Sparklines in DataGrids

Overview

A sparkline is a very small line chart, typically drawn without axes or coordinates. In a simple and highly condensed way, sparklines represent the general shape of a variation, typically over time, in a desired measurement, such as the temperature or the stock market price.

Sparklines are small enough to be embedded in text, or several sparklines can be combined together as elements of a small group.

Typical charts are designed to show as much data as possible and are set off from the flow of text. In contrast, sparklines are intended to be succinct, memorable, and located directly in the text.

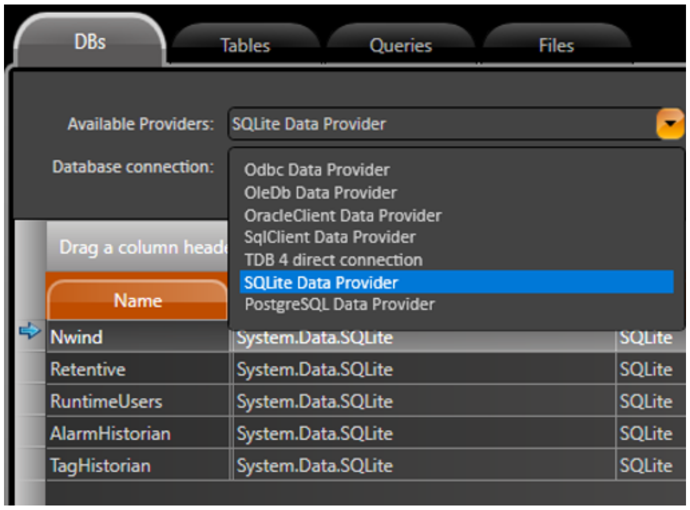
Requirements and Procedures

Below you will find the steps to implement a Sparkline in your project.

1. Create a database for the numeric information that will be displayed.
2. Configure the tables and queries that will be used in the data grid.
3. Configure the Draw Environment and implement the necessary code on CodeBehind.

Database

In *Edit>Datasets>DBs,* you need to include the external database by using one of the providers available in the *Available Providers* list. In this example, we are going to use a SQLite provider in a database called *Nwind*.



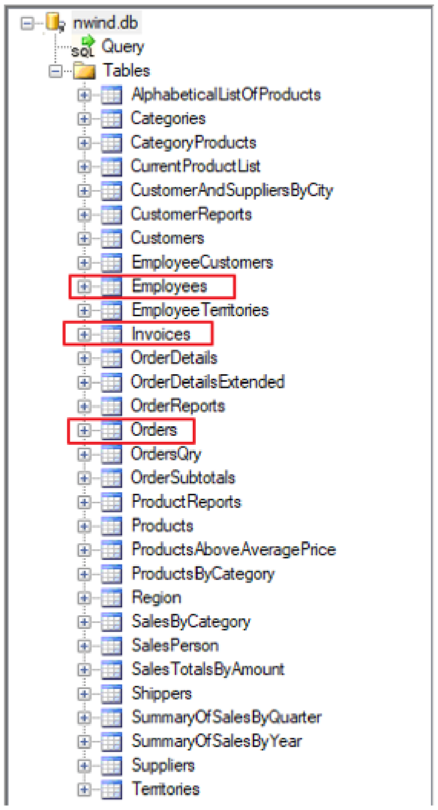
Available Providers: SQLite Data Provider

Database connection:

- Odbc Data Provider
- OleDb Data Provider
- OracleClient Data Provider
- SqlClient Data Provider
- TDB 4 direct connection
- SQLite Data Provider
- PostgreSQL Data Provider

Name		
Nwind	System.Data.SQLite	SQLite
Retentive	System.Data.SQLite	SQLite
RuntimeUsers	System.Data.SQLite	SQLite
AlarmHistorian	System.Data.SQLite	SQLite
TagHistorian	System.Data.SQLite	SQLite

This example database has many tables. However, not all of the tables will be used in the project. The image below shows all the available tables and highlights the ones that will be used.

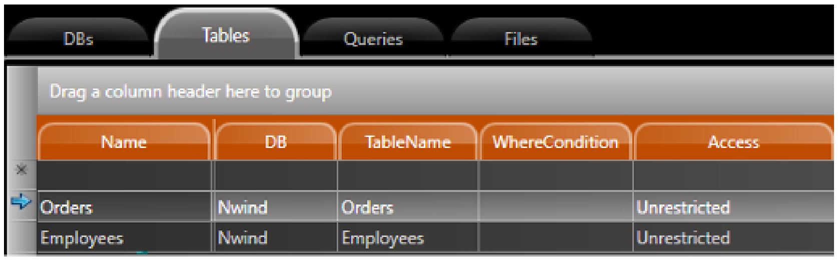


```
nwind.db
  Query
  Tables
    AlphabeticalListOfProducts
    Categories
    CategoryProducts
    CurrentProductList
    CustomerAndSuppliersByCity
    CustomerReports
    Customers
    EmployeeCustomers
    Employees
    EmployeeTerritories
    Invoices
    OrderDetails
    OrderDetailsExtended
    OrderReports
    Orders
    OrdersQry
    OrderSubtotals
    ProductReports
    Products
    ProductsAboveAveragePrice
    ProductsByCategory
    Region
    SalesByCategory
    SalesPerson
    SalesTotalsByAmount
    Shippers
    SummaryOfSalesByQuarter
    SummaryOfSalesByYear
    Suppliers
    Territories
```

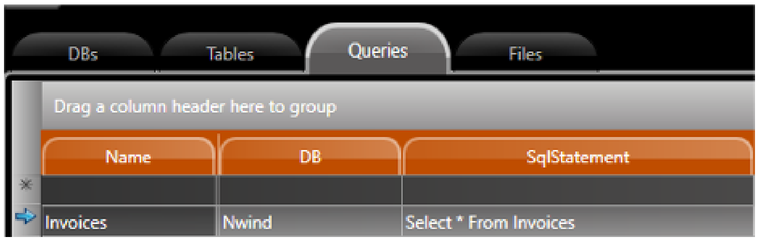
Tables and Queries

In *Edit>Dataset>Tables/Queries,* you can create internal tables and queries for internal usage.

The tables must be mapped to tables that already exist in the database, as explained in the previous section. Each query is defined for a specific database and may contain a predefined SQL Statement.



Name	DB	TableName	WhereCondition	Access
Orders	Nwind	Orders		Unrestricted
Employees	Nwind	Employees		Unrestricted

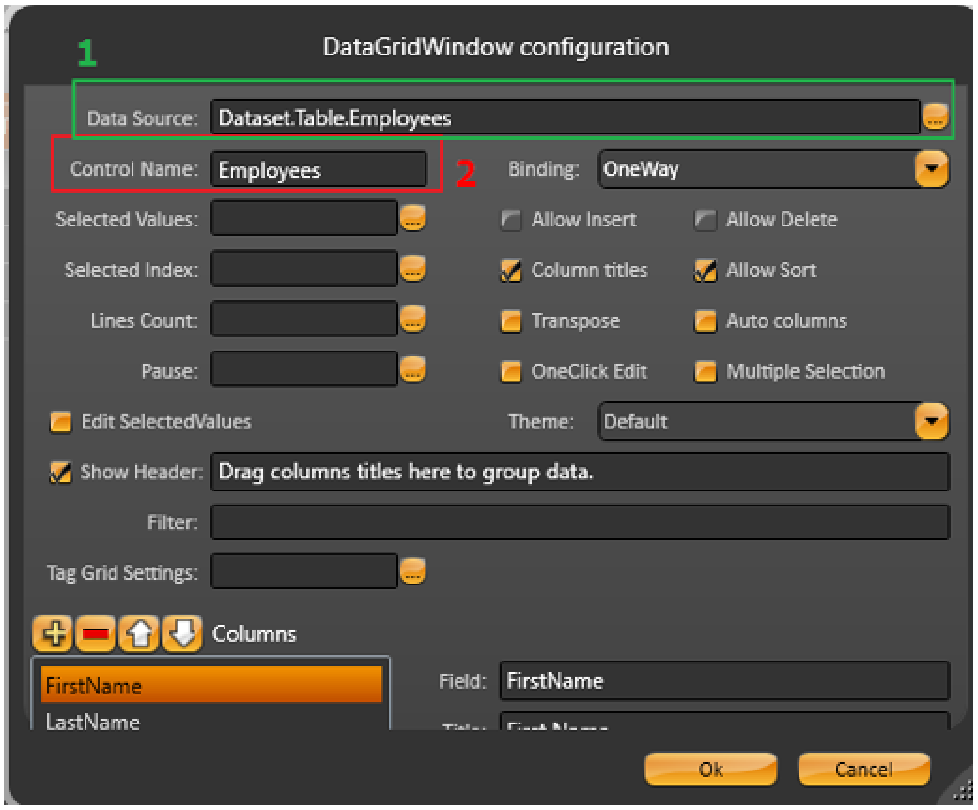


Name	DB	SqlStatement
Invoices	Nwind	Select * From Invoices

Project Configuration

After setting up the tables and queries that you will use, you need to configure one of the tables as the *DataSource* in the *DataGridWindow* configuration (1).

To manipulate a DataGrid control in CodeBehind, the best approach is to use the *Control Name* field (2) for the configuration dialog.


DataGridWindow configuration

1

Data Source: Dataset.Table.Employees

Control Name: Employees

2

Binding: OneWay

Selected Values:

Selected Index:

Lines Count:

Pause:

Allow Insert

Allow Delete

Column titles

Allow Sort

Transpose

Auto columns

OneClick Edit

Multiple Selection

Edit SelectedValues

Theme: Default

Show Header: Drag columns titles here to group data.

Filter:

Tag Grid Settings:

Columns

FirstName

LastName

Field: FirstName

Ok

Cancel

Then, you can use that given name (2) to get control of the element and change its properties. The below is an example.

```
TDataGridWindow grid = CurrentDisplay.GetDataGrid("Employees");
grid.GridControl.SetColumnSparkline("Orders", this.SparklineCollectionDelegate, style);
// SetColumnSparkline parameters - fieldName (string)
//                          - populateCallback (SparklineCollectionDelegate)
//                          - style (string)
```

A delegate is an object which refers to a method or you can consider it as a variable that holds a reference to a method. It provides a way to know which method is called when an event is triggered.

⚠ A delegate will only call a method that agrees with its signature and return type. A method can be either a static method associated with a class, or it can be an instance method associated with an object.

Below is a callback. The system will always look for a code EXACTLY like this one to create the Sparkline. So if you want to use this feature, you MUST have this code on your code behind.

```
Dictionary<int, ArgumentValue[]> mapEmployeeToOrders = null;
private ArgumentValue[] SparklineCollectionDelegate(string fieldName, System.Data.DataRow row)
{
ArgumentValue[] sourceCollection;
int  emplID  =  TK.To< int>(row["EmployeeID"]);
if (!this.mapEmployeeToOrders.TryGetValue(emplID, out sourceCollection))
       sourceCollection = new ArgumentValue[0];
return sourceCollection;
}
```

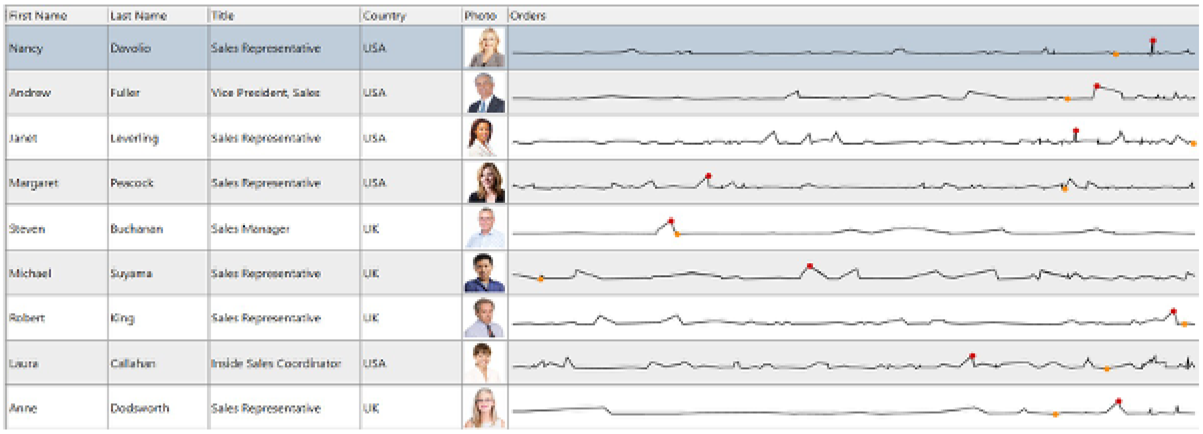
A sparkline is updated following the logic seen in the sample code below:

```
//         Get data from tables         // DataTable employees =  @Dataset.Table.Employees.SelectCommand();
DataTable orders = @Dataset t.Table.Orders.SelectCommand();
DataTable invoices = @Dataset.Query.Invoices.SelectCommand();
//         //

this.SetSparkline(style); // Set Style to Sparkline this.mapEmployeeToOrders = new Dictionary<int, ArgumentValue
[]>();
foreach (DataRow empl in employees.Rows)
{
int  emplID  =  TK.To<int>(empl["EmployeeID"]);
ist<ArgumentValue> sourceCollection = new List<ArgumentValue>();
foreach (DataRow order in orders.Select("EmployeeID = " + emplID))
{
int  orderID  =  TK.To< int>(order["OrderID"]); ArgumentValue av = new ArgumentValue();
av.Argument  =  order["OrderDate"];  //  x-axis  for  sparkline
foreach (DataRow invoice in invoices.Select("OrderID = " + orderID))
{
double value = TK.To< double >(av.Value);
double  quantity  =  TK.To< double >(invoice["Quantity"]);
double  unitPrice  =  TK.To< double >(invoice["UnitPrice"]);
av.Value = value + (quantity * unitPrice); // y-axis for sparkline
}
sourceCollection.Add(av);
}
//  map  values  (y-axis)  and  order  date  (x-axis)  to  employee  ID this.mapEmployeeToOrders[emplID] =
sourceCollection.ToArray();
}
```

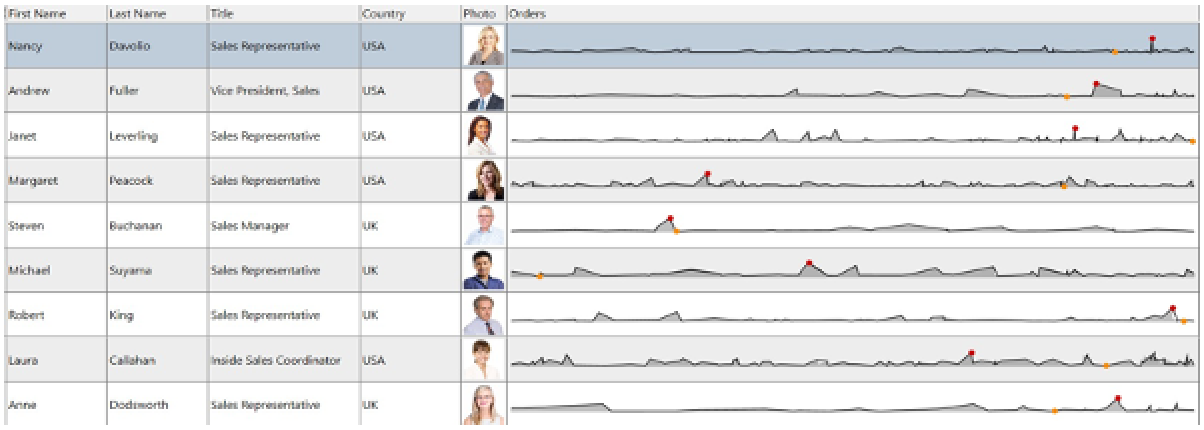
The available Sparkline styles are:

- **Line**



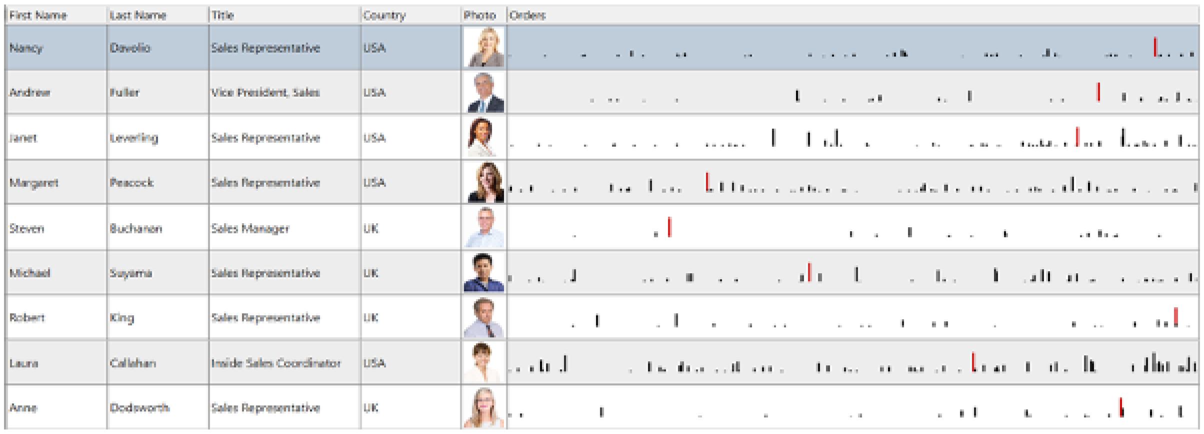
First Name	Last Name	Title	Country	Photo	Orders
Nancy	Davolio	Sales Representative	USA		
Andrew	Fuller	Vice President, Sales	USA		
Janet	Leverling	Sales Representative	USA		
Margaret	Peacock	Sales Representative	USA		
Steven	Buchanan	Sales Manager	UK		
Michael	Suyama	Sales Representative	UK		
Robert	King	Sales Representative	UK		
Laura	Callahan	Inside Sales Coordinator	USA		
Anne	Dodsworth	Sales Representative	UK		

- **Area**



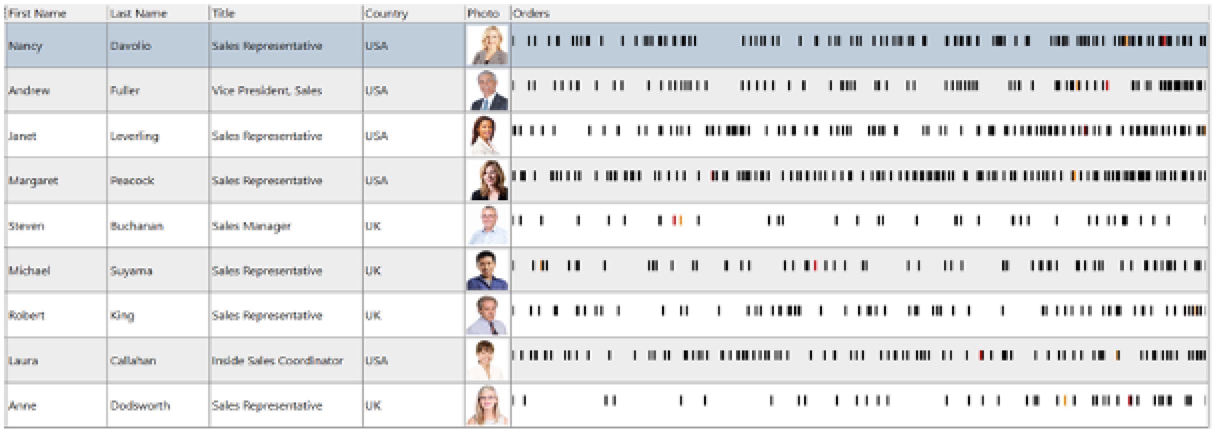
First Name	Last Name	Title	Country	Photo	Orders
Nancy	Davolio	Sales Representative	USA		
Andrew	Fuller	Vice President, Sales	USA		
Janet	Leverling	Sales Representative	USA		
Margaret	Peacock	Sales Representative	USA		
Steven	Buchanan	Sales Manager	UK		
Michael	Suyama	Sales Representative	UK		
Robert	King	Sales Representative	UK		
Laura	Callahan	Inside Sales Coordinator	USA		
Anne	Dodsworth	Sales Representative	UK		

- **Bar**



First Name	Last Name	Title	Country	Photo	Orders
Nancy	Davolio	Sales Representative	USA		
Andrew	Fuller	Vice President, Sales	USA		
Janet	Leverling	Sales Representative	USA		
Margaret	Peacock	Sales Representative	USA		
Steven	Buchanan	Sales Manager	UK		
Michael	Suyama	Sales Representative	UK		
Robert	King	Sales Representative	UK		
Laura	Callahan	Inside Sales Coordinator	USA		
Anne	Dodsworth	Sales Representative	UK		

- **WinLoss**



First Name	Last Name	Title	Country	Photo	Orders
Nancy	Davolio	Sales Representative	USA		
Andrew	Fuller	Vice President, Sales	USA		
Janet	Leverling	Sales Representative	USA		
Margaret	Peacock	Sales Representative	USA		
Steven	Buchanan	Sales Manager	UK		
Michael	Suyama	Sales Representative	UK		
Robert	King	Sales Representative	UK		
Laura	Callahan	Inside Sales Coordinator	USA		
Anne	Dodsworth	Sales Representative	UK		

Values are represented by bars that either grow up or grow down from an invisible line.