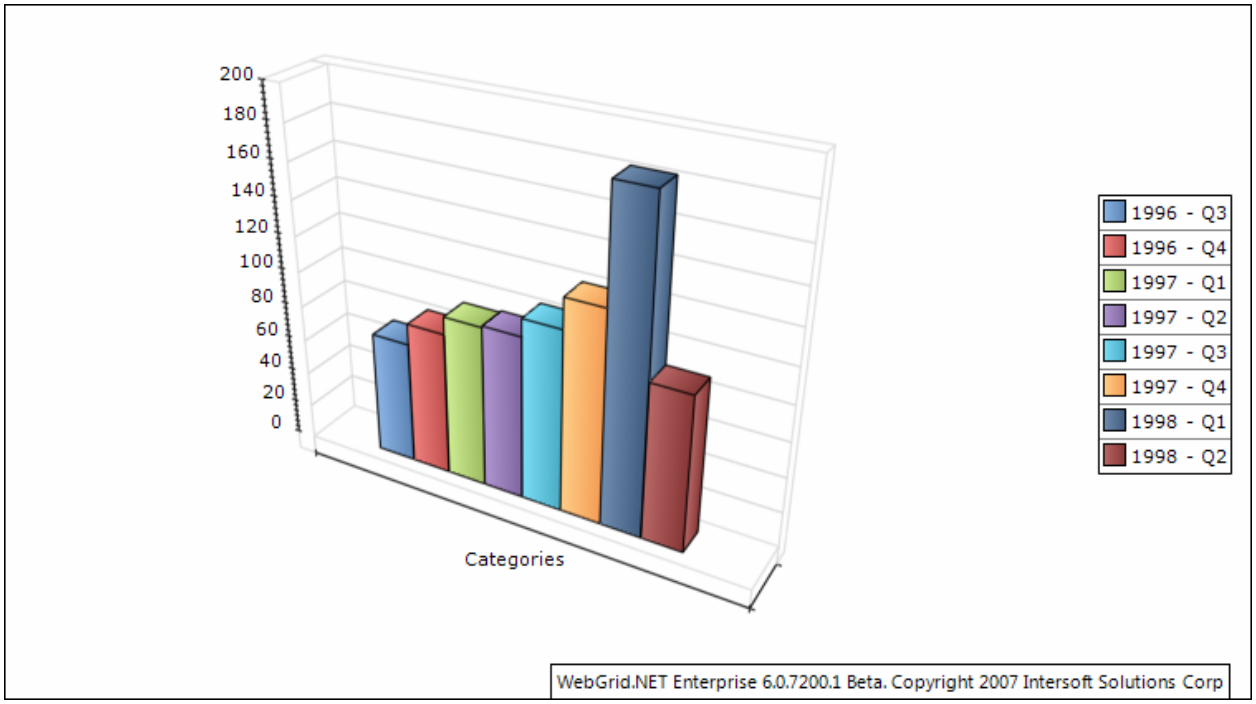
1. GlobalAmbientLight
2. LightSources collection.

    In each light sources you can specify:

    - LightSourceType
    - Ambient Color
    - Difuse Color
    - Specular Color
    - Position
    - Direction
    - Constant Attenuation
    - Linear Attenuation
    - Quadratic Attenuation
    - Spot Cutoff
    - Spot Exponent

Reference samples: WebGridSamples\ ChartLighting.aspx

Projection



In 3D chart types you can change the projection view of the chart from:

```
ChartProjectionSettings projection =
WebGrid1.ChartSettings.ProjectionSettings;
```

You can select the projection types from available predefined projection, or use your own projection settings by manipulating the following properties:

1. Projection Type
2. Elevation
3. Perspective Angle
4. Rotation
5. Viewer Rotation
6. X Depth
7. Y Depth
8. Zoom

Reference Samples: WebGridSamples\ ChartProjection.aspx