Essential Studio 2013 Volume 4 - v.11.4.0.26

Essential PivotGrid WindowsForms

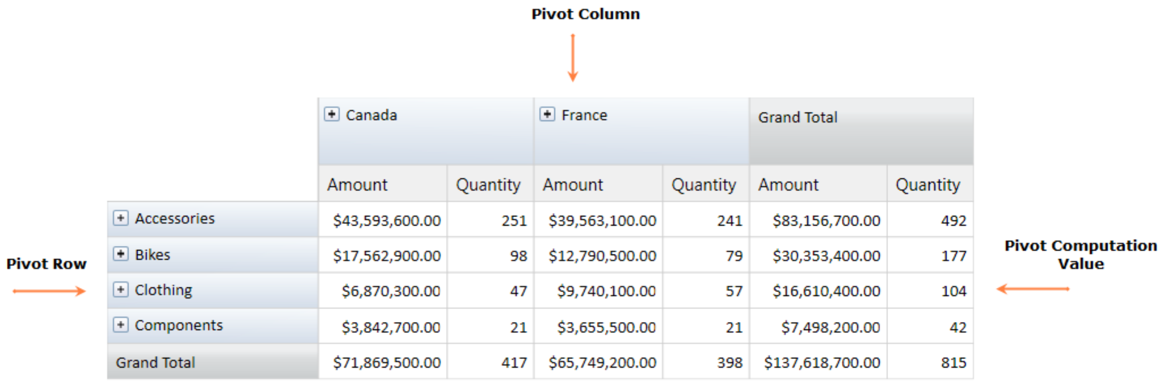
Contents

1 Overview

This section covers information on Essential PivotGrid for Windows Forms, its key features, prerequisites for using the control, its compatibility with various operating systems, and finally, documentation details complimentary with the product. It comprises the following subsections:

1.1 Introduction to Essential PivotGrid Windows Forms

Essential PivotGrid for Windows Forms is a powerful cell-oriented, extensible grid control. It simulates the pivot table feature of Excel. The PivotGrid, as the name implies, pivots data to organize the data in a cross-tabulated form. The major advantage with a pivot grid is that you can extract the desired information from a large list within seconds. Along with presenting the data, a pivot grid also enables you to summarize and group data.



	⊞ Canada		⊞ France		Grand Total	
	Amount	Quantity	Amount	Quantity	Amount	Quantity
⊞ Accessories	$43,593,600.00	251	$39,563,100.00	241	$83,156,700.00	492
⊞ Bikes	$17,562,900.00	98	$12,790,500.00	79	$30,353,400.00	177
⊞ Clothing	$6,870,300.00	47	$9,740,100.00	57	$16,610,400.00	104
⊞ Components	$3,842,700.00	21	$3,655,500.00	21	$7,498,200.00	42
Grand Total	$71,869,500.00	417	$65,749,200.00	398	$137,618,700.00	815

Figure 1: PivotGrid Control

Key Features

Important features of the PivotGrid control are listed below:

- Multilevel drill up/down capability—row drill-down, column drill-down, multilevel row/column drill-down
- Data-binding support
- Auto Calculation of total summary
- Filters
- Grouping support
- Sorting

User Guide Organization

This product comes with numerous samples as well as an extensive documentation to guide you. This User Guide provides detailed information on the features and functionalities of the PivotGrid control. It is organized into the following sections:

- **Overview** - This section gives a brief introduction to our product and its key features.
- **Deployment** - This section elaborates on the install location of the samples, license, and so on.
- **Getting Started** - This section guides you on getting started with a BI application, PivotGrid control, and so on.
- **Concepts and Features** - This section includes features of the PivotGrid control, which are illustrated with use-case scenarios, code examples, and screenshots.

Document Conventions

The following conventions will help you quickly identify the important sections of information while using the content.

Table 1: Conventions Table

Convention	Icon	Description
Note	📝 ***Note:***	Represents important information.
Example	**Example**	Represents an example.
Tip	💡	Represents useful hints that will help you in using the controls/features.
Additional Information	ℹ	Represents additional information on the topic.

1.2 IT Scenarios

The Essential PivotGrid for Windows Forms finds its main application in the financial domain, where it is used to organize and analyze business data.

1.3 Prerequisites and Compatibility

This section covers the requirements mandatory for using the PivotGrid control. It also lists operating systems and browsers compatible with the product.

Prerequisites

The prerequisites details are listed below:

Table 2: Prerequisites Table

Development Environments	• Visual Studio 2012 • Visual Studio 2010 • Visual Studio 2008

.NET Framework Versions	• .NET 4.5 RTM • .NET 4.0 • .NET 3.5 SP1

Compatibility

The compatibility details are listed below:

Table 3: Compatibility Table

Operating Systems	• Windows 8 (32 bit and 64 bit) • Windows Server 2008 (32 bit and 64 bit) • Windows Server 2003 (32 bit and 64 bit) • Windows 7 (32 bit and 64 bit) • Windows Vista (32 bit and 64 bit) • Windows XP

1.4 Documentation

Syncfusion provides the following documentation sections to provide all the necessary information pertaining to the PivotGrid control.

Type of Documentation	Location
Read Me	[drive:]\Program Files\Syncfusion\Essential Studio\<version number>\Infrastructure\Read Me\BI_Windows.html
Release Notes	[drive:]\Program Files\Syncfusion\Essential Studio\<version number>\Infrastructure\Release Notes\BI.html
User Guide (this document)	**Online** http://help.syncfusion.com/resources (Navigate to the Pivot Analyses for Windows Forms User Guide.) ***Note: Click Download as PDF to access a PDF version.*** **Installed Documentation** Dashboard -> Documentation -> Installed Documentation.
Class Reference	**Online** http://help.syncfusion.com/resources (Navigate to the Windows Forms User Guide. Select *Pivot Grid*, and then click the Class Reference link found in the upper right section of the page.)

	Installed Documentation Dashboard -> Documentation -> Installed Documentation.

2 Installation and Deployment

This section covers information on the install location, samples, licensing, patch updates and updates of the current version of Essential Studio. It is comprised of the following subsections:

2.1 Installation

For step-by-step installation procedures for installing Essential Studio, refer to the **Installation** topic under **Installation and Deployment** in the **Common UG**.

See Also

For licensing, patches and information on adding or removing selective components refer to the following topics in Common UG under **Installation and Deployment**.

- Licensing
- Patches
- Add/Remove Components

2.2 Samples and Location

This section covers the location of the installed samples and describes the procedure to run the samples through the Sample Browser and online. It also provides the location of the source code.

Samples Installation Location

The PivotGrid samples are installed in the following location, locally on the disk:

Windows XP:

C:\Syncfusion\Essential Studio<version number>\Windows\PivotGrid.Windows\Samples

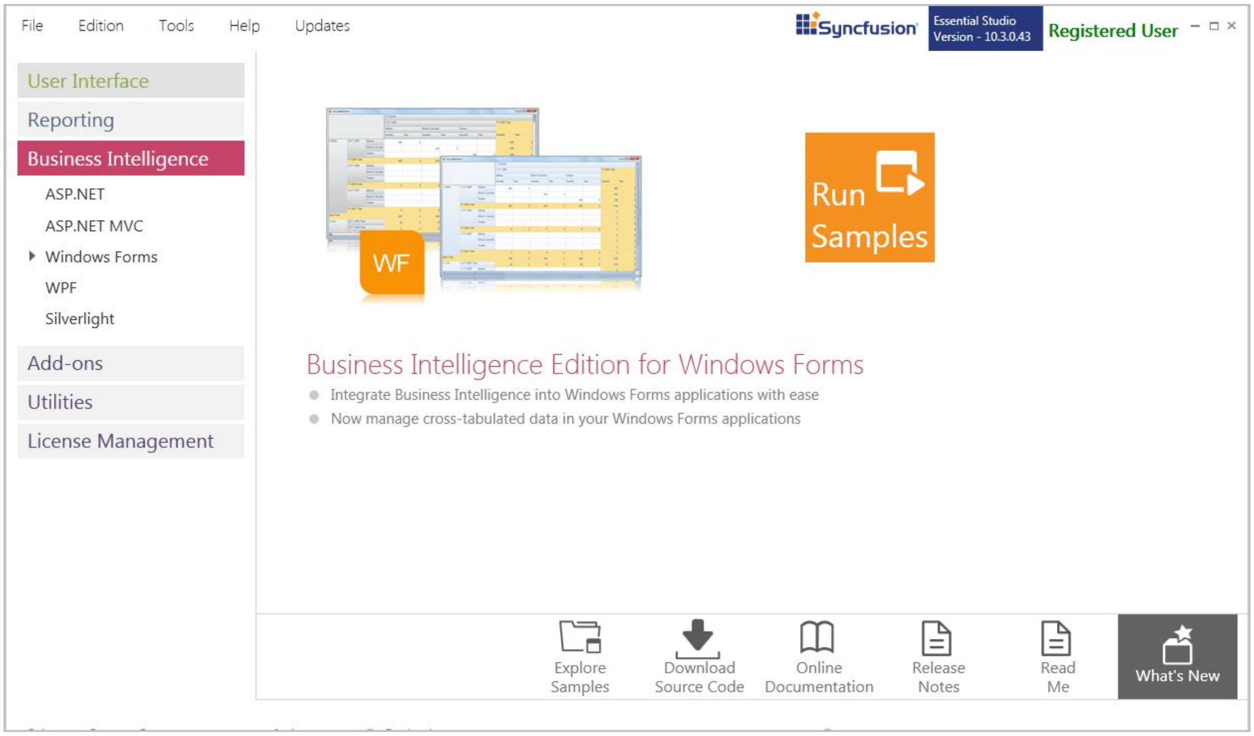
Windows 8/7/Vista:

C:\Users\<user name>\AppData\Local\Syncfusion\EssentialStudio\<version number>\Windows\PivotGrid.Windows\Samples

Viewing Samples

To view the samples, follow the steps below:

1. Click **Start→All Programs→Syncfusion→Essential Studio <version number> →Dashboard**.



Business Intelligence Edition for Windows Forms

- Integrate Business Intelligence into Windows Forms applications with ease
- Now manage cross-tabulated data in your Windows Forms applications

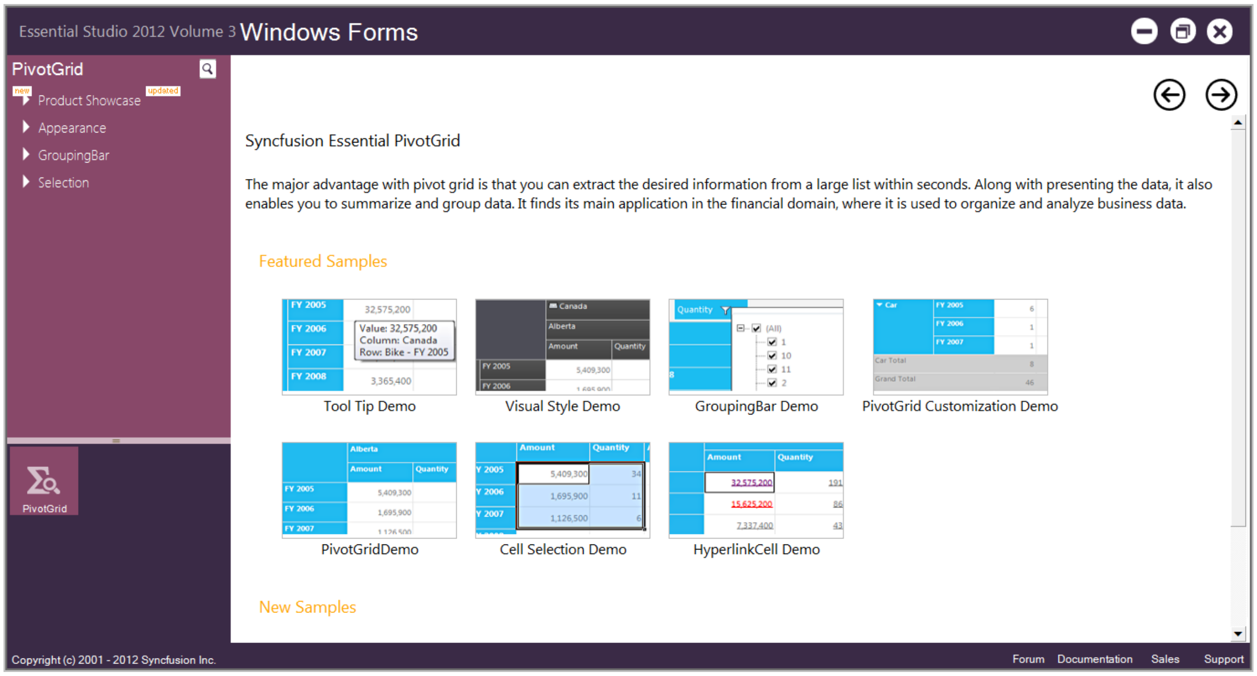
Figure 2: Syncfusion Essential Studio Dashboard BI

2. In the Dashboard window, click **Run Samples** for **Windows Forms** under **BI Edition**.

Note: You can view the samples in any of the following three ways:

- **Run Samples** - Click to view the locally installed samples.
- **Online Samples** - Click to view online samples.
- **Explore Samples** - Explore UI Windows Forms on disk.

3. Click **PivotGrid**. The Pivot Grid samples are displayed.



Syncfusion Essential PivotGrid

The major advantage with pivot grid is that you can extract the desired information from a large list within seconds. Along with presenting the data, it also enables you to summarize and group data. It finds its main application in the financial domain, where it is used to organize and analyze business data.

Figure 3: PivotGrid samples

4. Select any sample and browse through the features.

Source Code Location

The default location of the PivotGrid source code is:

[System Drive]:\Program Files\Syncfusion\Essential Studio\[Version Number]\BI\PivotGrid\Src

2.3 Deployment Requirements

The following assemblies need to be referenced in your application to use Essential PivotGrid for WF.

- Syncfusion.Core
- Syncfusion.Grid.Base
- Syncfusion.Grid.Windows
- Syncfusion.Shared.Base
- Syncfusion.Linq.Base
- Syncfusion.Shared.Windows
- Syncfusion.PivotAnalysis.Base
- Syncfusion.PivotAnalysis.Windows

2.4 Properties Table for PivotGrid

Table 4: Properties Table

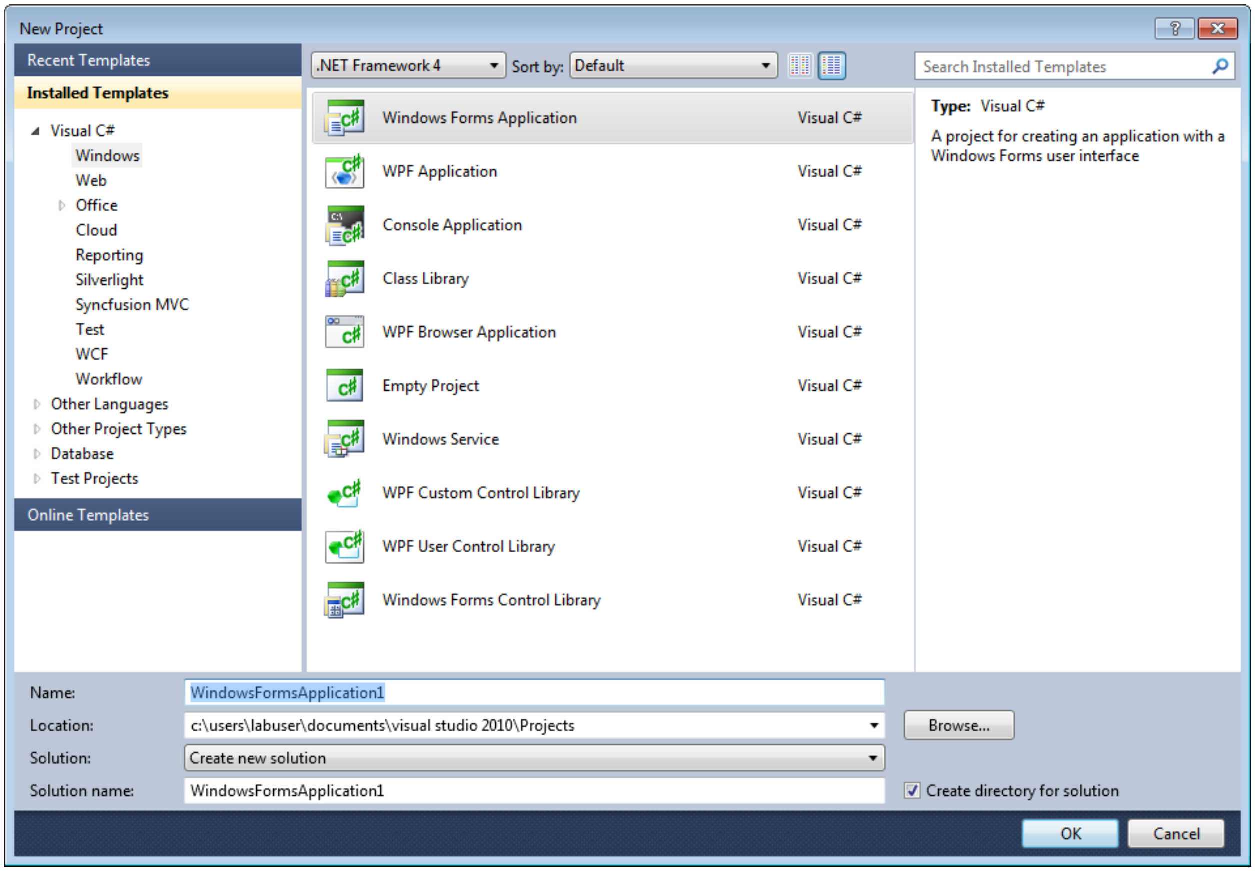
Property Name	Type	Description
DeferLayoutUpdate	bool	Gets or sets a value to specify whether the layout should be updated immediately after updating the pivoting info, or if it should wait for a *Refresh()* call.
FreezeHeaders	bool	Gets or sets a value to specify whether headers of a grid has to be frozen or not.
DataSource	object	Gets or sets the source of data for a pivot table. This object should be an IEnumerable or IQueryable list.
PivotCalculations	Hashtable	Gets the collection of Pivot Calculations.
PivotColumns	Hashtable	Gets the collection of pivot columns.
PivotEngine	PivotEngine	Gets or sets the pivot engine for a grid.
PivotRows	Hashtable	Gets the collection of pivot rows.
ShowCalculationsAs Columns	bool	Gets or sets a value to specify whether calculations should appear as rows or columns. The default behavior is for calculations to appear as columns.
ShowGrandTotals	bool	Gets or sets a value to specify whether grand total calculations should be computed by the engine.
PivotCellInfo	PivotCellInfo	Gets or sets the PivotCellInfo in order to check the cell type.

3 Getting Started

3.1 Creating PivotGrid through Visual Studio

To create the PivotGrid through Visual Studio:

1. Open the **Start** menu, and then click **Microsoft Visual Studio 2008**.
2. On the **File** menu, click **New Project**. The **New Project** dialog box appears as follows.



New Project

.NET Framework 4 Sort by: Default

Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: WindowsFormsApplication1

Location: c:\users\labuser\documents\visual studio 2010\Projects

Solution: Create new solution

Solution name: WindowsFormsApplication1

☑ Create directory for solution

Recent Templates
Installed Templates
Visual C#
Windows
Web
Office
Cloud
Reporting
Silverlight
Syncfusion MVC
Test
WCF
Workflow
Other Languages
Other Project Types
Database
Test Projects
Online Templates
Search Installed Templates
Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
Class Library	Visual C#
WPF Browser Application	Visual C#
Empty Project	Visual C#
Windows Service	Visual C#
WPF Custom Control Library	Visual C#
WPF User Control Library	Visual C#
Windows Forms Control Library	Visual C#
Browse...
OK
Cancel

Figure 4: New Project Dialog Box

3. Select **WindowsForms Application**, and then click **OK**.
4. Drag the **PivotGridControl** control from the **Toolbox** to the **Design** page.

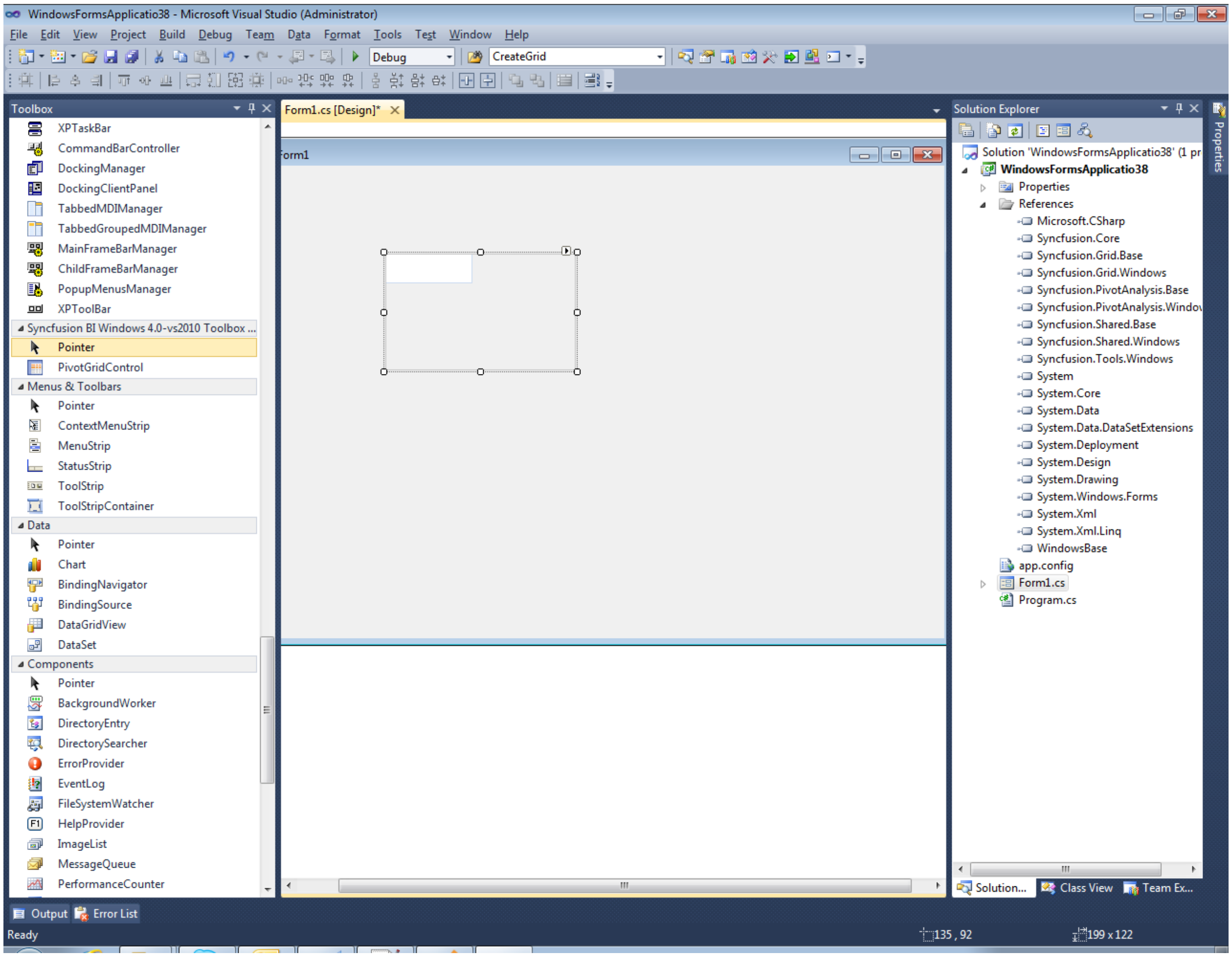


Figure 5: PivotGrid in Design page

3.2 Populating Data to PivotGrid

The PivotGrid requires the following information in order to populate data:

- **ItemSource** - The data source for the pivot table. This object should be either an IEnumerable list or a DataTable.
- **PivotRows** - Elements that need to be added in PivotGrid rows.
- **PivotColumns** - Elements that need to be added in PivotGrid columns.
- **PivotCalculations** - Calculation values that appear as value cells in PivotGrid.

To populate a PivotGrid with the sample IList data, refer to the code below.

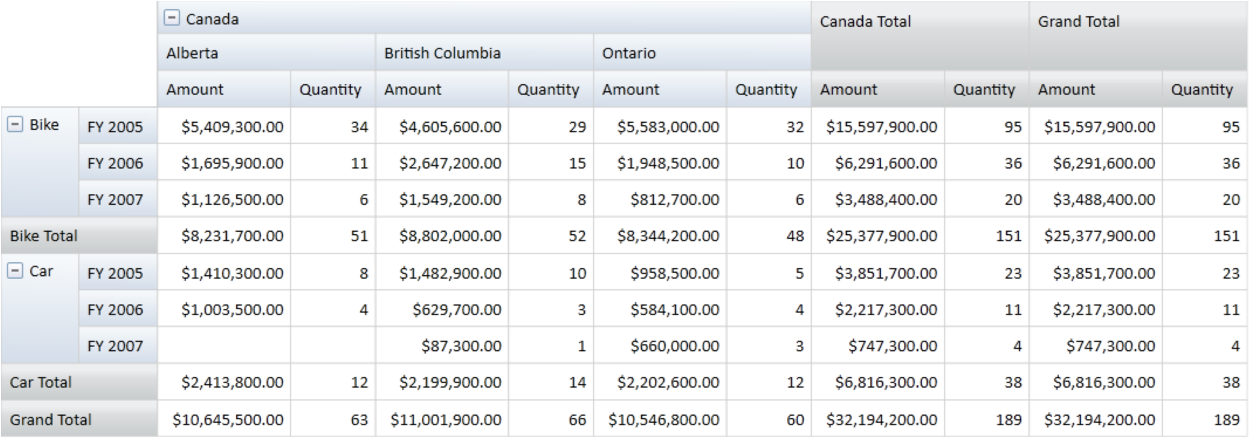
[C#]

```
protected void Form1_Load(object sender, System.EventArgs e)
```

```
{
            // Specifying the ItemSource for Pivot Grid
            this.PivotGridControl1.ItemSource =
ProductSales.GetSalesData();
            // Adding Pivot Rows to Grid
            this.PivotGridControl1.PivotRows.Add(new PivotItem {
FieldMappingName = "Product", TotalHeader = "Total" });
            this.PivotGridControl1.PivotRows.Add(new PivotItem {
FieldMappingName = "Year", TotalHeader = "Total" });
            // Adding Pivot Colums to Grid
            this.PivotGridControl1.PivotColumns.Add(new PivotItem {
FieldMappingName = "Country", TotalHeader = "Total" });
            this.PivotGridControl1.PivotColumns.Add(new PivotItem {
FieldMappingName = "State", TotalHeader = "Total" });
            // Adding PivotCalculations to Grid
            this.PivotGridControl1.PivotCalculations.Add(new
PivotComputationInfo { FieldName = "Amount",Format="C" , SummaryType =
SummaryType.DoubleTotalSum });
            this.PivotGridControl1.PivotCalculations.Add(new
PivotComputationInfo { FieldName = "Quantity", Format ="#,##0"});


}
```

```
[VB]


Protected Sub Form1_Load (ByVal sender As Object, ByVal e As
System.EventArgs)
                  ' Specifying the ItemSource for Pivot Grid
                  Me.PivotGridControl1.ItemSource =
ProductSales.GetSalesData()
                  ' Adding Pivot Rows to Grid
                  Me.PivotGridControl1.PivotRows.Add(New PivotItem With
{.FieldMappingName = "Product", .TotalHeader = "Total"})
                  Me.PivotGridControl1.PivotRows.Add(New PivotItem With
{.FieldMappingName = "Year", .TotalHeader = "Total"})
                  ' Adding Pivot Colums to Grid
                  Me.PivotGridControl1.PivotColumns.Add(New PivotItem
With {.FieldMappingName = "Country", .TotalHeader = "Total"})

                  Me.PivotGridControl1.PivotColumns.Add(New PivotItem
With {.FieldMappingName = "State", .TotalHeader = "Total"})

                  ' Adding PivotCalculations to Grid

                  Me.PivotGridControl1.PivotCalculations.Add(New
PivotComputationInfo With {.FieldName = "Amount", .Format="C",
.SummaryType = SummaryType.DoubleTotalSum})

                  Me.PivotGridControl1.PivotCalculations.Add(New
PivotComputationInfo With {.FieldName = "Quantity", .Format ="#,##0"})


End Sub
```

When the code above runs, the following output will be generated.



		⊟ Canada						Canada Total		Grand Total	
		Alberta		British Columbia		Ontario					
		Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity
⊟ Bike	FY 2005	$5,409,300.00	34	$4,605,600.00	29	$5,583,000.00	32	$15,597,900.00	95	$15,597,900.00	95
	FY 2006	$1,695,900.00	11	$2,647,200.00	15	$1,948,500.00	10	$6,291,600.00	36	$6,291,600.00	36
	FY 2007	$1,126,500.00	6	$1,549,200.00	8	$812,700.00	6	$3,488,400.00	20	$3,488,400.00	20
Bike Total		$8,231,700.00	51	$8,802,000.00	52	$8,344,200.00	48	$25,377,900.00	151	$25,377,900.00	151
⊟ Car	FY 2005	$1,410,300.00	8	$1,482,900.00	10	$958,500.00	5	$3,851,700.00	23	$3,851,700.00	23
	FY 2006	$1,003,500.00	4	$629,700.00	3	$584,100.00	4	$2,217,300.00	11	$2,217,300.00	11
	FY 2007			$87,300.00	1	$660,000.00	3	$747,300.00	4	$747,300.00	4
Car Total		$2,413,800.00	12	$2,199,900.00	14	$2,202,600.00	12	$6,816,300.00	38	$6,816,300.00	38
Grand Total		$10,645,500.00	63	$11,001,900.00	66	$10,546,800.00	60	$32,194,200.00	189	$32,194,200.00	189

Figure 6: Pivot Grid Control with Pivoted Data

3.3 Class Diagram

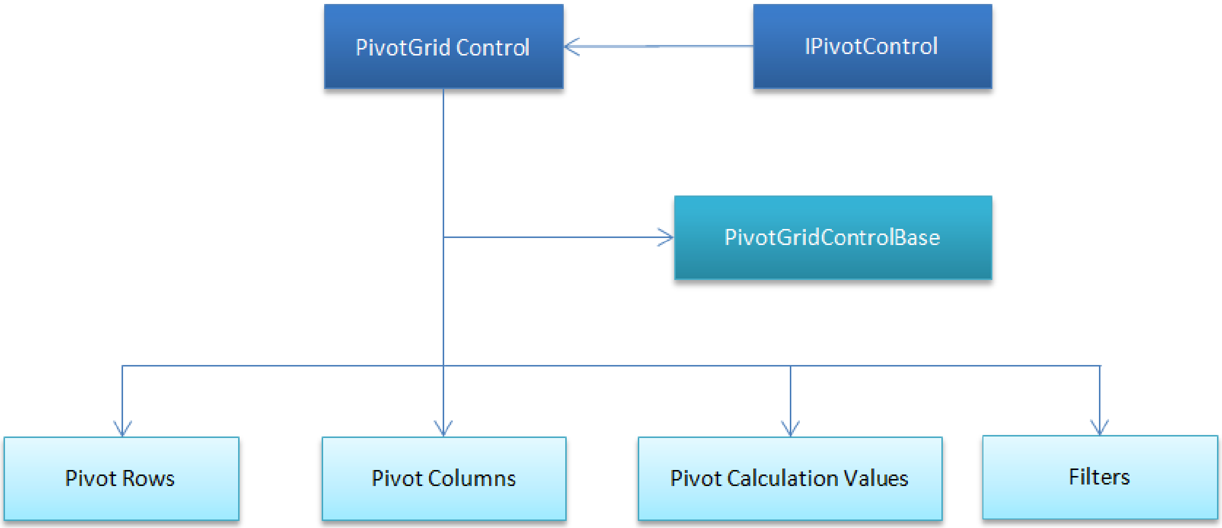


Figure 7: PivotGrid WF Class Diagram

4 Concepts and Features

4.1 PivotItem

PivotItem is an item in a PivotTable field. It provides the information needed to define a pivot item for either a row or column. It consists of the following fields.

Table 5: Properties Table for PivotItem

Property Name	Description	Type	Value it Accepts	Reference link
Comparer	Gets or sets the IComparer object used for sorting. If this value is null, then sorting will be performed under the assumption that this field is IComparable.	IComparer	-	-
FieldHeader	Gets or sets the title you want to see in the header for this pivot item.	string	-	-
FieldMappingName	Gets or sets the property's mapping name.	string	-	-
Format	Gets or sets the format item for the specified field.	string	-	-
TotalHeader	Gets or sets the string you want to append to the pivot item's summary cells.	string	-	-

4.1.1 Defining PivotItem in Code-Behind

The PivotItem can be defined in code-behind. The following code example illustrates this.

[C#]

```
// Defining PivotItem
PivotItem m_PivotItem = new PivotItem() { FieldHeader="Product",
FieldMappingName ="Product", TotalHeader ="Total" };
// Adding PivotItem to PivotRows
this.PivotGridControl1.PivotRows.Add(m_PivotItem);
```

[VB]

```
' Defining PivotItem
Dim m_PivotItem As PivotItem = New PivotItem() With
{.FieldHeader="Product", .FieldMappingName ="Product", .TotalHeader
="Total"}
' Adding PivotItem to PivotRows
Me.PivotGridControl1.PivotRows.Add(m_PivotItem)
```

4.1.2 Sorting Using PivotItem

By default, the PivotGrid will sort data in ascending order. The sorting order can be changed using the *Comparer* field of *PivotItem*.

[C#]

```
// Adding Pivot Rows to Grid with FieldMappingName, TotalHeader and
Comparer
this.PivotGridControl1.PivotRows.Add(new PivotItem { FieldMappingName =
"Product", TotalHeader = "Total", Comparer = new ReverseOrderComparer()
});


/// <summary>
/// Reverse Order Comparer for sorting data in Descending order
/// </summary>
public class ReverseOrderComparer : IComparer
{
   #region IComparer Members

   public int Compare(object x, object y)
   {
      if (x == null && y == null)
        return 0;
      else if (y == null)
        return 1;
      else if (x == null)
        return -1;
      else
        return -x.ToString().CompareTo(y.ToString());
   }

   #endregion
}
```

```
[VB]
' Adding Pivot Rows to Grid with FieldMappingName, TotalHeader and
Comparer
Me.PivotGridControl1.PivotRows.Add(New PivotItem With
{.FieldMappingName = "Product", .TotalHeader = "Total", .Comparer = New
ReverseOrderComparer()})


''' <summary>
''' Reverse Order Comparer for sorting data in Descending order
''' </summary>
public class ReverseOrderComparer : IComparer
'   #Region "IComparer Members"

   public Integer Compare(Object x, Object y)
        If x Is Nothing AndAlso y Is Nothing Then
            Return 0
        ElseIf y Is Nothing Then
```

```
            Return 1
        ElseIf x Is Nothing Then
            Return -1
        Else
            Return -x.ToString().CompareTo(y.ToString())
        End If

'   #End Region
```

4.2 PivotComputationInfo

This class holds the information needed for calculations that appear in PivotGrid. For each calculation, there is an associated *PivotComputationInfo* object that is added to the *PivotCalculations* collection. The properties available in the *PivotComputationInfo* are listed in the following table.

Table 6: Properties Table for PivotComputationInfo

Property Name	Description	Type	Value it Accepts	Reference link
CalculationName	Gets or sets what is displayed in the pivot table if more than one calculation is included in the PivotGrid.	string	-	-
Description	Gets or sets a description of the calculation.	string	-	-
FieldName	Gets or sets the name of the property to be used in this calculation.	string	-	-
Format	Gets of sets the format string to be used to format the calculation results in the PivotGrid.	string	-	-
Summary	Gets or sets the SummaryBase object that is used to define this	SummaryBase	-	-

	calculation. This value is automatically set when you specify any non-custom value of SummaryType; if you specify SummaryType.Custom, then you are required to set Summary to be an instance of your custom SummaryBase-derived object.			
SummaryType	Gets or sets the SummaryType enumeration for this calculation. Setting it to any value of than Custom will also properly set Summary.	SummaryType	DoubleTotalSum DoubleAverage DoubleMaximum DoubleMinimum DoubleStandardDeviation DoubleVariance Count DecimalTotalSum IntTotalSum Custom	-

4.2.1 Defining PivotComputationInfo and Code-Behind

The *PivotComputationInfo* can be defined in C# or VB code.

```
[C#]


// Defining PivotComputationInfo

PivotComputationInfo m_PivotComputationInfo = new
PivotComputationInfo() { CalculationName="Amount", FieldName="Amount",
SummaryType= SummaryType.Count };
// Adding PivotComputationInfo to PivotCalculations
this.pivotGrid1.PivotCalculations.Add(m_PivotComputationInfo);
```

```
[VB]


' Defining PivotComputationInfo
Dim m_PivotComputationInfo As PivotComputationInfo = New
PivotComputationInfo() With {.CalculationName="Amount",
.FieldName="Amount", .SummaryType= SummaryType.Count}
' Adding PivotComputationInfo to PivotCalculations
Me.pivotGrid1.PivotCalculations.Add(m_PivotComputationInfo)
```

4.2.2 Format String in PivotComputationInfo

The *PivotComputationInfo* property replaces each format specification in a specified string with the textual equivalent of a corresponding value.

```
[C#]


// Decimal Format
PivotComputationInfo m PivotComputationInfo = new
PivotComputationInfo() { CalculationName="Total", FieldName="Quantity",
SummaryType= SummaryType.Count, Format="0.00"};
```

```
[VB]


' Decimal Format
Dim m_PivotComputationInfo As PivotComputationInfo = New
PivotComputationInfo() With {.CalculationName="Total",
.FieldName="Quantity", .SummaryType= SummaryType.Count, .Format="0.00"}
```

The following table lists the different types of format settings.

Table 7: Types of format settings

Format	Description
0.00	Decimal
C	Currency
#,##0	Thousand Separator
# ' degrees'	Literal String Specifier
D	Long Date

4.3 SummaryType

SummaryType determines the type of field summary such as count, sum, average, etc. It is an enumerator that should be defined in the *PivotComputationInfo* class. It contains the following types for performing calculations.

Table 8: Summary Type

Summary Type	Description
DoubleTotalSum	Computes the sum of double or integer values.
DoubleAverage	Computes the simple average of double or integer values.
DoubleMaximum	Computes the maximum of double or integer values.
DoubleMinimum	Computes the minimum of double or integer values.
DoubleStandardDeviation	Computes the standard deviation of double or integer values.
DoubleVariance	Computes the variance of double or integer values.
Count	Computes the count of double or integer values.
DecimalTotalSum	Computes the sum of decimal values.

IntTotalSum	Computes the sum of integer values.
Custom	Specifies that you are using a custom SummaryBase object to define the calculation.

4.4 Sorting

Sorting enables you to quickly visualize and understand the data in a better way, organize and find the data that you want, and ultimately make more effective decisions. By default, the PivotGrid will populate the data in ascending order. The sorting order can be changed using the *Comparer* field of *PivotItem*.

```
[C#]
// Adding Pivot Rows to Grid with FieldMappingName, TotalHeader and
Comparer
this.PivotGridControl1.PivotRows.Add(new PivotItem { FieldMappingName =
"Product", TotalHeader = "Total", Comparer = new ReverseOrderComparer()
});


/// <summary>
/// Reverse Order Comparer for sorting data in Descendingorder
/// </summary>
public class ReverseOrderComparer : IComparer
{
   #region IComparer Members

   public int Compare(object x, object y)
   {
      if (x == null && y == null)
        return 0;
      else if (y == null)
        return 1;
      else if (x == null)
        return -1;
      else
        return -x.ToString().CompareTo(y.ToString());
   }

   #endregion
}

```

[VB]

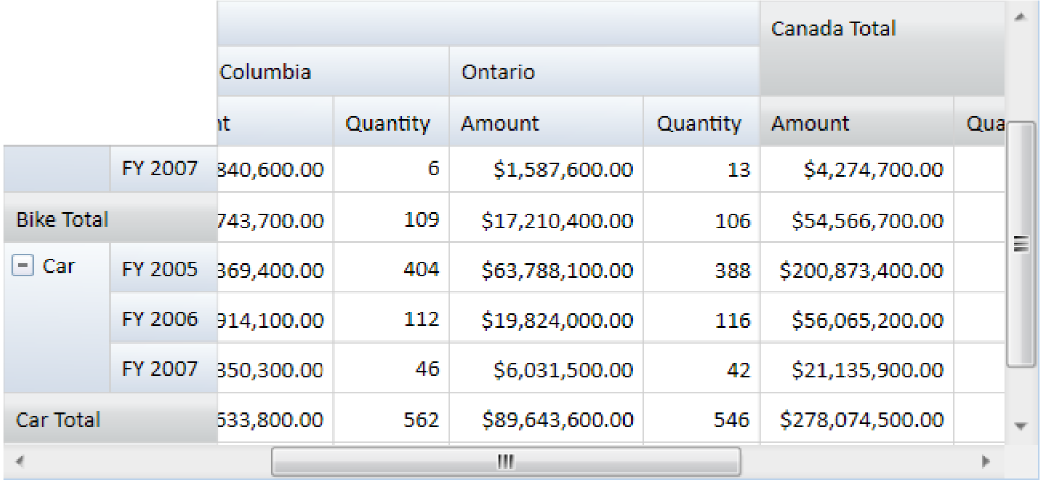
```
' Adding Pivot Rows to Grid with FieldMappingName, TotalHeader and
Comparer
Me.PivotGridControl1.PivotRows.Add(New PivotItem With
{.FieldMappingName = "Product", .TotalHeader = "Total", .Comparer = New
ReverseOrderComparer()})


''' <summary>
''' Reverse Order Comparer for sorting data in Descending order
''' </summary>
public class ReverseOrderComparer : IComparer
'    #Region "IComparer Members"

   public Integer Compare(Object x, Object y)
        If x Is Nothing AndAlso y Is Nothing Then
            Return 0
        ElseIf y Is Nothing Then
            Return 1
        ElseIf x Is Nothing Then
            Return -1
        Else
            Return -x.ToString().CompareTo(y.ToString())
        End If

'    #End Region

```

4.5 Freezing Headers

The PivotGrid for Windows Forms provides built-in support for freezing column and row headers. This is achieved by setting *FreezeHeaders* property of *PivotGrid* to *True*. This feature also enables scrolling through the value cells.

```
[C#]
// To Freeze Grid Headers
this.PivotGridControl1.FreezeHeaders = true;
```

```
[VB]
' To Freeze Grid Headers
Me.PivotGridControl1.FreezeHeaders = True
```



						Canada Total	
		Columbia		Ontario			
		ıt	Quantity	Amount	Quantity	Amount	Qua
	FY 2007	340,600.00	6	$1,587,600.00	13	$4,274,700.00	
Bike Total		743,700.00	109	$17,210,400.00	106	$54,566,700.00	
⊟ Car	FY 2005	369,400.00	404	$63,788,100.00	388	$200,873,400.00	
	FY 2006	914,100.00	112	$19,824,000.00	116	$56,065,200.00	
	FY 2007	350,300.00	46	$6,031,500.00	42	$21,135,900.00	
Car Total		533,800.00	562	$89,643,600.00	546	$278,074,500.00	

Figure 8: PivotGrid with Frozen Headers

4.6 Grouping Bar

The PivotGrid Grouping Bar enables the drag and drop feature of fields between different areas such as column, row, value and filter. By using the Grouping Bar, you can add, rearrange, or remove fields to show data in the PivotGrid exactly the way you want.

The Grouping Bar has field headers that identify fields in the pivot grid. One field header contains:

- Caption string - identifies the field's content
- Sort indicator - identifies the sort order applied to the field's values

- Filter button - end-users can use it to filter field values

The headers of all visible fields are contained within header areas. The headers of row and column fields are displayed within the row header and column header areas, respectively. The headers of data fields are displayed within the data header area.

Use Case Scenarios

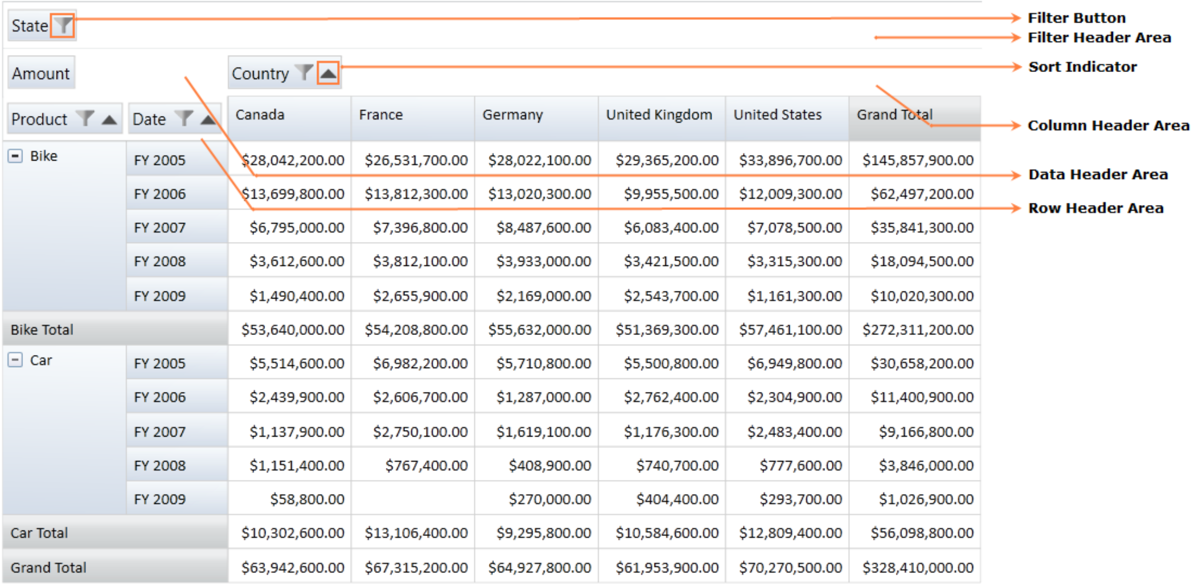
At times, you may expect the Grid to perform sorting and filtering at run-time.

Adding Grouping Bar

By default, Grouping Bar is enabled. It can be disabled by setting *ShowGroupBar* property of PivotGrid to *False*.

```
[C#]
// Instantiating PivotGridControl
PivotGridControl pivotGridControl1 = new PivotGridControl();
// Adding PivotRows
pivotGridControl1.PivotRows.Add(new PivotItem { FieldHeader = "Product"
});
pivotGridControl1.PivotColumns.Add(new PivotItem { FieldHeader = "Date"
});
// Adding PivotColumns
pivotGridControl1.PivotColumns.Add(new PivotItem { FieldHeader =
"Country" });
pivotGridControl1.PivotColumns.Add(new PivotItem { FieldHeader =
"State" });
// Adding PivotCalculations
pivotGridControl1.PivotCalculations.Add(new PivotComputationInfo {
FieldName="Amount" , Format="C"});
pivotGridControl1.PivotCalculations.Add(new PivotComputationInfo {
FieldName = "Quantity", Format = "#,##0" });
```

```
[VB]
' Instantiating PivotGridControl
Dim pivotGridControl1 As PivotGridControl = New PivotGridControl()
' Adding PivotRows
pivotGridControl1.PivotRows.Add(New PivotItem With {.FieldHeader =
"Product"})
pivotGridControl1.PivotColumns.Add(New PivotItem With {.FieldHeader =
"Date"})
' Adding PivotColumns
pivotGridControl1.PivotColumns.Add(New PivotItem With {.FieldHeader =
"Country"})
pivotGridControl1.PivotColumns.Add(New PivotItem With {.FieldHeader =
"State"})
' Adding PivotCalculations
pivotGridControl1.PivotCalculations.Add(New PivotComputationInfo With
{.FieldName="Amount", .Format="C"})
pivotGridControl1.PivotCalculations.Add(New PivotComputationInfo With
{.FieldName = "Quantity", .Format = "#,##0"})
```



Filter Button

Filter Header Area

Sort Indicator

Column Header Area

Data Header Area

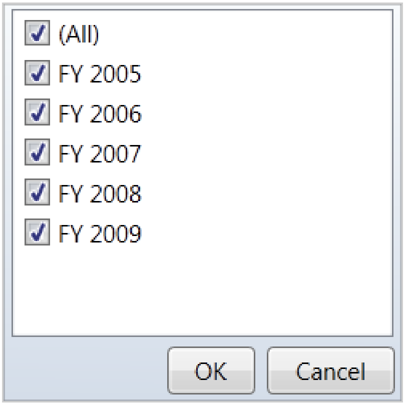
Row Header Area

		Canada	France	Germany	United Kingdom	United States	Grand Total
⊟ Bike	FY 2005	$28,042,200.00	$26,531,700.00	$28,022,100.00	$29,365,200.00	$33,896,700.00	$145,857,900.00
	FY 2006	$13,699,800.00	$13,812,300.00	$13,020,300.00	$9,955,500.00	$12,009,300.00	$62,497,200.00
	FY 2007	$6,795,000.00	$7,396,800.00	$8,487,600.00	$6,083,400.00	$7,078,500.00	$35,841,300.00
	FY 2008	$3,612,600.00	$3,812,100.00	$3,933,000.00	$3,421,500.00	$3,315,300.00	$18,094,500.00
	FY 2009	$1,490,400.00	$2,655,900.00	$2,169,000.00	$2,543,700.00	$1,161,300.00	$10,020,300.00
Bike Total		$53,640,000.00	$54,208,800.00	$55,632,000.00	$51,369,300.00	$57,461,100.00	$272,311,200.00
⊟ Car	FY 2005	$5,514,600.00	$6,982,200.00	$5,710,800.00	$5,500,800.00	$6,949,800.00	$30,658,200.00
	FY 2006	$2,439,900.00	$2,606,700.00	$1,287,000.00	$2,762,400.00	$2,304,900.00	$11,400,900.00
	FY 2007	$1,137,900.00	$2,750,100.00	$1,619,100.00	$1,176,300.00	$2,483,400.00	$9,166,800.00
	FY 2008	$1,151,400.00	$767,400.00	$408,900.00	$740,700.00	$777,600.00	$3,846,000.00
	FY 2009	$58,800.00		$270,000.00	$404,400.00	$293,700.00	$1,026,900.00
Car Total		$10,302,600.00	$13,106,400.00	$9,295,800.00	$10,584,600.00	$12,809,400.00	$56,098,800.00
Grand Total		$63,942,600.00	$67,315,200.00	$64,927,800.00	$61,953,900.00	$70,270,500.00	$328,410,000.00

Figure 9: PivotGrid Grouping Bar

4.6.1 Filtering in Grouping Bar

Data filtering displays only a subset of data that meets criteria specified by you and hides data that you don’t want to be displayed. The items present in the filter header area, column header area and row header area, provide the option of run-time filtering which is represented as a funnel symbol on it. On clicking the symbol, it opens a filter popup which displays a list of elements through which filtering can be applied.



☑ (All)
☑ FY 2005
☑ FY 2006
☑ FY 2007
☑ FY 2008
☑ FY 2009

Figure 10: Filter Popup

The following code example illustrates how to disable the filtering in the Grouping Bar.

```
[C#]
// Disabling Filtering
pivotGridControl1.AllowFiltering = false;
```

```
[VB]
// Disabling Filtering
pivotGridControl1.AllowFiltering = False
```

4.6.2 Sorting Indicator

The sort indicator in the item represents the sort type such as ascending order or descending order. By default, the PivotGrid will populate the data in ascending order. The sorting order can be changed by clicking on the item present in the row header area and column header area.

The following image illustrates the sort indicator with sort types.

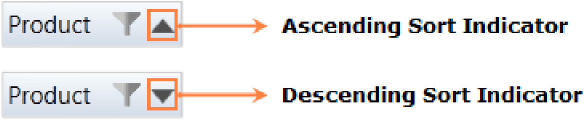


Figure 11: Sort Indicator

The following code example illustrates how to disable sorting in the Grouping Bar.

```
[C#]
// Disabling Sorting.
pivotGridControl1.AllowSorting = false;
```

```
[VB]
// Disabling Sorting.
pivotGridControl1.AllowSorting = False
```

4.7 Cell Selection

The PivotGrid for Windows Forms supports cell selection where you can select grid value cells similar to Microsoft Excel. On cell selection, an event called *PivotGridSelectionChanged* will be triggered and the *PivotGridSelectionChangedEventArgs* will return an *IEnumerable* collection of column, row and value of the corresponding selected cell.

Use Case Scenarios

Using the cell selection support, you can select the cells that can be copied to clipboard or notepad. You can perform custom operation on cell selection and also bind any control based on the selected cell values.

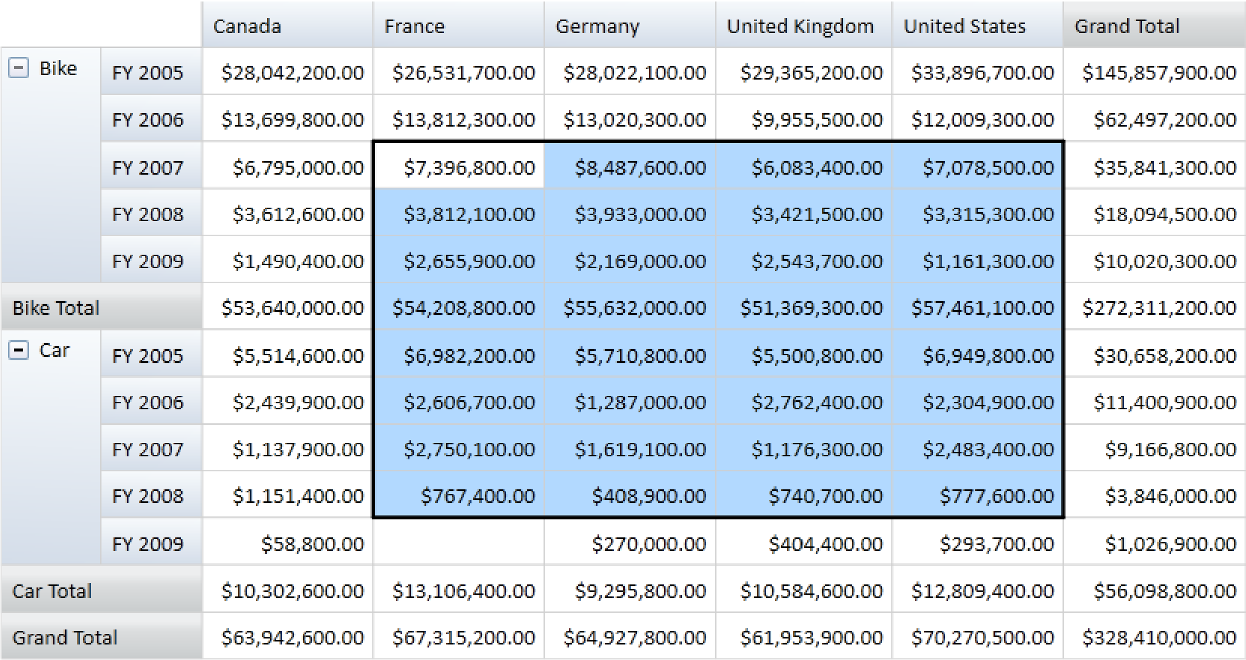
Adding Cell Selection

The following code snippets show how to create a PivotGrid and specify cell selection.

```
[C#]
// Instantiating PivotGridControl
PivotGridControl pivotGridControl1 = new PivotGridControl();
// Adding PivotRows
pivotGridControl1.PivotRows.Add(new PivotItem { FieldHeader = "Product"
});
pivotGridControl1.PivotColumns.Add(new PivotItem { FieldHeader = "Date"
});
// Adding PivotColumns
pivotGridControl1.PivotColumns.Add(new PivotItem { FieldHeader =
"Country" });

pivotGridControl1.PivotColumns.Add(new PivotItem { FieldHeader = 
"State" });

// Adding PivotCalculations

pivotGridControl1.PivotCalculations.Add(new PivotComputationInfo { 
FieldName="Amount" , Format="C"});

pivotGridControl1.PivotCalculations.Add(new PivotComputationInfo { 
FieldName = "Quantity", Format = "#,##0" });

// Enabling cell selection

this.pivotGridControl1.AllowSelection = false;
```

[VB]

```
' Instantiating PivotGridControl

Dim pivotGridControl1 As PivotGridControl = New PivotGridControl()

' Adding PivotRows

pivotGridControl1.PivotRows.Add(New PivotItem With {.FieldHeader = 
"Product"})

pivotGridControl1.PivotColumns.Add(New PivotItem With {.FieldHeader = 
"Date"})

' Adding PivotColumns

pivotGridControl1.PivotColumns.Add(New PivotItem With {.FieldHeader = 
"Country"})

pivotGridControl1.PivotColumns.Add(New PivotItem With {.FieldHeader = 
"State"})

' Adding PivotCalculations

pivotGridControl1.PivotCalculations.Add(New PivotComputationInfo With 
{.FieldName="Amount", .Format="C"})

pivotGridControl1.PivotCalculations.Add(New PivotComputationInfo With 
{.FieldName = "Quantity", .Format = "#,##0"})

' Enabling cell selection

Me.pivotGridControl1.AllowSelection = False
```



		Canada	France	Germany	United Kingdom	United States	Grand Total
⊟ Bike	FY 2005	$28,042,200.00	$26,531,700.00	$28,022,100.00	$29,365,200.00	$33,896,700.00	$145,857,900.00
	FY 2006	$13,699,800.00	$13,812,300.00	$13,020,300.00	$9,955,500.00	$12,009,300.00	$62,497,200.00
	FY 2007	$6,795,000.00	$7,396,800.00	$8,487,600.00	$6,083,400.00	$7,078,500.00	$35,841,300.00
	FY 2008	$3,612,600.00	$3,812,100.00	$3,933,000.00	$3,421,500.00	$3,315,300.00	$18,094,500.00
	FY 2009	$1,490,400.00	$2,655,900.00	$2,169,000.00	$2,543,700.00	$1,161,300.00	$10,020,300.00
Bike Total		$53,640,000.00	$54,208,800.00	$55,632,000.00	$51,369,300.00	$57,461,100.00	$272,311,200.00
⊟ Car	FY 2005	$5,514,600.00	$6,982,200.00	$5,710,800.00	$5,500,800.00	$6,949,800.00	$30,658,200.00
	FY 2006	$2,439,900.00	$2,606,700.00	$1,287,000.00	$2,762,400.00	$2,304,900.00	$11,400,900.00
	FY 2007	$1,137,900.00	$2,750,100.00	$1,619,100.00	$1,176,300.00	$2,483,400.00	$9,166,800.00
	FY 2008	$1,151,400.00	$767,400.00	$408,900.00	$740,700.00	$777,600.00	$3,846,000.00
	FY 2009	$58,800.00		$270,000.00	$404,400.00	$293,700.00	$1,026,900.00
Car Total		$10,302,600.00	$13,106,400.00	$9,295,800.00	$10,584,600.00	$12,809,400.00	$56,098,800.00
Grand Total		$63,942,600.00	$67,315,200.00	$64,927,800.00	$61,953,900.00	$70,270,500.00	$328,410,000.00

Figure 12: PivotGrid Cell Selection

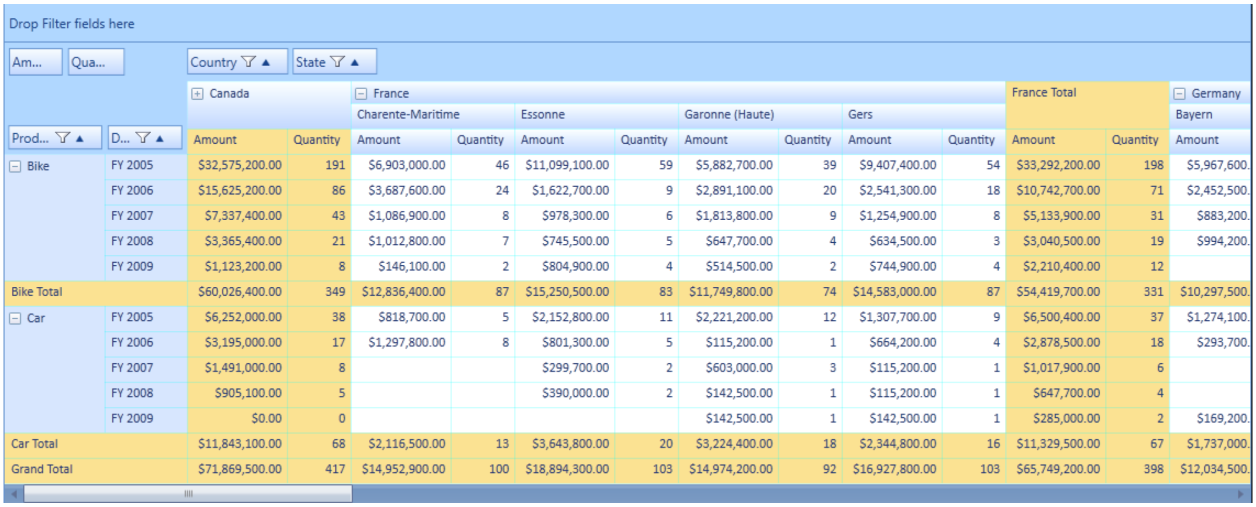
4.8 Subtotal Hiding

The subtotal hiding feature is used to show or hide the subtotals in the PivotGrid. In the case of larger data table, this feature enables the user to have an abstract view of the data by hiding subtotals using the *ShowSubTotals* property.

Use Case Scenarios

When the user has more computational fields with subtotals in each group of their PivotGrid, the user might find it difficult to view all the data. In that case, the user can hide the subtotals and make it visible when required.

The following screen shots shows the PivotGrid with shown and hidden sub totals.

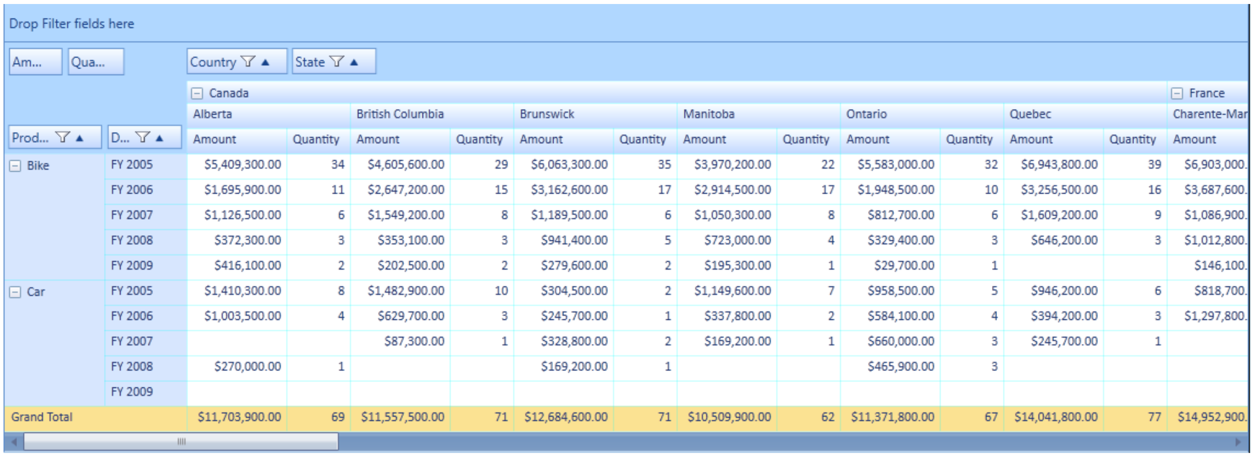


Drop Filter fields here

Am... | Qua... | Country ▲ | State ▲

		⊞ Canada		⊟ France								France Total		⊟ Germany
				Charente-Maritime		Essonne		Garonne (Haute)		Gers				Bayern
Prod... ▲	D... ▲	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount
⊟ Bike	FY 2005	$32,575,200.00	191	$6,903,000.00	46	$11,099,100.00	59	$5,882,700.00	39	$9,407,400.00	54	$33,292,200.00	198	$5,967,600.
	FY 2006	$15,625,200.00	86	$3,687,600.00	24	$1,622,700.00	9	$2,891,100.00	20	$2,541,300.00	18	$10,742,700.00	71	$2,452,500.
	FY 2007	$7,337,400.00	43	$1,086,900.00	8	$978,300.00	6	$1,813,800.00	9	$1,254,900.00	8	$5,133,900.00	31	$883,200.
	FY 2008	$3,365,400.00	21	$1,012,800.00	7	$745,500.00	5	$647,700.00	4	$634,500.00	3	$3,040,500.00	19	$994,200.
	FY 2009	$1,123,200.00	8	$146,100.00	2	$804,900.00	4	$514,500.00	2	$744,900.00	4	$2,210,400.00	12	
Bike Total		$60,026,400.00	349	$12,836,400.00	87	$15,250,500.00	83	$11,749,800.00	74	$14,583,000.00	87	$54,419,700.00	331	$10,297,500.
⊟ Car	FY 2005	$6,252,000.00	38	$818,700.00	5	$2,152,800.00	11	$2,221,200.00	12	$1,307,700.00	9	$6,500,400.00	37	$1,274,100.
	FY 2006	$3,195,000.00	17	$1,297,800.00	8	$801,300.00	5	$115,200.00	1	$664,200.00	4	$2,878,500.00	18	$293,700.
	FY 2007	$1,491,000.00	8			$299,700.00	2	$603,000.00	3	$115,200.00	1	$1,017,900.00	6	
	FY 2008	$905,100.00	5			$390,000.00	2	$142,500.00	1	$115,200.00	1	$647,700.00	4	
	FY 2009	$0.00	0					$142,500.00	1	$142,500.00	1	$285,000.00	2	$169,200.
Car Total		$11,843,100.00	68	$2,116,500.00	13	$3,643,800.00	20	$3,224,400.00	18	$2,344,800.00	16	$11,329,500.00	67	$1,737,000.
Grand Total		$71,869,500.00	417	$14,952,900.00	100	$18,894,300.00	103	$14,974,200.00	92	$16,927,800.00	103	$65,749,200.00	398	$12,034,500.

Figure 13: PivotGrid with Subtotals



Drop Filter fields here

Am... | Qua... | Country ▲ | State ▲

		⊟ Canada												⊟ France
		Alberta		British Columbia		Brunswick		Manitoba		Ontario		Quebec		Charente-Mar
Prod... ▲	D... ▲	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount
⊟ Bike	FY 2005	$5,409,300.00	34	$4,605,600.00	29	$6,063,300.00	35	$3,970,200.00	22	$5,583,000.00	32	$6,943,800.00	39	$6,903,000.
	FY 2006	$1,695,900.00	11	$2,647,200.00	15	$3,162,600.00	17	$2,914,500.00	17	$1,948,500.00	10	$3,256,500.00	16	$3,687,600.
	FY 2007	$1,126,500.00	6	$1,549,200.00	8	$1,189,500.00	6	$1,050,300.00	8	$812,700.00	6	$1,609,200.00	9	$1,086,900.
	FY 2008	$372,300.00	3	$353,100.00	3	$941,400.00	5	$723,000.00	4	$329,400.00	3	$646,200.00	3	$1,012,800.
	FY 2009	$416,100.00	2	$202,500.00	2	$279,600.00	2	$195,300.00	1	$29,700.00	1			$146,100.
⊟ Car	FY 2005	$1,410,300.00	8	$1,482,900.00	10	$304,500.00	2	$1,149,600.00	7	$958,500.00	5	$946,200.00	6	$818,700.
	FY 2006	$1,003,500.00	4	$629,700.00	3	$245,700.00	1	$337,800.00	2	$584,100.00	4	$394,200.00	3	$1,297,800.
	FY 2007			$87,300.00	1	$328,800.00	2	$169,200.00	1	$660,000.00	3	$245,700.00	1	
	FY 2008	$270,000.00	1			$169,200.00	1			$465,900.00	3			
	FY 2009													
Grand Total		$11,703,900.00	69	$11,557,500.00	71	$12,684,600.00	71	$10,509,900.00	62	$11,371,800.00	67	$14,041,800.00	77	$14,952,900.

Figure 14: PivotGrid with Subtotals Hidden

Properties

Table 9: Property Table

Property	Description	Data Type	Reference links
ShowSubTotals	Shows or hides the sub totals	Boolean	-

Methods

Table 10: Method Table

Method	Description	Parameters	Return Type	Reference links

SubTotalsRendering	Handles rendering of cells(showing or hiding the cells) by calculating the cell range values in the Pivot Engine based on the ShowSubTotals property value in the control	-	Void	-

Sample Link

Follow the steps given below to view a sample of this feature:

1. Select Start > Programs > Syncfusion > Essential Studio x.x.x.x -> Dashboard.
2. Click **Run Samples** under UI edition.
3. Select **PivotGrid.**
4. Navigate to **Selection** > **Cell Selection Demo.**

4.8.1 Showing or Hiding Subtotals in PivotGrid

The user can show or hide the PivotGrid subtotals using *ShowSubTotals* property. To show sub totals, set this property to true. To hide subtotals, set this property to false. By default the value of the *ShowSubTotals* property is set to true.

The following code example illustrates how to set values for the *ShowSubTotals* property to show the subtotals.

```
[C#]

this.pivotGridControl1.ShowSubTotals = true;
```

```
[VB]

Me.pivotGridControl1.ShowSubTotals = True
```

The following code example illustrates how to set values for the *ShowSubTotals* property to hide the subtotals.

```
[C#]


this.pivotGridControl1.ShowSubTotals = false;
```

```
[VB]


Me.pivotGridControl1.ShowSubTotals = False
```

4.9 Exporting Pivot Grid to Excel, Word, and PDF

The PivotGrid is exported into different formats and the formatting styles are applied as per the visual style of PivotGrid.

There are three options to export PivotGrid:

1. Export to Excel
2. Export to Word
3. Export to PDF

Export to Excel:

Exporting data to an Excel spreadsheet is one of the most commonly preferred features in the .NET world. Essential Grid control has an in-built support to export an Excel spreadsheet. The class **ExcelExport** provides support for exporting data from the PivotGrid control to an Excel spreadsheet for verification and/or computation.

This class automatically copies a grid's styles and formats to an Excel spreadsheet. This enables you to export PivotGrid to an Excel document.

The following code illustrates exporting PivotGrid to an Excel:

```
ExcelExport excelExport = new ExcelExport(pivotGridControl1,
Syncfusion.XlsIO.ExcelVersion.Excel2010);

excelExport.ExportMode = (ExportAsPivotTable) ? ExportModes.PivotTable
: ExportModes.Cell;

excelExport.Export(FileName);
```

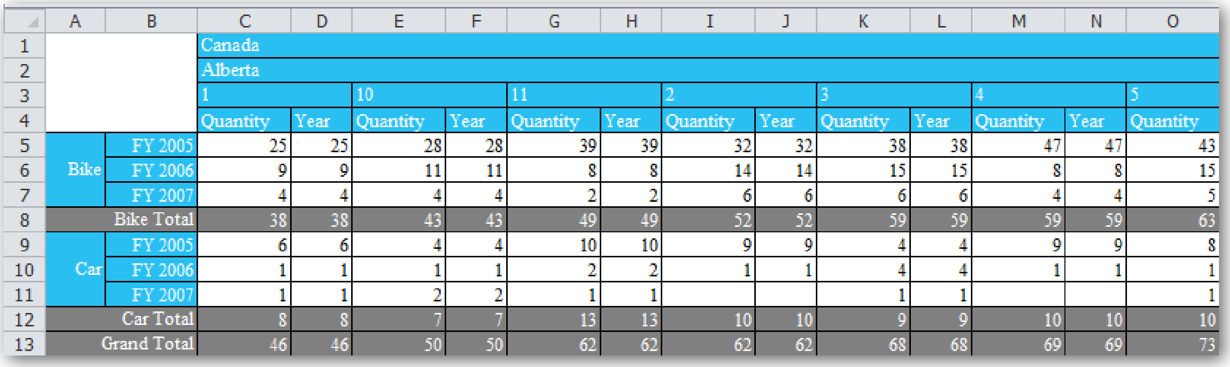
Export to Excel provides two options:

1. Cell-by-Cell Export
2. Pivot Table Export

Cell-by-Cell Export:

In the cell-by-cell export, the contents are exported cell by cell with the formats applied on it. The cell-by-cell formatting will have all the styles applied on it.

The following image shows the cell-by-cell export:



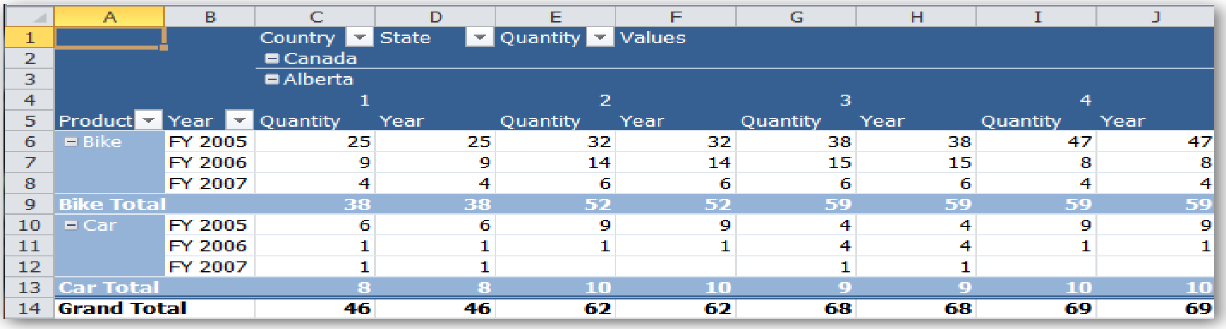
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1			Canada												
2			Alberta												
3			1		10		11		2		3		4		5
4			Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year	Quantity
5	Bike	FY 2005	25	25	28	28	39	39	32	32	38	38	47	47	43
6		FY 2006	9	9	11	11	8	8	14	14	15	15	8	8	15
7		FY 2007	4	4	4	4	2	2	6	6	6	6	4	4	5
8	Bike Total		38	38	43	43	49	49	52	52	59	59	59	59	63
9	Car	FY 2005	6	6	4	4	10	10	9	9	4	4	9	9	8
10		FY 2006	1	1	1	1	2	2	1	1	4	4	1	1	1
11		FY 2007	1	1	2	2	1	1			1	1			1
12	Car Total		8	8	7	7	13	13	10	10	9	9	10	10	10
13	Grand Total		46	46	50	50	62	62	62	62	68	68	69	69	73

Pivot Table Export:

In this type of export, the user can export the entire PivotGrid with its functionalities such as sorting and filtering. PivotGrid pivots the data via drag-and-drop to organize the data in a cross-tabulated form. The major advantage of PivotGrid is that you can extract any desired information within a limited span of time.

Apart from presenting the data in a proper manner, you can also summarize and group the data. PivotGrid has its main application in the financial domain. It is used to organize and analyze the business data.

The following image depicts the exported PivotTable:



	A	B	C	D	E	F	G	H	I	J
1			Country ▾	State ▾	Quantity ▾	Values				
2			⊟ Canada							
3			⊟ Alberta							
4			1		2		3		4	
5	Product ▾	Year ▾	Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year
6	⊟ Bike	FY 2005	25	25	32	32	38	38	47	47
7		FY 2006	9	9	14	14	15	15	8	8
8		FY 2007	4	4	6	6	6	6	4	4
9	Bike Total		38	38	52	52	59	59	59	59
10	⊟ Car	FY 2005	6	6	9	9	4	4	9	9
11		FY 2006	1	1	1	1	4	4	1	1
12		FY 2007	1	1			1	1		
13	Car Total		8	8	10	10	9	9	10	10
14	Grand Total		46	46	62	62	68	68	69	69

Export to Word:

Essential Grid has an in-built support to export Word. The user can download the data from the PivotGrid control into a Word document for offline verification and/or computation. This can be achieved by making use of the **PivotWordExport** `class`.

The below code snippets contains codes that exports PivotGrid to Word:

```
[C#]
```

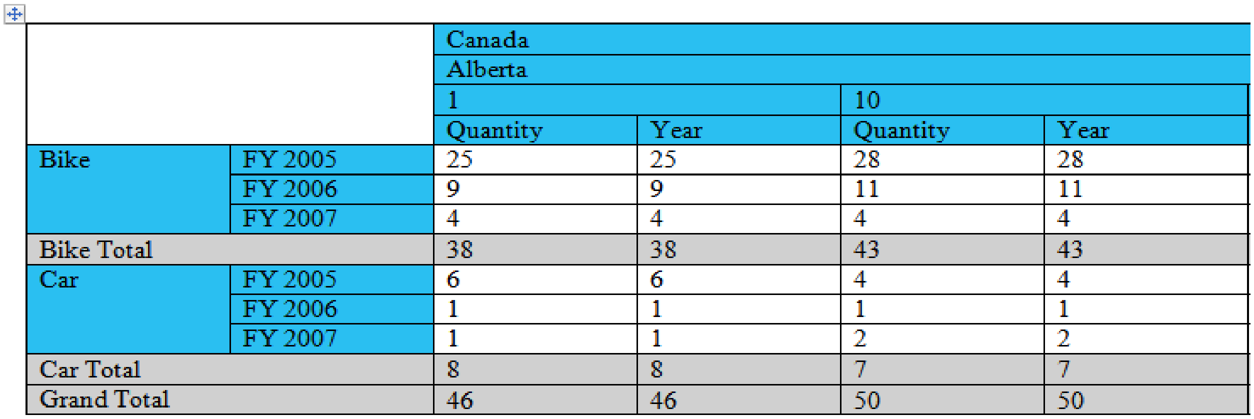
```
PivotWordExport wordExport = new PivotWordExport();

wordExport.pivotGridToWord(savedialog.FileName, pivotGridControl1);
```

```
[VB]

Dim wordExport As New PivotWordExport()

wordExport.pivotGridToWord(savedialog.FileName, pivotGridControl1)
```

Merging is applied to all the cells and the exported file is as same as that of the original PivotGrid.

The formatting is applied based on the visual styles of the grid.

The below image depicts the conversion of PivotGrid content to a Word file:



		Canada			
		Alberta			
		1		10	
		Quantity	Year	Quantity	Year
Bike	FY 2005	25	25	28	28
	FY 2006	9	9	11	11
	FY 2007	4	4	4	4
Bike Total		38	38	43	43
Car	FY 2005	6	6	4	4
	FY 2006	1	1	1	1
	FY 2007	1	1	2	2
Car Total		8	8	7	7
Grand Total		46	46	50	50

Exporting to PDF:

Essential Grid control supports conversion of PivotGrid content to a PDF file. Data in the PivotGrid control can be converted to a PDF document for offline verification and/or computation. This can be achieved by making use of the **PivotPdfExport** class. The PDF libraries are used to support the conversion of PivotGrid content to a PDF page.

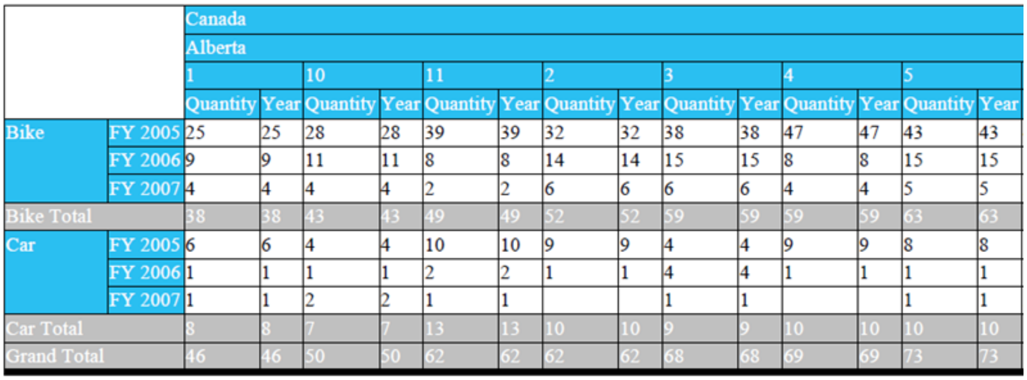
While exporting to PDF, PivotGrid is read row by row and exported into the PDF document.

The **Export** method is used to export PivotGrid content to a PDF file. The following sample code illustrates how to convert the PivotGrid content to PDF.

```
[C#]
PivotPdfExport pdfExport = new PivotPdfExport(pivotGridControl1);
pdfExport.Export(savedialog.FileName);
```

```
[VB]
Dim pdfExport As New PivotPdfExport(pivotGridControl1)
pdfExport.Export(savedialog.FileName)
```

The below image depicts the conversion of PivotGrid content to a PDF file:



		Canada													
		Alberta													
		1		10		11		2		3		4		5	
		Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year	Quantity	Year
Bike	FY 2005	25	25	28	28	39	39	32	32	38	38	47	47	43	43
	FY 2006	9	9	11	11	8	8	14	14	15	15	8	8	15	15
	FY 2007	4	4	4	4	2	2	6	6	6	6	4	4	5	5
Bike Total		38	38	43	43	49	49	52	52	59	59	59	59	63	63
Car	FY 2005	6	6	4	4	10	10	9	9	4	4	9	9	8	8
	FY 2006	1	1	1	1	2	2	1	1	4	4	1	1	1	1
	FY 2007	1	1	2	2	1	1			1	1			1	1
Car Total		8	8	7	7	13	13	10	10	9	9	10	10	10	10
Grand Total		46	46	50	50	62	62	62	62	68	68	69	69	73	73

Sample Location

A sample is placed in the following location:

<Installed Location>\Syncfusion\EssentialStudio\[Version Number]\Windows\PivotGrid.Windows\Samples\2.0\Exporting\Export Demo

4.10 Print Option

The print option is extended for the PivotGrid control to allow users to preview the contents before the contents are printed on paper.

This feature is used to print the PivotGrid control in landscape and portrait views. This feature has overridden the GridPrintDocumentAdv from Syncfusion.GridHelperClasses.Windows.

The pivot grid visual style color is automatically applied in the printed document based on the visual styles of the grid.

The print functionality can be invoked using the following code:

```
 [C#]

private void button1_Click_1(object sender, EventArgs e)
{

 try
   {
PivotGridPrintDocumentAdv pd = new
PivotGridPrintDocumentAdv(this.pivotGridControl1);
pd.DefaultPageSettings.Margins = new
System.Drawing.Printing.Margins(25, 25, 25, 25);
PrintPreviewDialog previewDialog = new PrintPreviewDialog();
previewDialog.Document = pd;
previewDialog.Show();
    }

catch (Exception ex)
    {
MessageBox.Show("Error while print preview" + ex.ToString());
    }

}
```

```
[VB]

Private Sub button1_Click_1(ByVal sender As Object, ByVal e As
EventArgs)

 Try

Dim pd As New PivotGridPrintDocumentAdv(Me.pivotGridControl1)

pd.DefaultPageSettings.Margins = New
System.Drawing.Printing.Margins(25, 25, 25, 25)
Dim previewDialog As New PrintPreviewDialog()
previewDialog.Document = pd
previewDialog.Show()

Catch ex As Exception
MessageBox.Show("Error while print preview" & ex.ToString())

End Try

End Sub
```

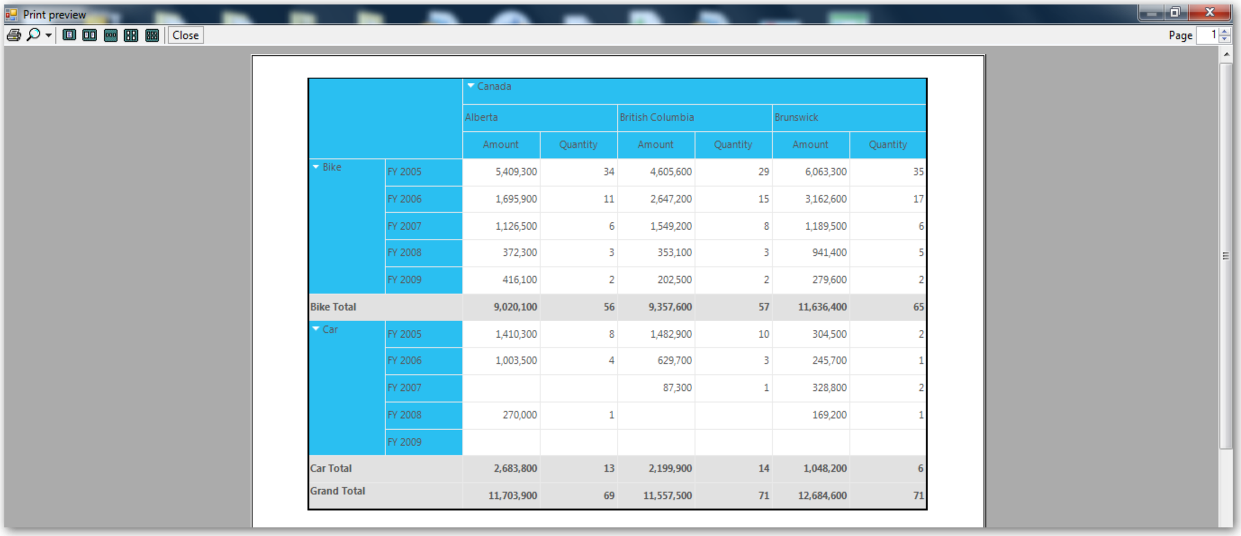
Headers and footers can also be added by using the **DrawGridPrintHeader** and **DrawGridPrintFooter** events. The following code illustrates how to add the header and footer.

```
[C#]
pd.DrawGridPrintHeader+=new
GridPrintDocumentAdv.DrawGridHeaderFooterEventHandler(pd_DrawGridPrintH
eader);
```

```
pd.DrawGridPrintFooter+=new
GridPrintDocumentAdv.DrawGridHeaderFooterEventHandler(pd_DrawGridPrintF
ooter);
```

```
[VB]

AddHandler pd.DrawGridPrintHeader, AddressOf pd_DrawGridPrintHeader

AddHandler pd.DrawGridPrintFooter, AddressOf pd_DrawGridPrintFooter
```

The following image shows the printed output of the pivot grid:



		Canada					
		Alberta		British Columbia		Brunswick	
		Amount	Quantity	Amount	Quantity	Amount	Quantity
Bike	FY 2005	5,409,300	34	4,605,600	29	6,063,300	35
	FY 2006	1,695,900	11	2,647,200	15	3,162,600	17
	FY 2007	1,126,500	6	1,549,200	8	1,189,500	6
	FY 2008	372,300	3	353,100	3	941,400	5
	FY 2009	416,100	2	202,500	2	279,600	2
Bike Total		9,020,100	56	9,357,600	57	11,636,400	65
Car	FY 2005	1,410,300	8	1,482,900	10	304,500	2
	FY 2006	1,003,500	4	629,700	3	245,700	1
	FY 2007			87,300	1	328,800	2
	FY 2008	270,000	1			169,200	1
	FY 2009						
Car Total		2,683,800	13	2,199,900	14	1,048,200	6
Grand Total		11,703,900	69	11,557,500	71	12,684,600	71

Figure 15: Pivot Grid in Print Preview

Sample Link

A sample is available in the following location:

<Installed Location>\Syncfusion\EssentialStudio\[Version Number]\Windows\PivotGrid.Windows \Samples\2.0\Print\Print Grid Demo

Index

1

2

3

4