



# Creating Daily Schedule Reports using SharpShooter Reports (Part I)

---

*By Maxim Edapin, a software developer at Perpetuum Software LLC*

## Introduction:

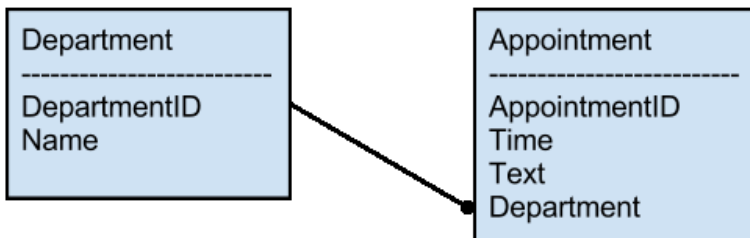
In the first part of the articles series devoted to the creation of various daily schedules we will start from building a simple Windows Forms application that will display the reports we create. And we will also create our first report - a schedule of appointments for every department. The appointments will be displayed one by one and sorted by time.

## Prerequisites:

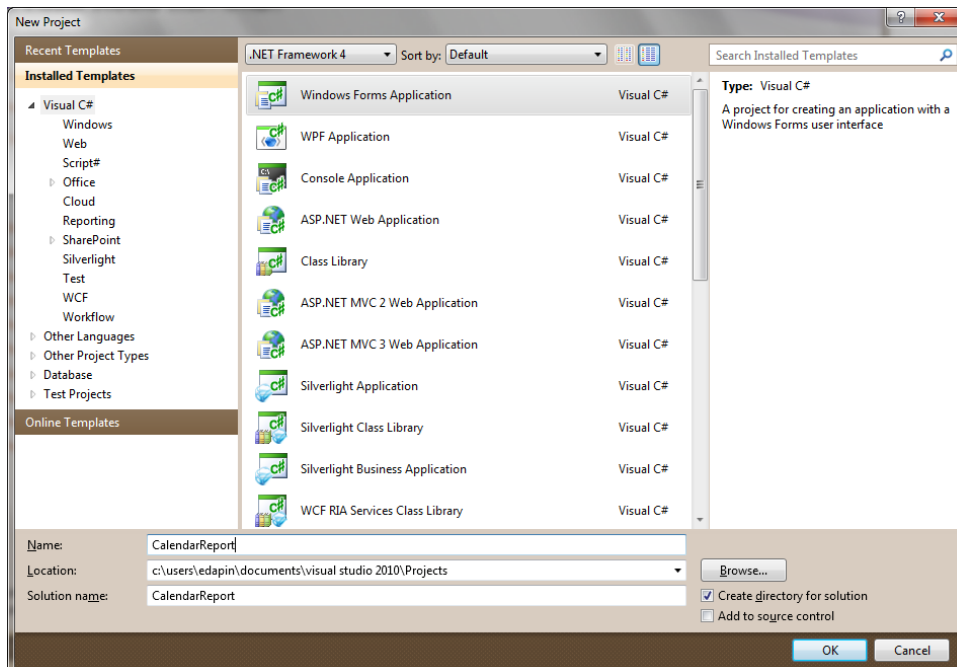
- .NET Framework 2.0, 3.5 or 4.0
- Visual Studio 2005/2008/2010
- SharpShooter Reports 6.3 or higher

## Creating a Windows Forms Application:

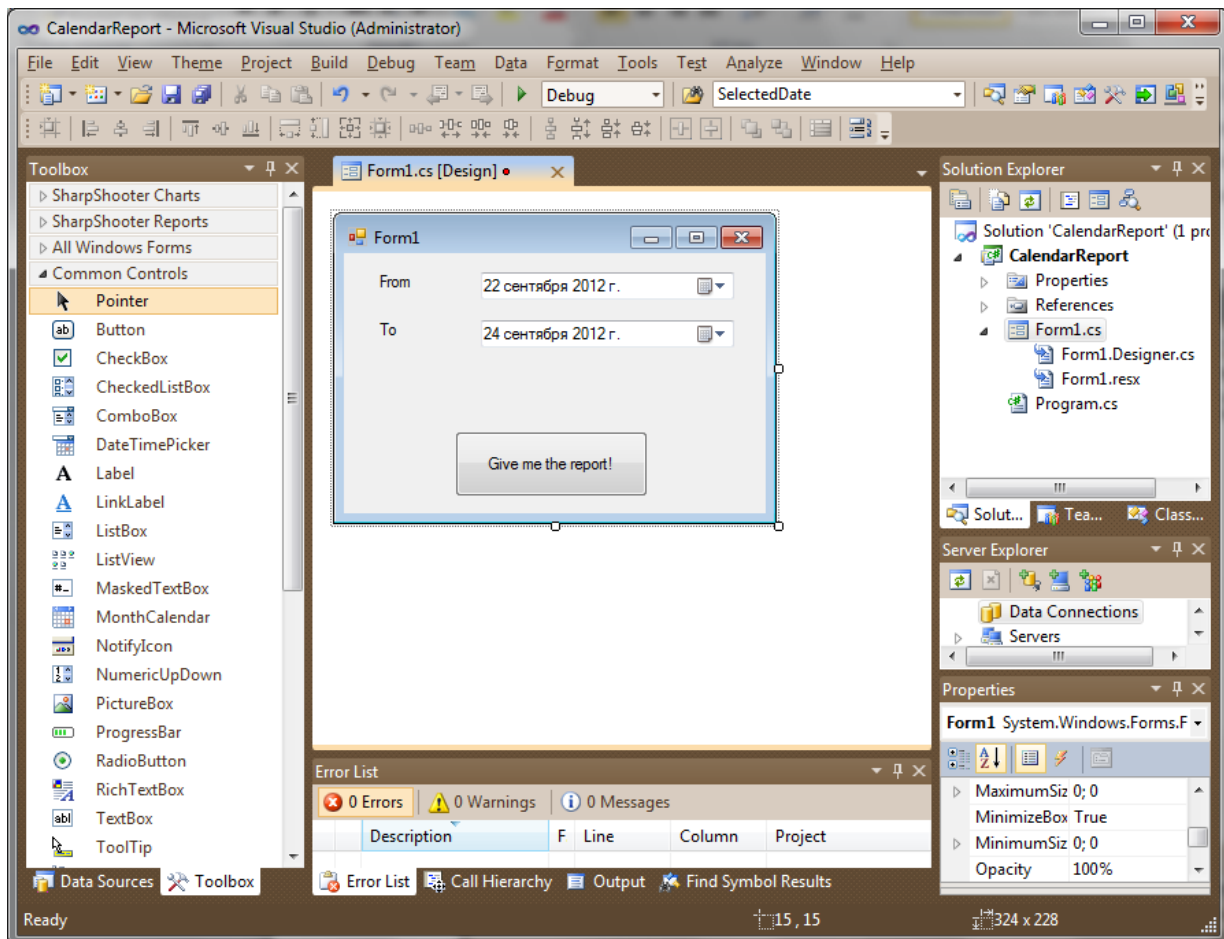
For the start, let's assume that we have the following database structure:



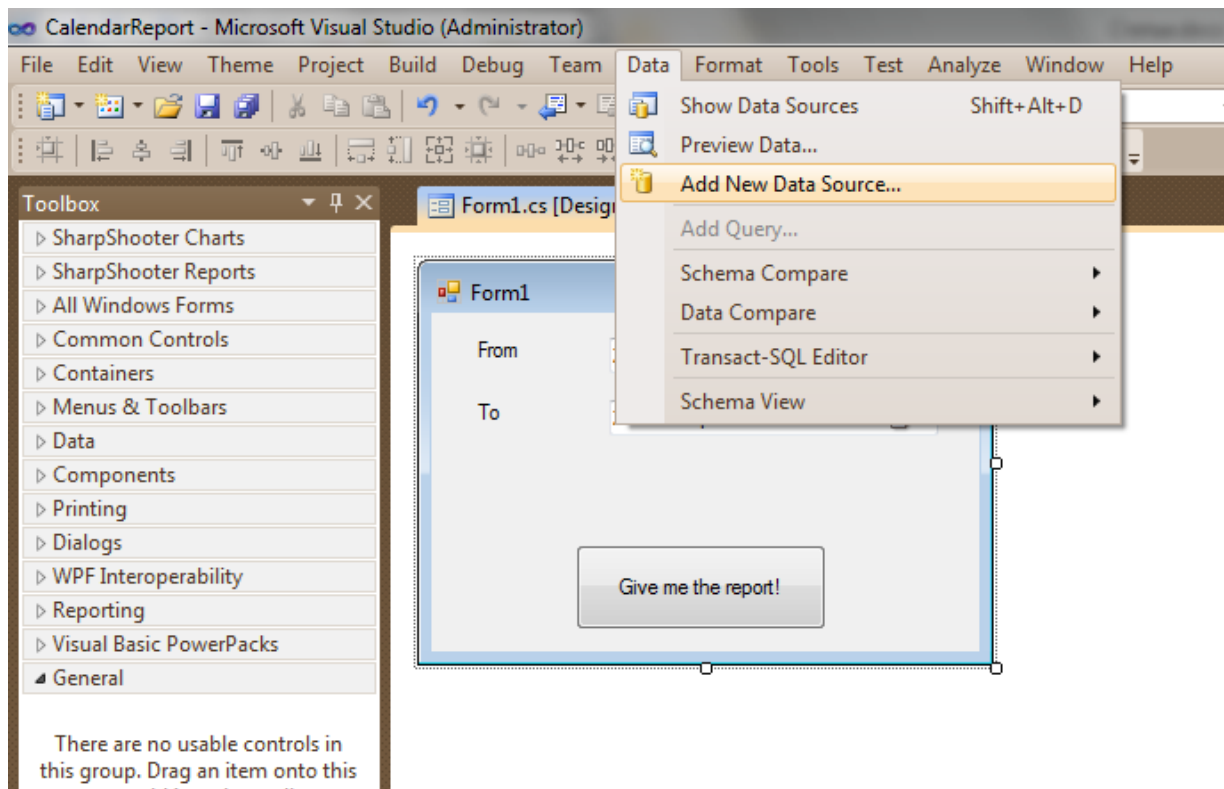
Now we create a new WindowsFormsApplication project.

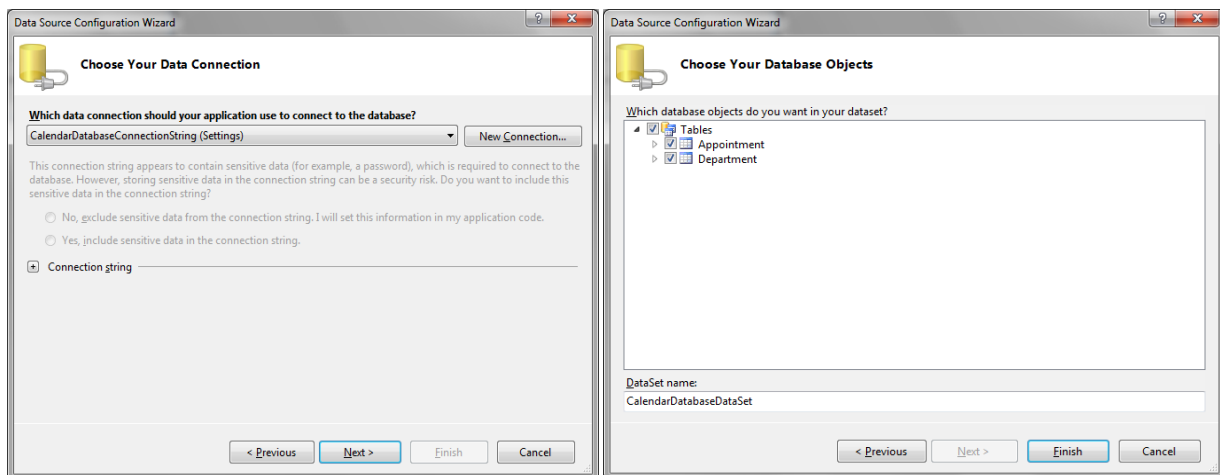
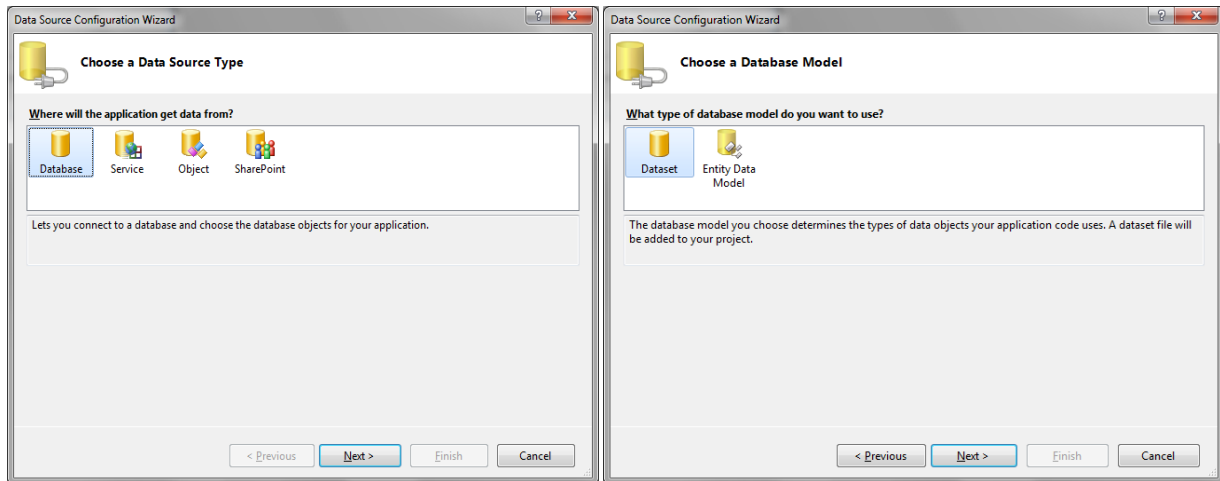


Add a couple of controls: two DatePicker elements for setting period of time for the schedule, and one button to run report generation.

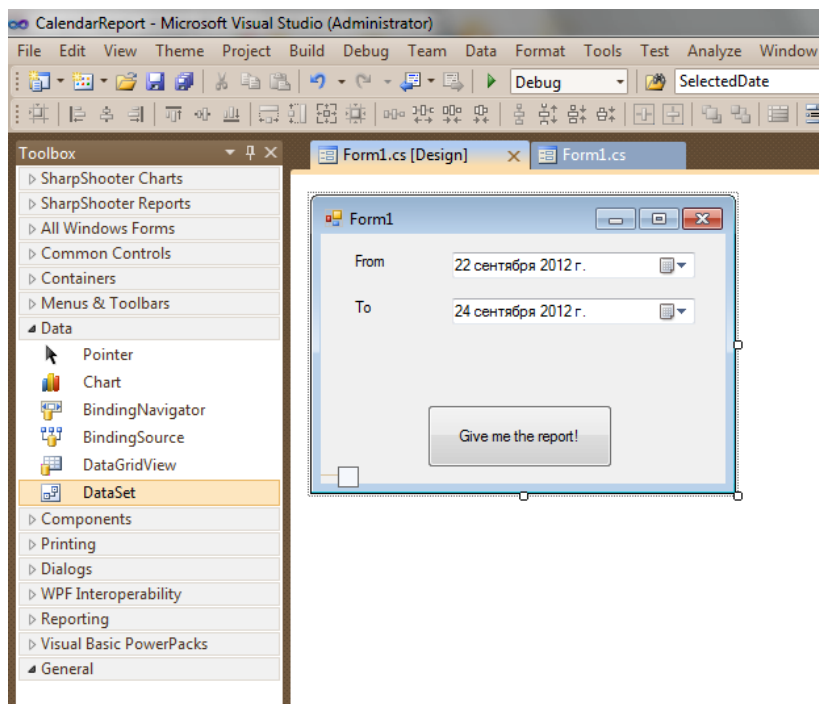


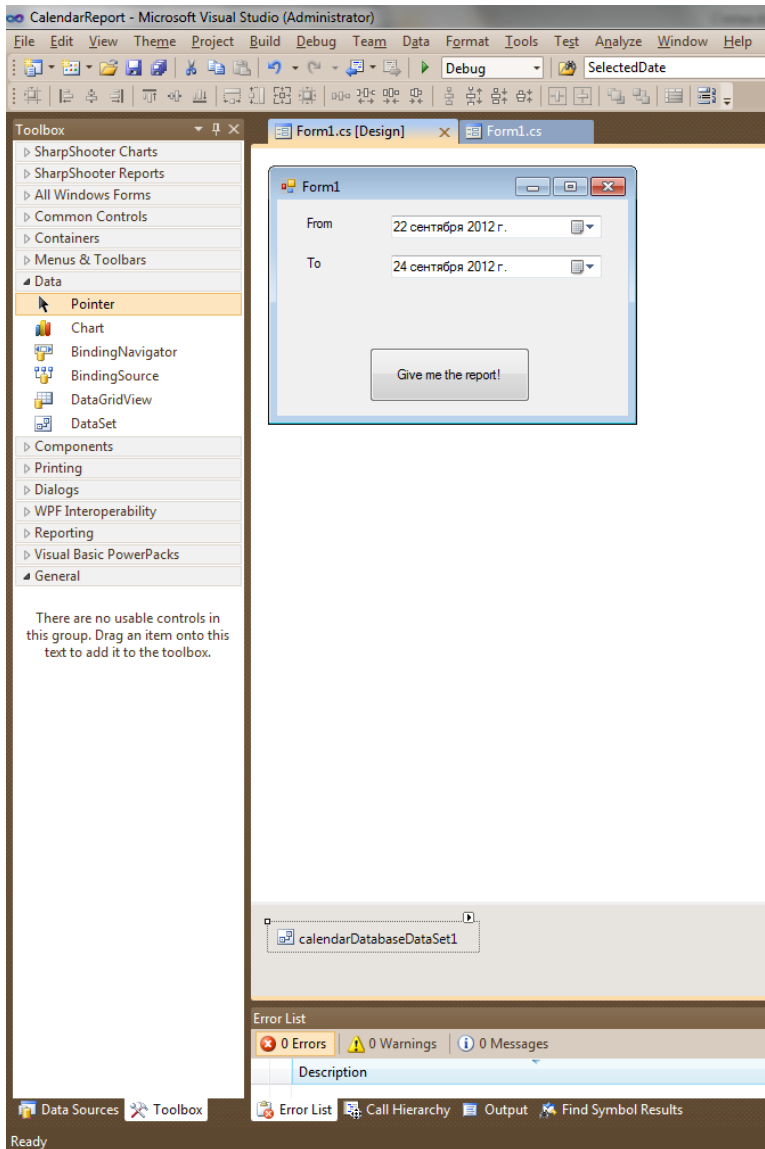
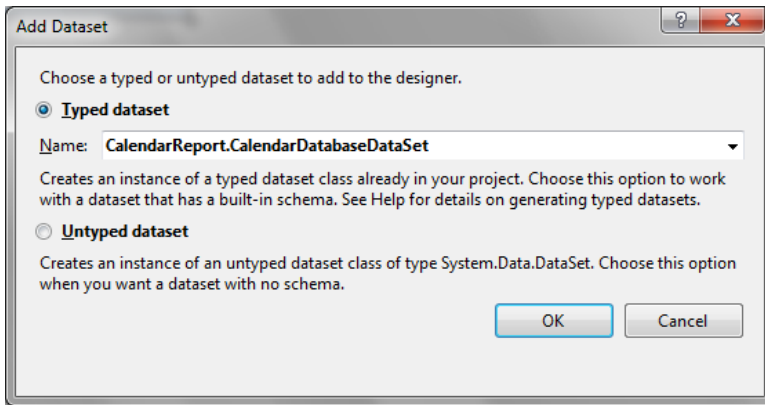
Now, let's add a database as a data source:





Now we need to pass the database through DataSet

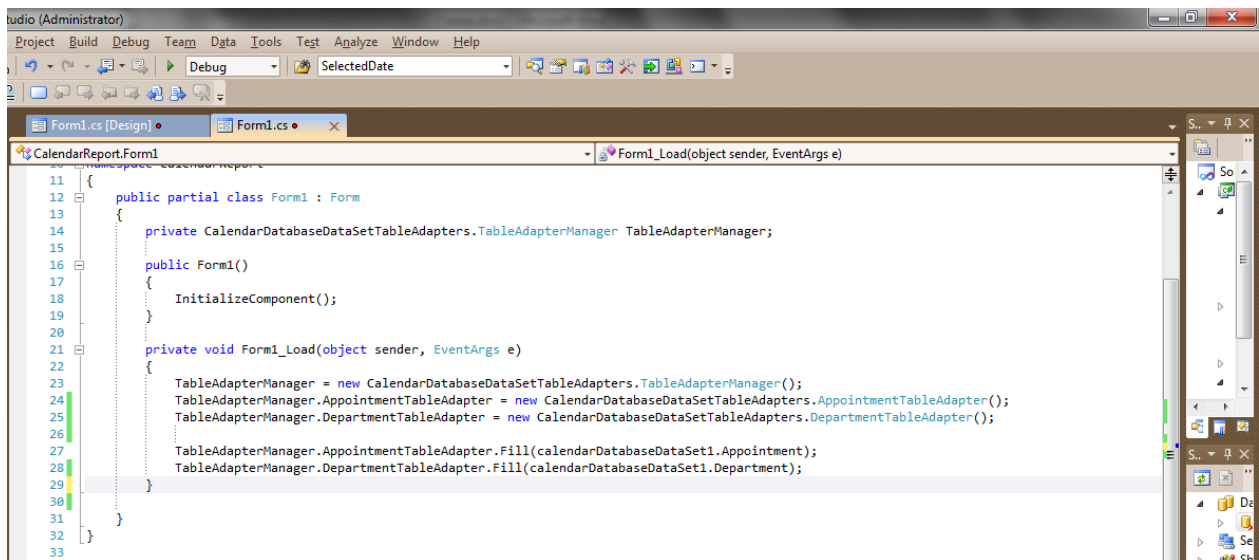




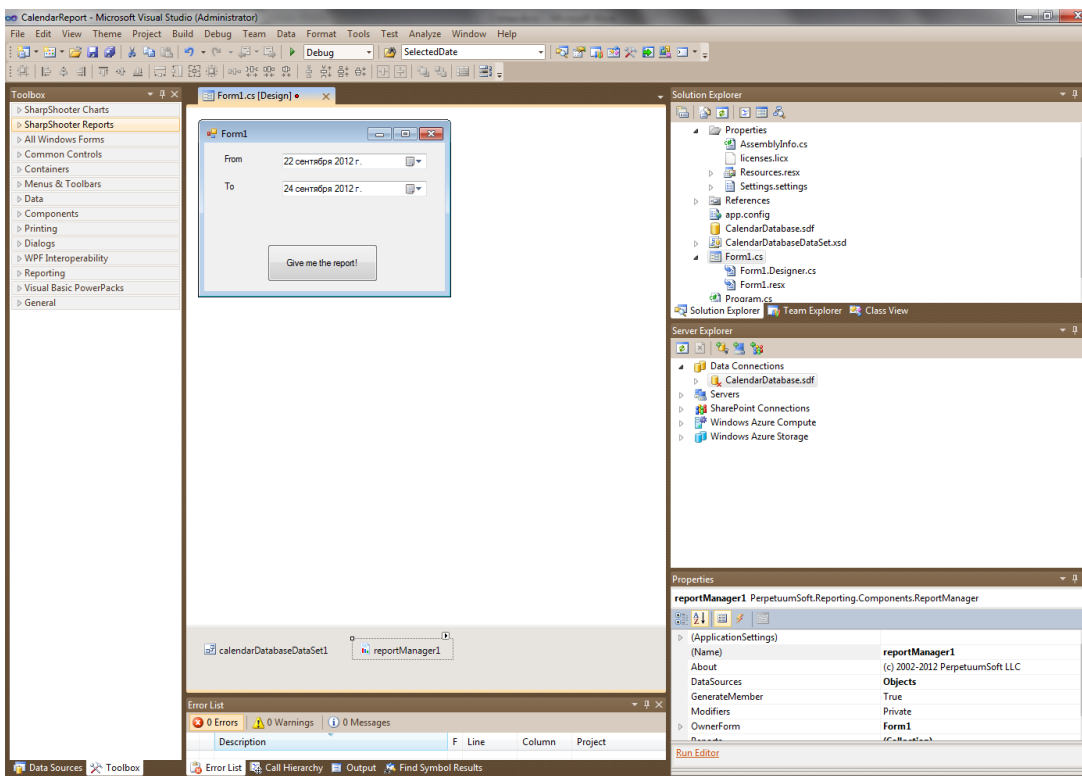
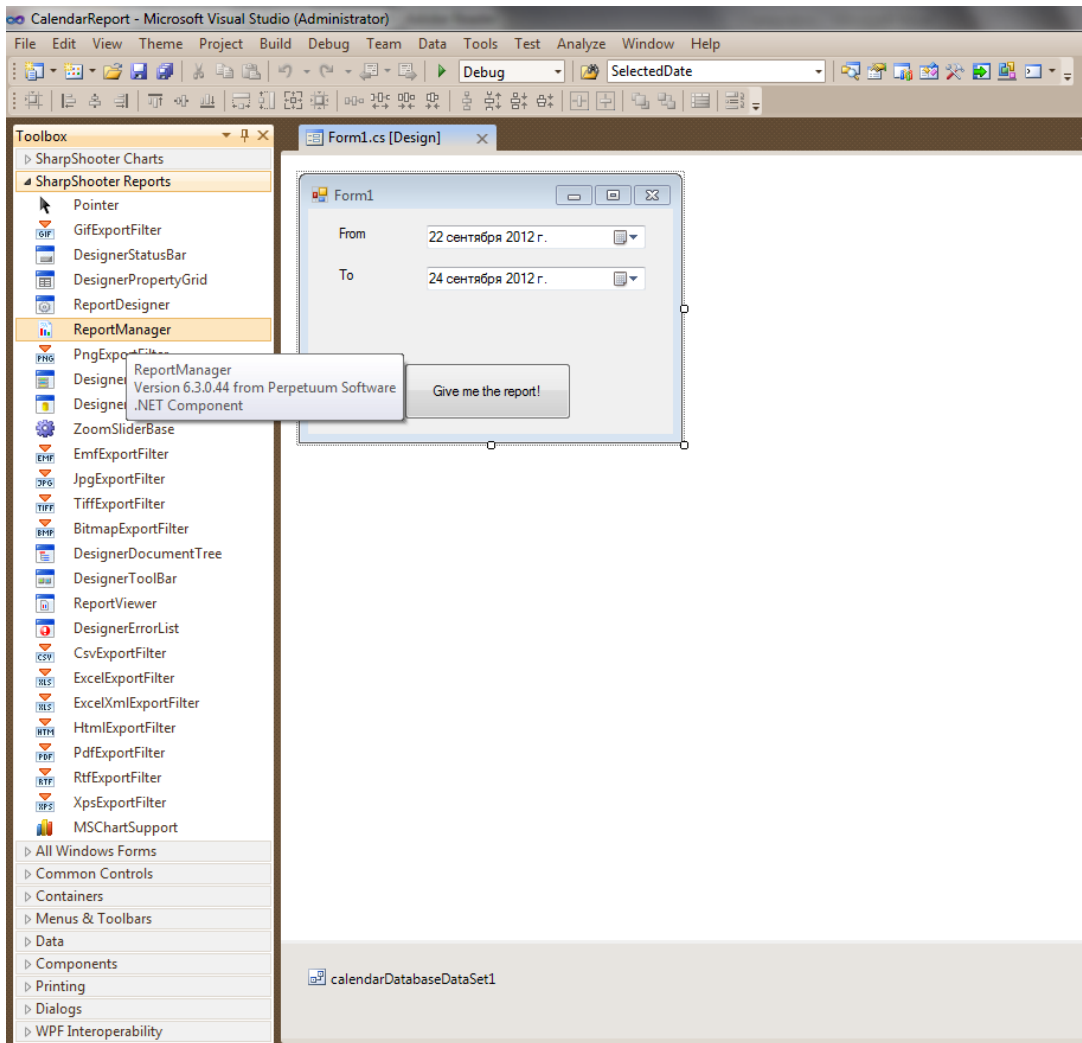
Let's create TableAdapterManager and table adapters.



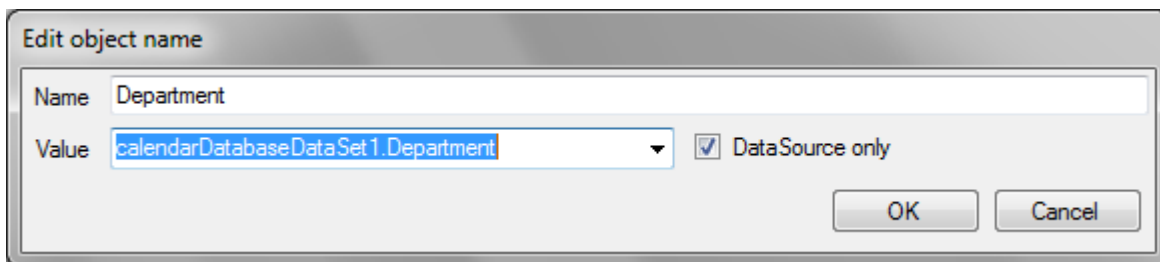
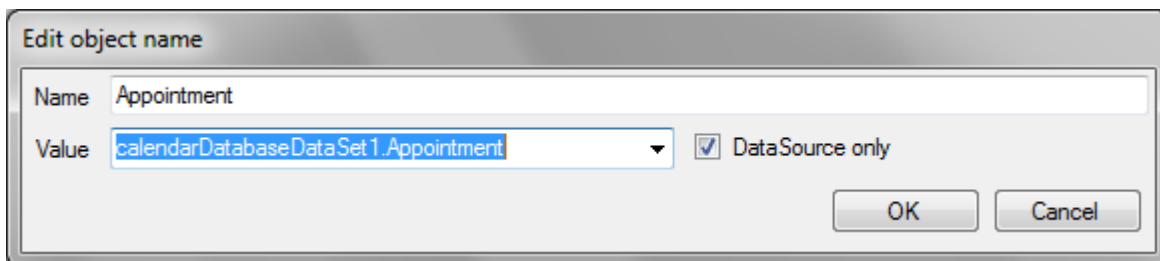
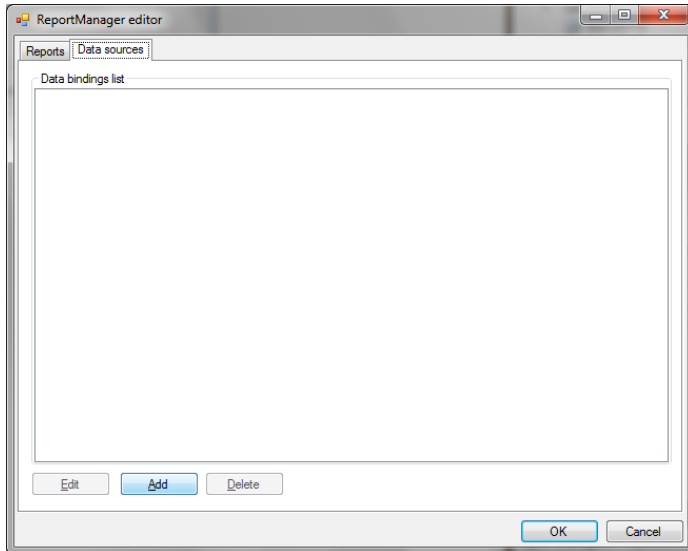
Fill in the DataSet



Now drag and drop the ReportManager component on the form:

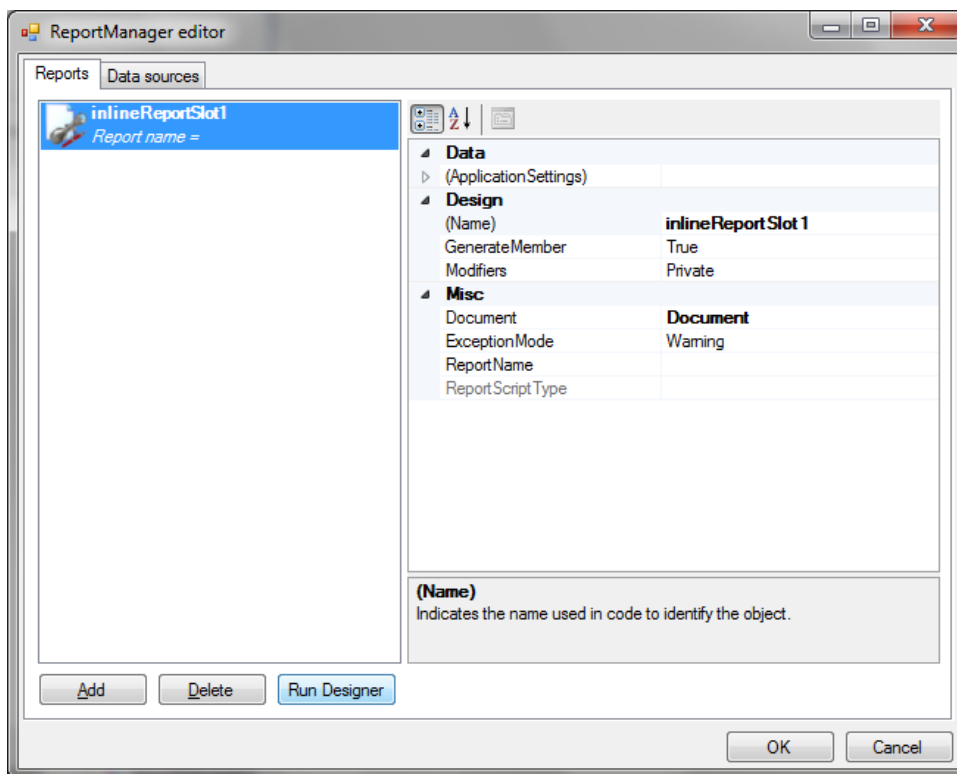
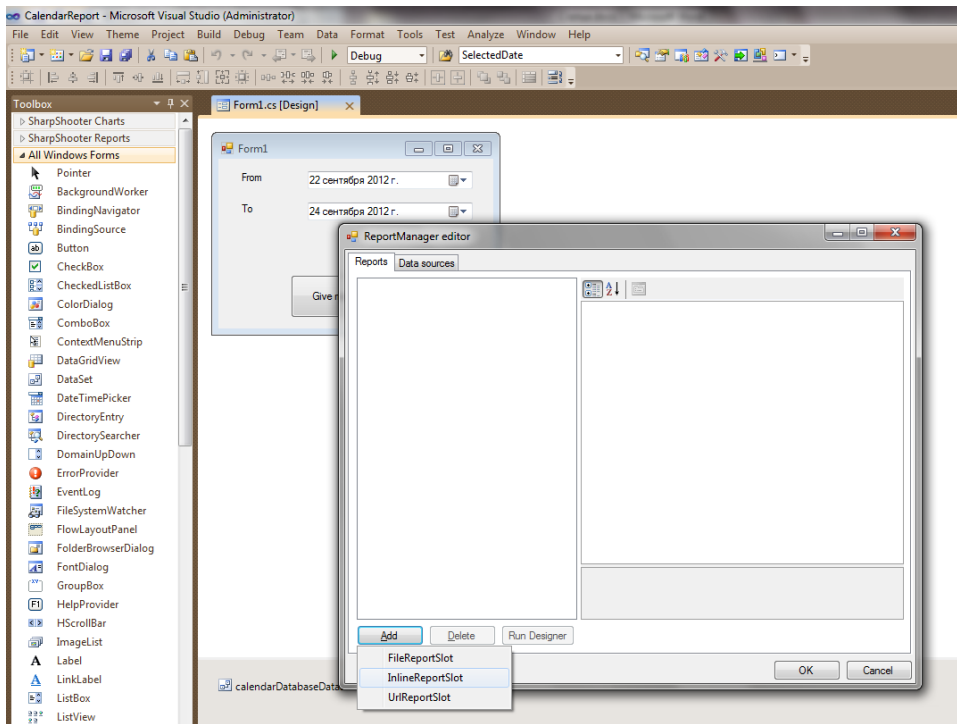


Add data sources for the report:

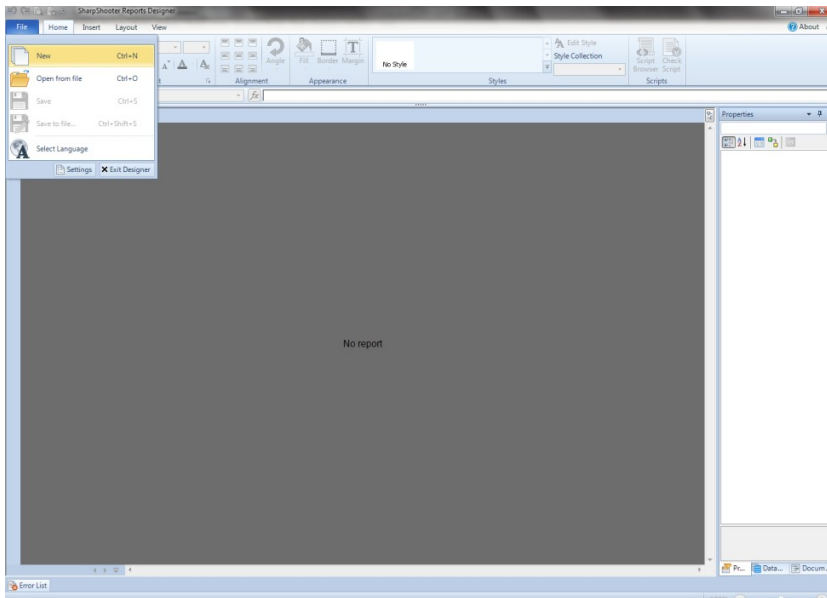


Now, let's add a report template:

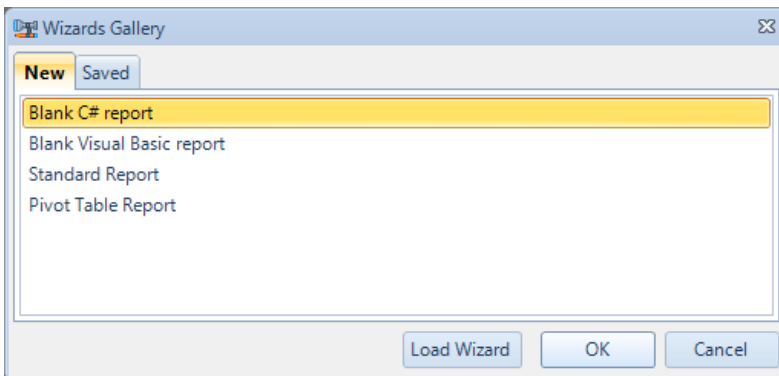




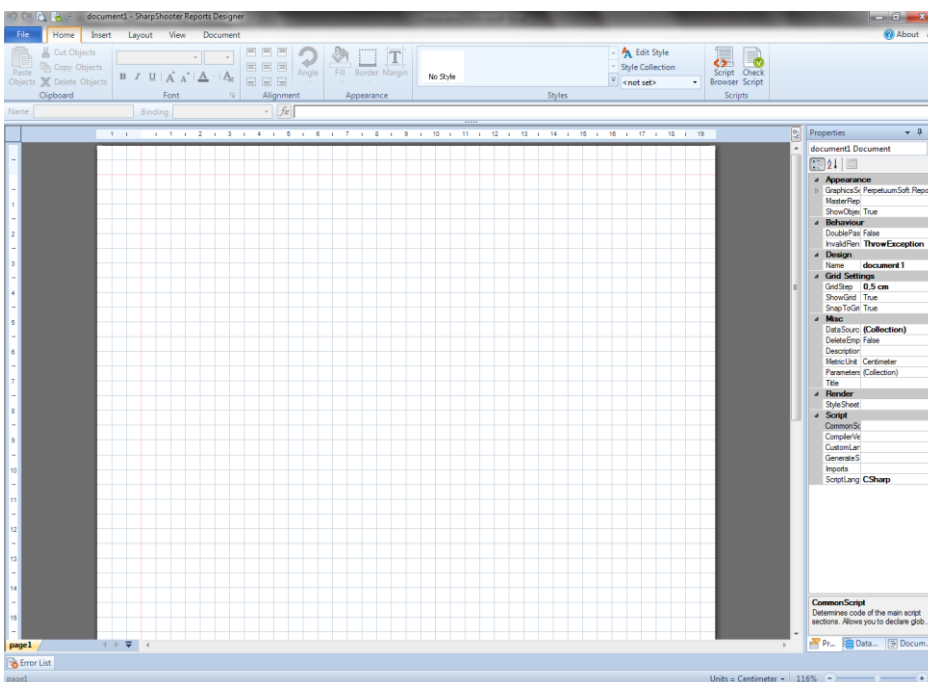
And run the template designer by clicking the Run Designer button. Create a blank template in the designer.



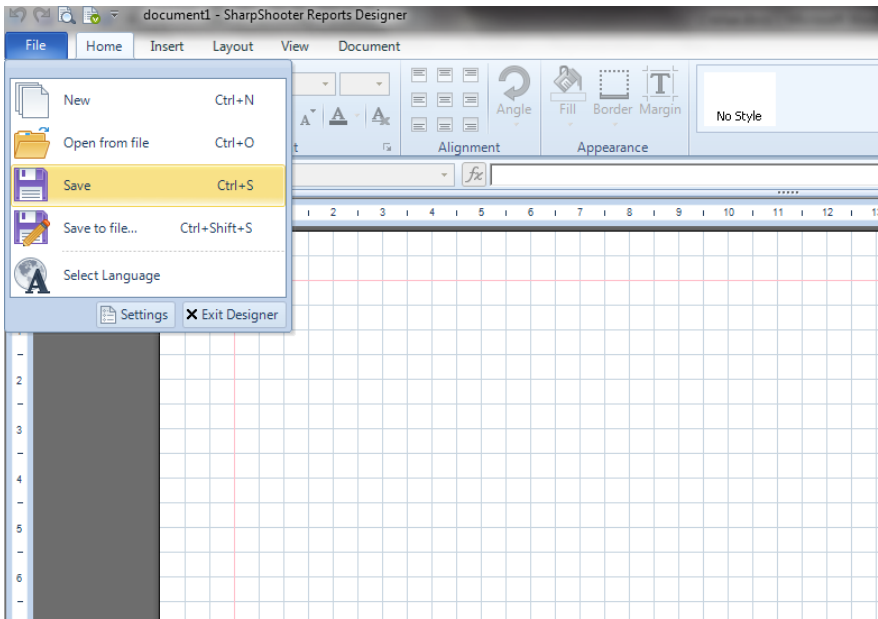
Select C# as script language and press OK.



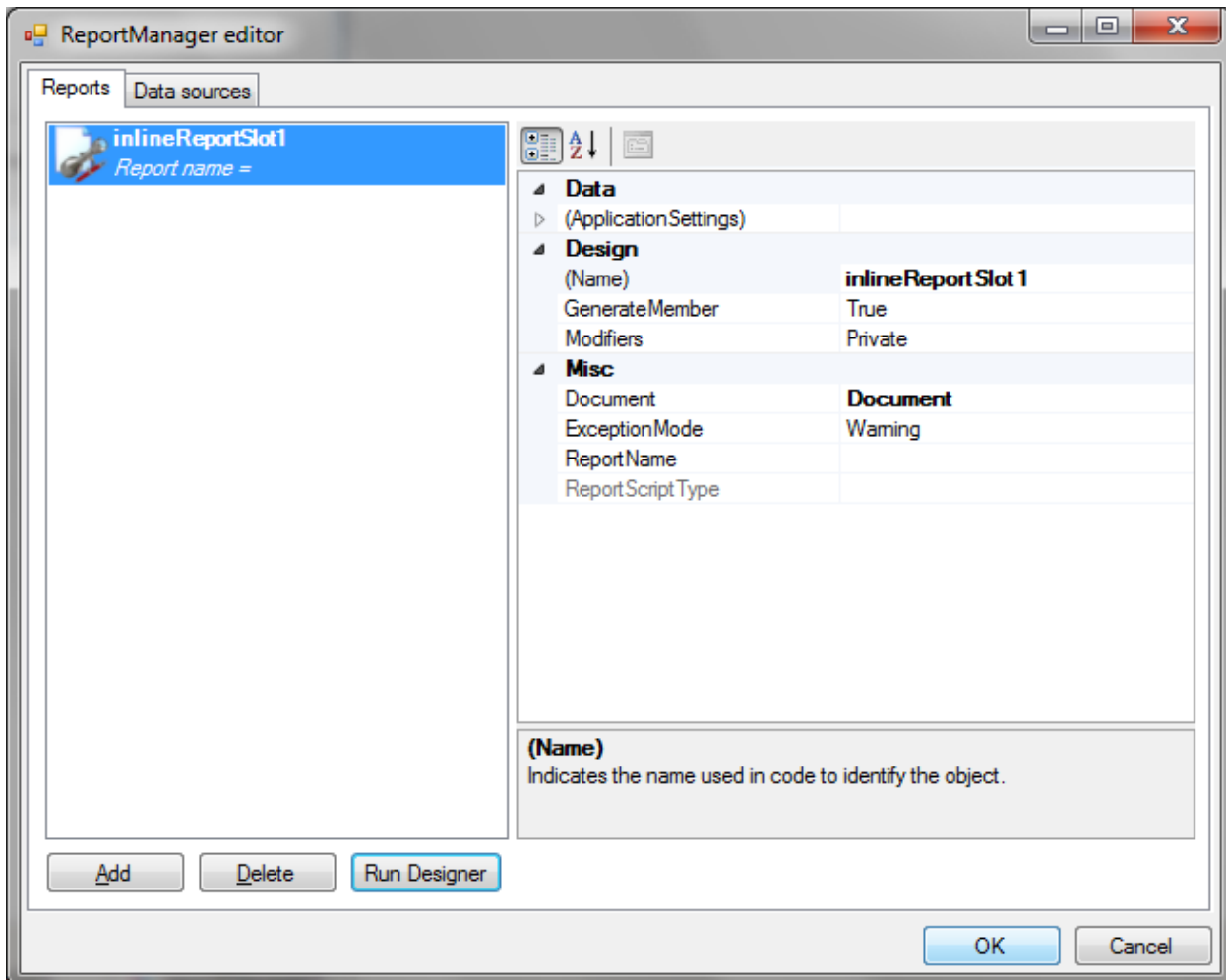
Blank template is ready.



Now save it through the File menu or by pressing Ctrl + S.



Now you can close the Designer.



Close the report designer by pressing the OK button.

Let's add handlers for button pressing and for the finish of report drawing.

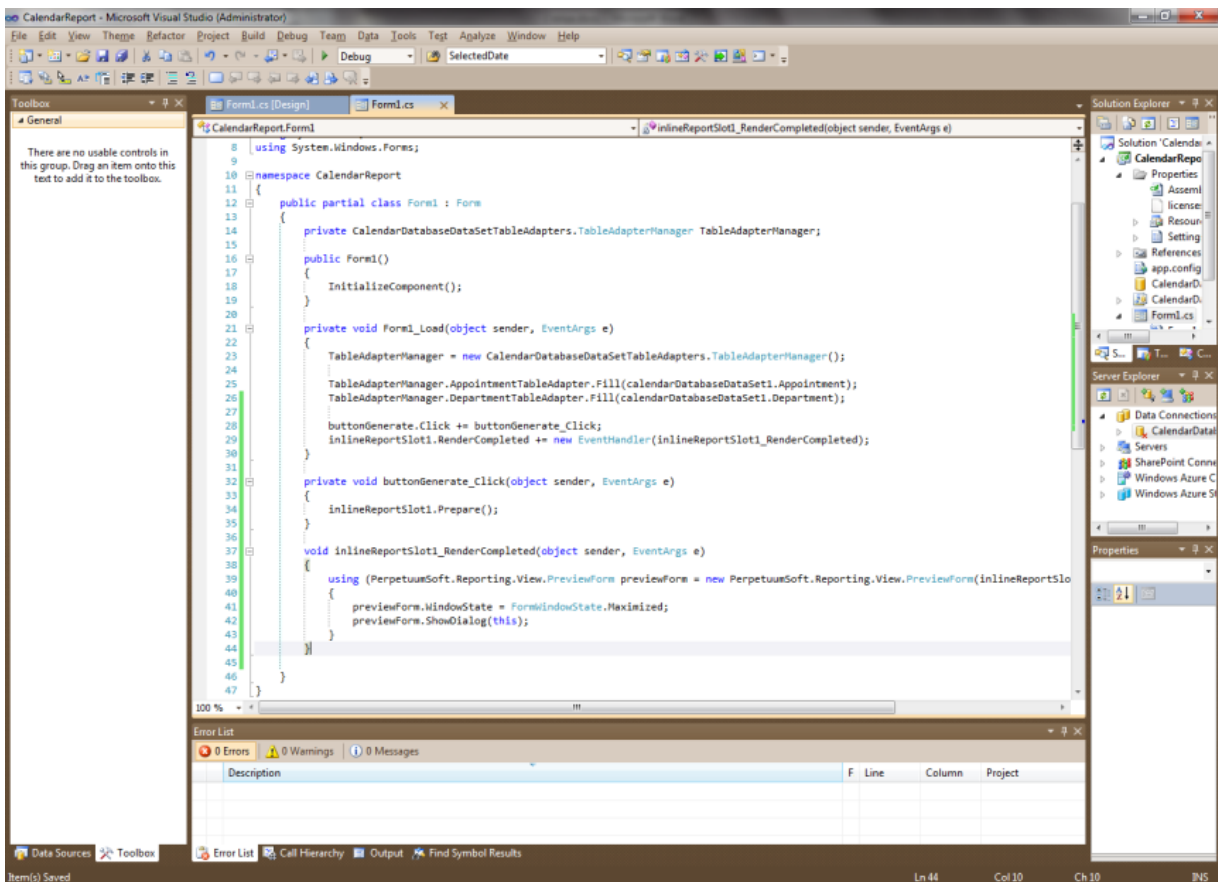
```
private void Form1_Load(object sender, EventArgs e)
{
    TableAdapterManager = new CalendarDatabaseDataSetTableAdapters.TableAdapterManager();
    TableAdapterManager.AppointmentTableAdapter
        = new CalendarDatabaseDataSetTableAdapters.AppointmentTableAdapter();
    TableAdapterManager.DepartmentTableAdapter
        = new CalendarDatabaseDataSetTableAdapters.DepartmentTableAdapter();

    TableAdapterManager.AppointmentTableAdapter.Fill(calendarDatabaseDataSet1.Appointment);
    TableAdapterManager.DepartmentTableAdapter.Fill(calendarDatabaseDataSet1.Department);

    buttonGenerate.Click += buttonGenerate_Click;
    inlineReportSlot1.RenderCompleted += new
    EventHandler(inlineReportSlot1_RenderCompleted);
}

private void buttonGenerate_Click(object sender, EventArgs e)
{
    GenerateDays();
    inlineReportSlot1.Prepare();
}

void inlineReportSlot1_RenderCompleted(object sender, EventArgs e)
{
    using (PerpetuumSoft.Reporting.View.PreviewForm previewForm
        = new PerpetuumSoft.Reporting.View.PreviewForm(inlineReportSlot1))
    {
        previewForm.WindowState = FormWindowState.Maximized;
        previewForm.ShowDialog(this);
    }
}
```



Now everything is ready for report building.

## Schedule №1

For the beginning we can do a simple report, for example as follows:

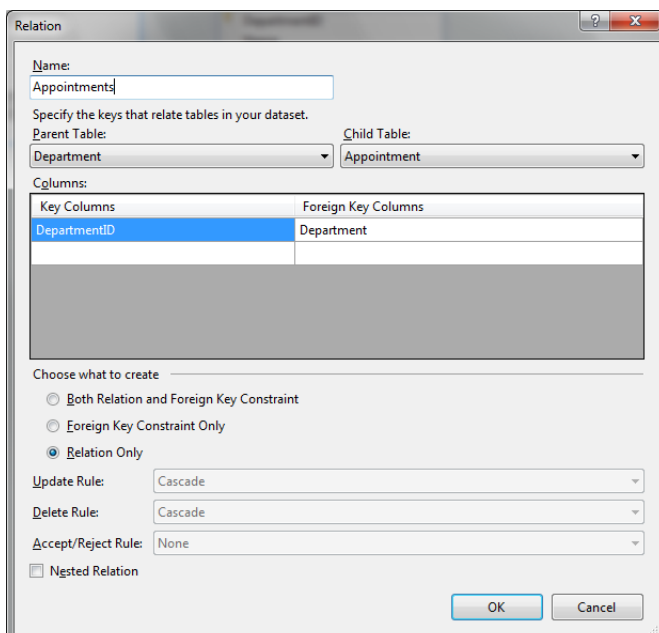
18.09.2012	
Department A	
Time	Appointment
9:00	feed cat
10:00	feed dog
12:00	feed hamster
Department B	
Time	Appointment
9:00	feed other cat
10:00	feed other dog
12:00	feed other hamster
Department C	
Time	Appointment
9:00	sleep

There is a schedule for every date for every department. The tasks are one by one sorted by time.

Let's formalize the task:

1. We need to display all days within the set period.
2. Display date for every day and headers for all departments.
3. Display appointments assigned for a particular day for every department.

For the beginning, let's create the Appointments relation for DataSet from the data base. We will use it to display all appointments assigned to a particular department.



Relation

Name: Appointments

Specify the keys that relate tables in your dataset.

Parent Table: Department Child Table: Appointment

Columns:

Key Columns	Foreign Key Columns
DepartmentID	Department

Choose what to create

☐ Both Relation and Foreign Key Constraint  
☐ Foreign Key Constraint Only  
☒ Relation Only

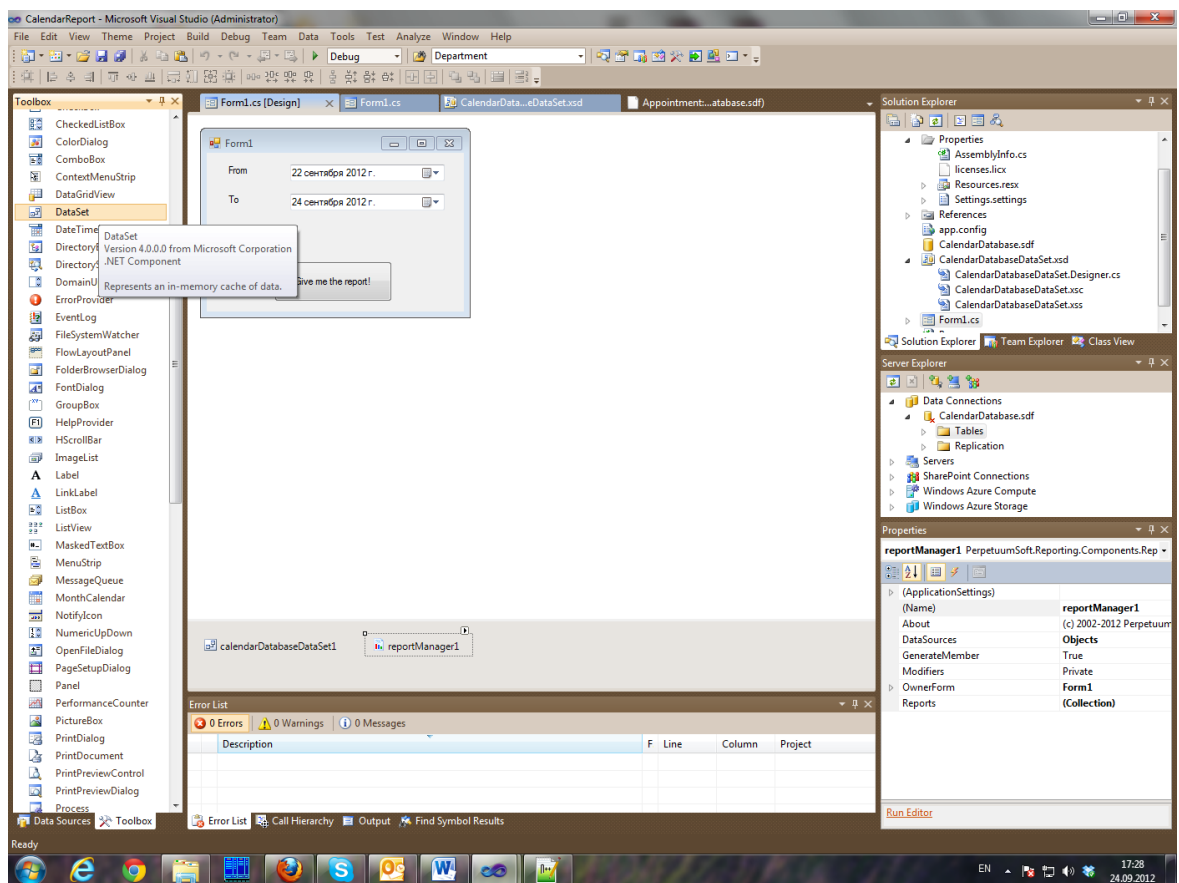
Update Rule: Cascade

Delete Rule: Cascade

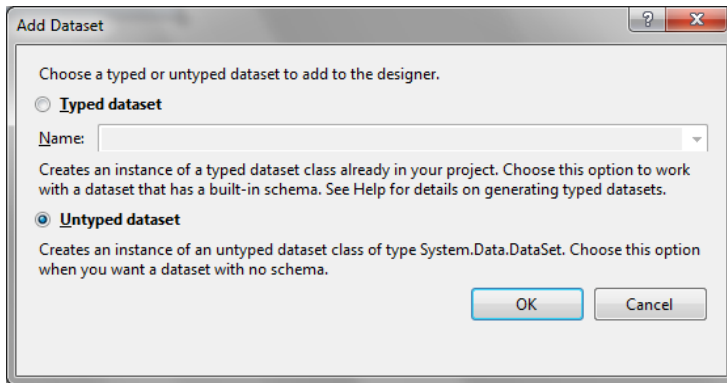
Accept/Reject Rule: None

☐ Nested Relation

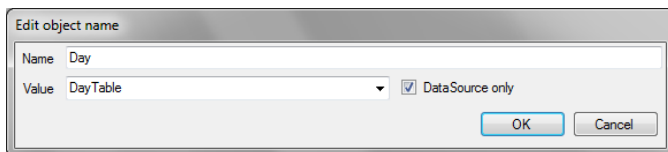
OK Cancel



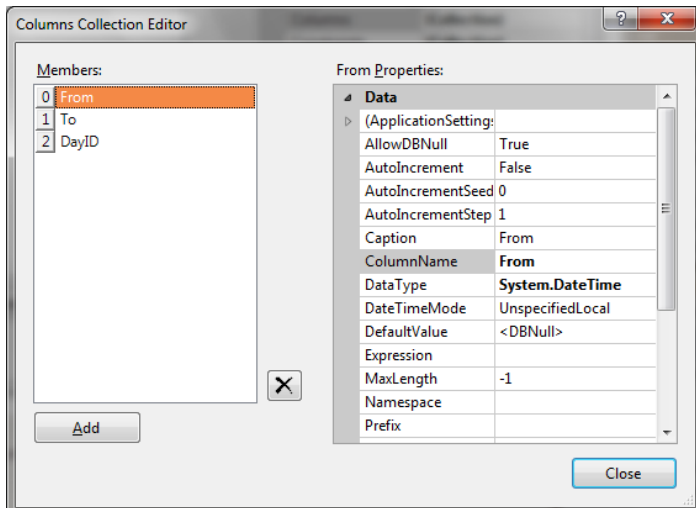
14



Now we will use a table with days for which we create a schedule.



Every record of a table will contain start and end of a day.

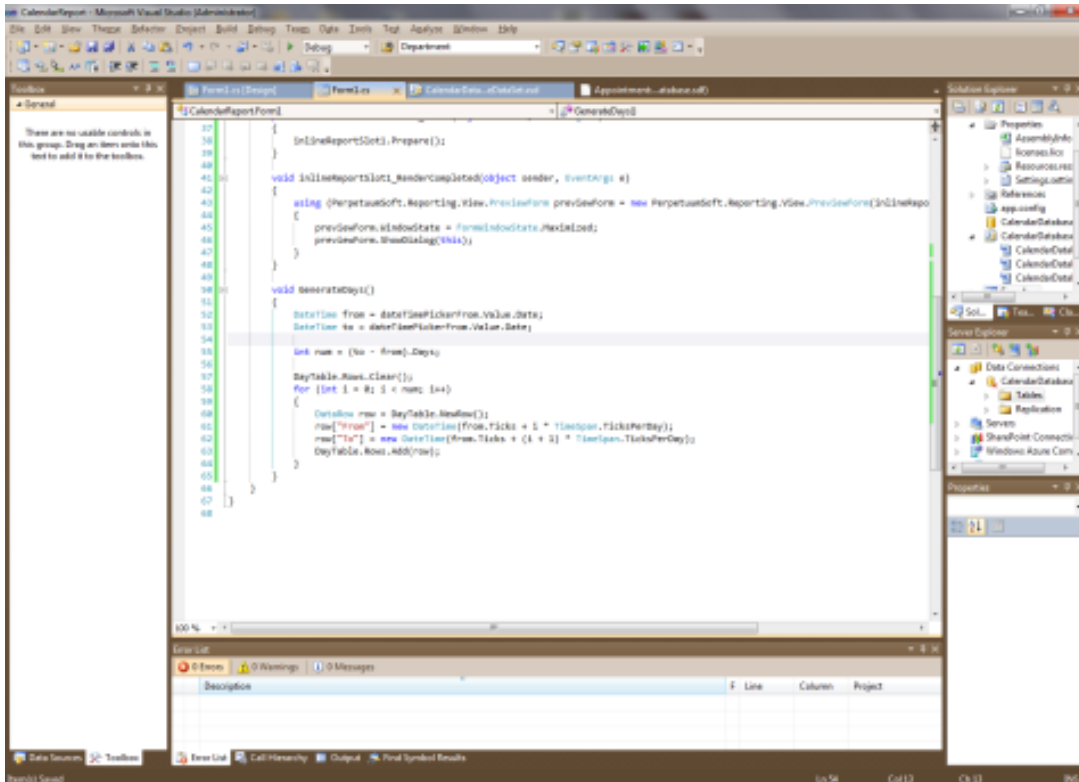


Now, the days table is needed to be filled with time periods which set days. To do that I wrote the GenerateDays function.

```
void GenerateDays()
{
    DateTime from = dateTimePickerFrom.Value.Date;
    DateTime to = dateTimePickerTo.Value.Date;

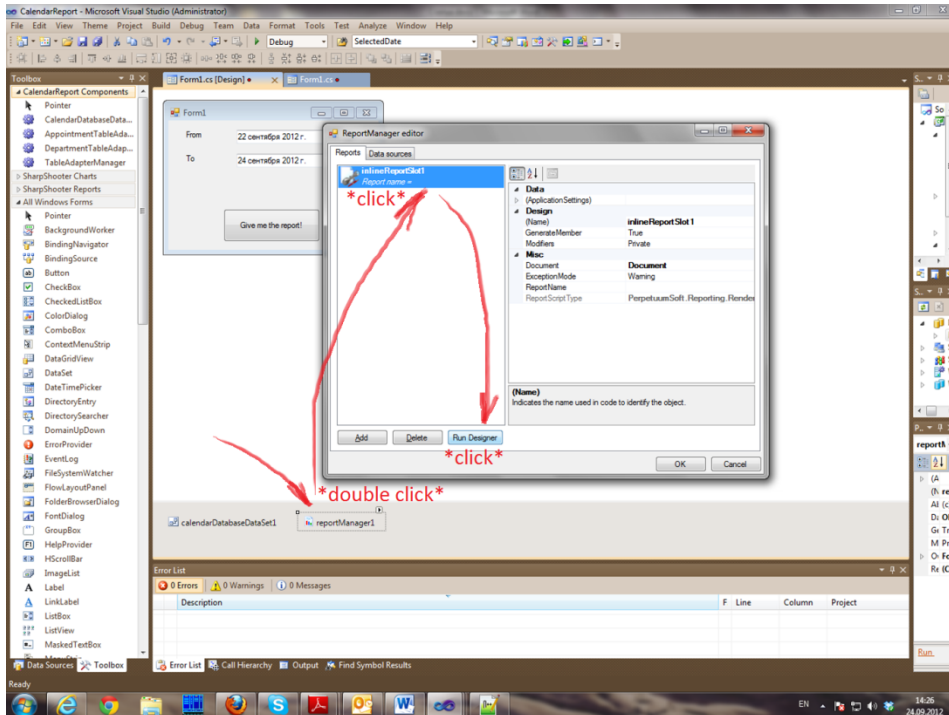
    int num = (to - from).Days;

    DayTable.Rows.Clear();
    for (int i = 0; i < num; i++)
    {
        DataRow row = DayTable.NewRow();
        row["From"] = new DateTime(from.Ticks + i * TimeSpan.TicksPerDay);
        row["To"] = new DateTime(from.Ticks + (i + 1) * TimeSpan.TicksPerDay);
        DayTable.Rows.Add(row);
    }
}
```

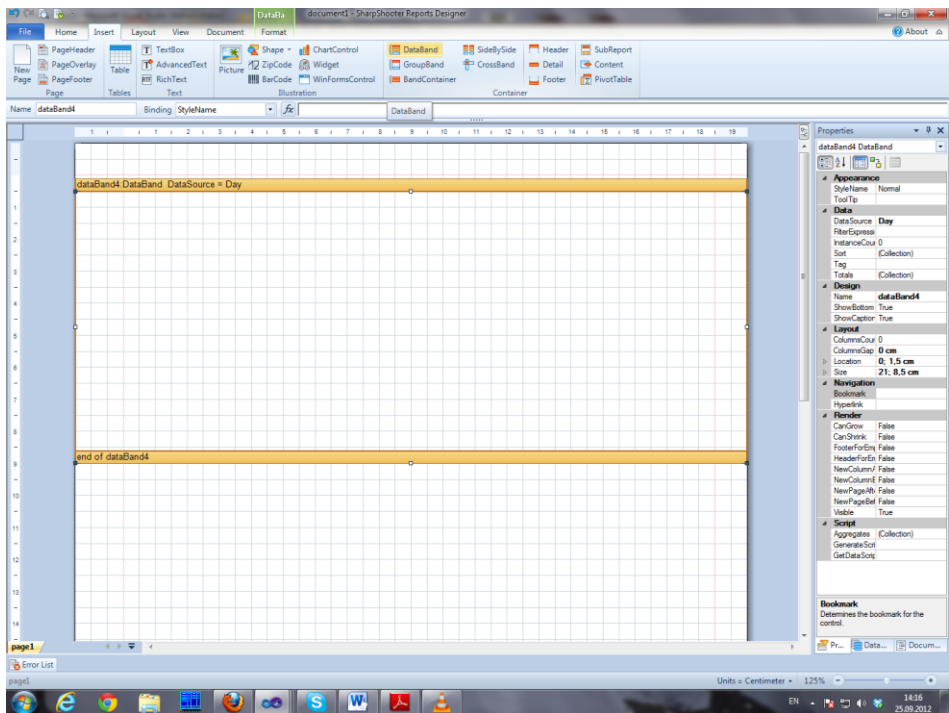




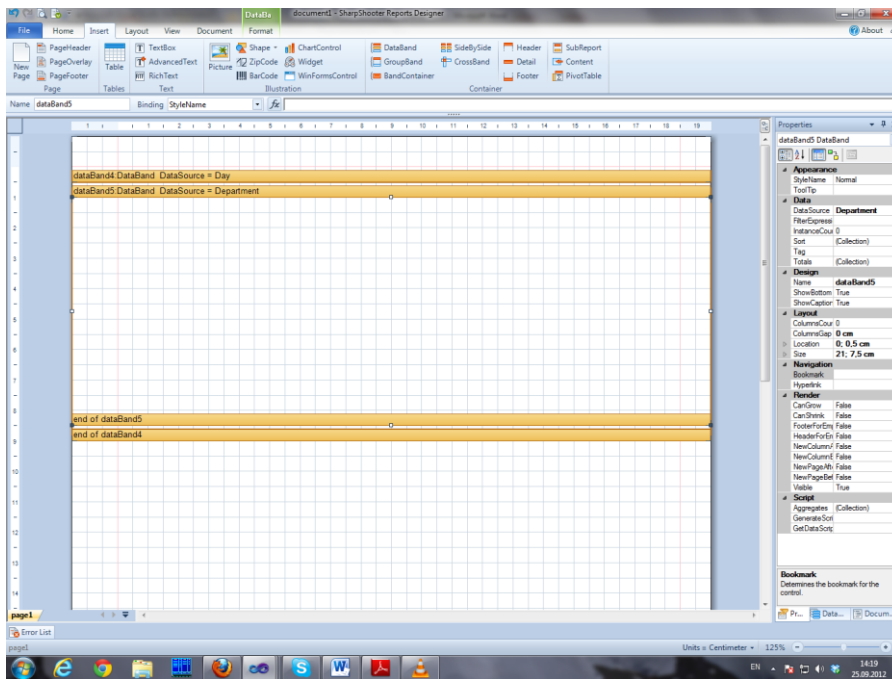
Now we are ready to move to a template.



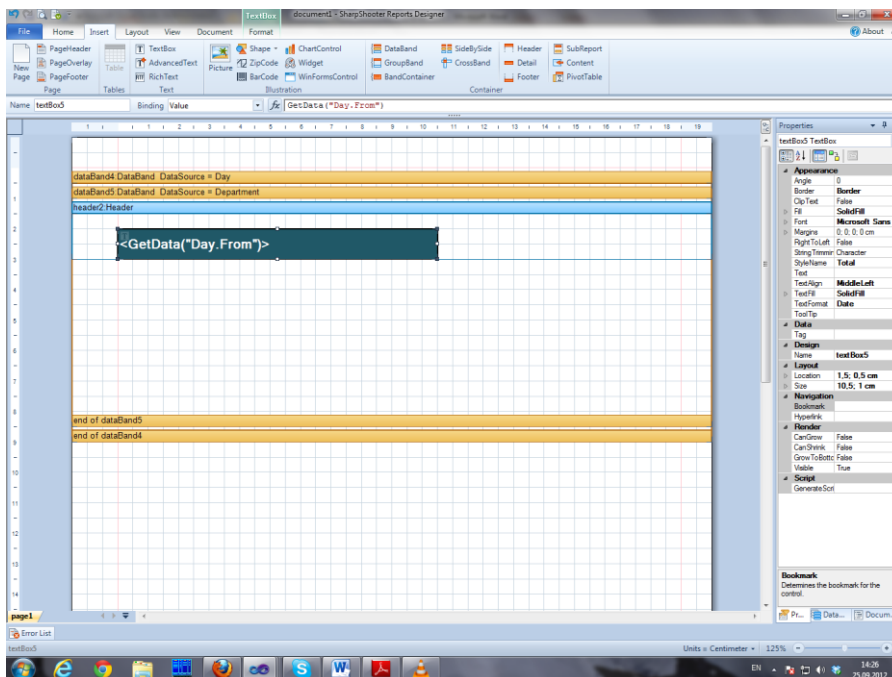
The first DataBand (dataBand4) will go through days. We add it to a template and set the DataSource property to Days.



The second DataBand (dataBand5) will go through all departments. Its DataSource – Department.

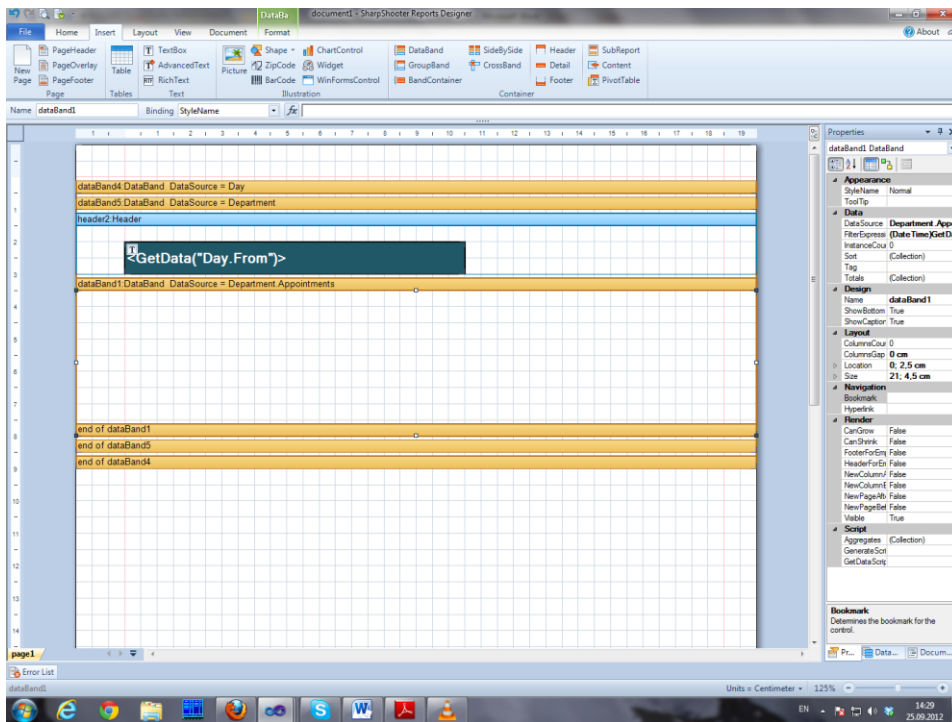


But before a list of departments the dataBand5 should display a date of the current day from the dataBand4. We will use the Header element for this purpose.



Put the TextBox inside and bind it to the GetData("Day.From") value.

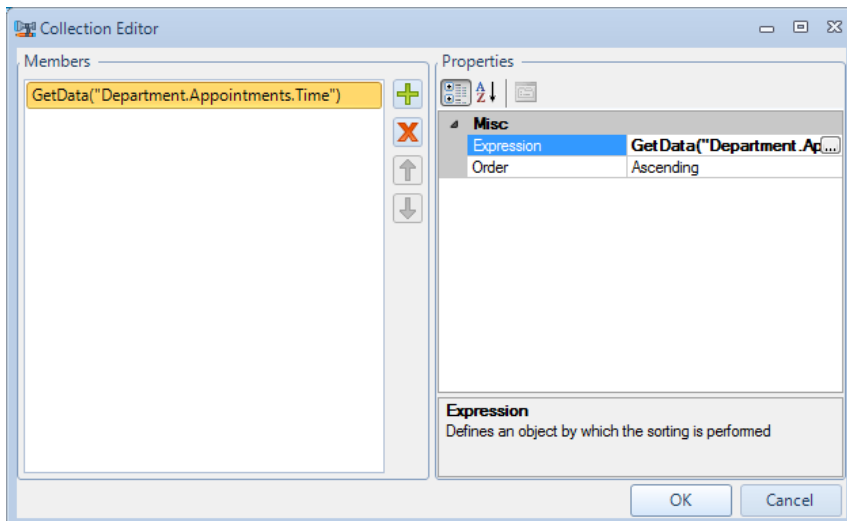
Now, we go through all assigned appointments for every department. To do this we add one more DataBand (dataBand1) to dataBand5. The DataSource property we set the Department.Appointments relation.



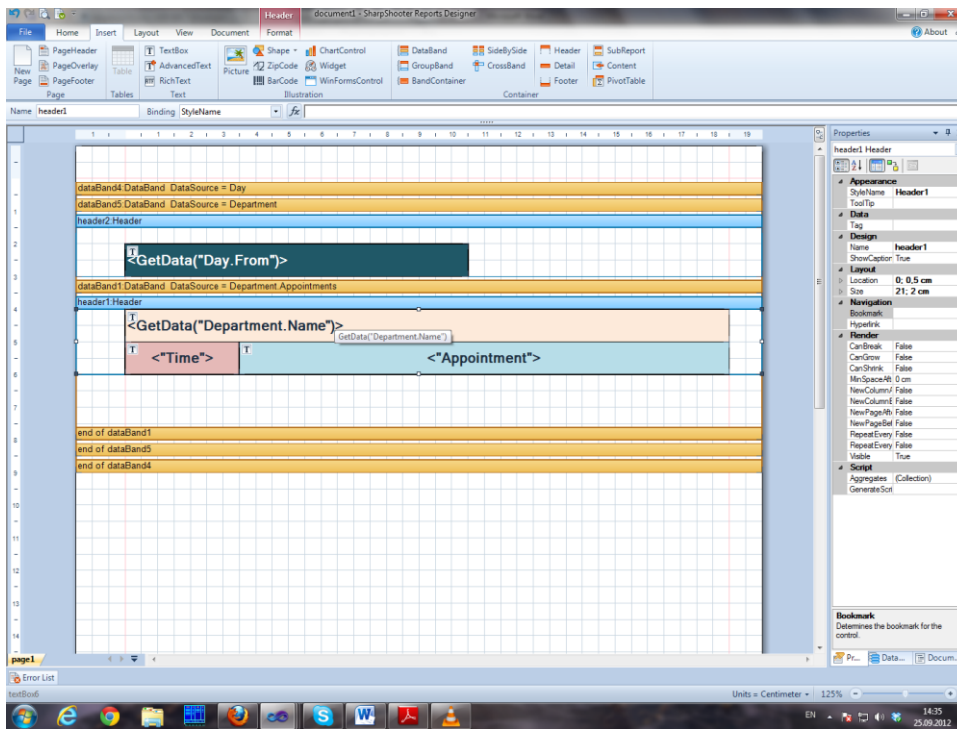
But we need only appointments for the current day. To filter by this appointment parameter in dataBand1, we need to set the following script in the FilterExpression property:

```
(DateTime) GetData ("Department.Appointments.Time") >= (DateTime) dataBand4 ["From"]
&&
(DateTime) GetData ("Department.Appointments.Time") < (DateTime) dataBand4 ["To"]
```

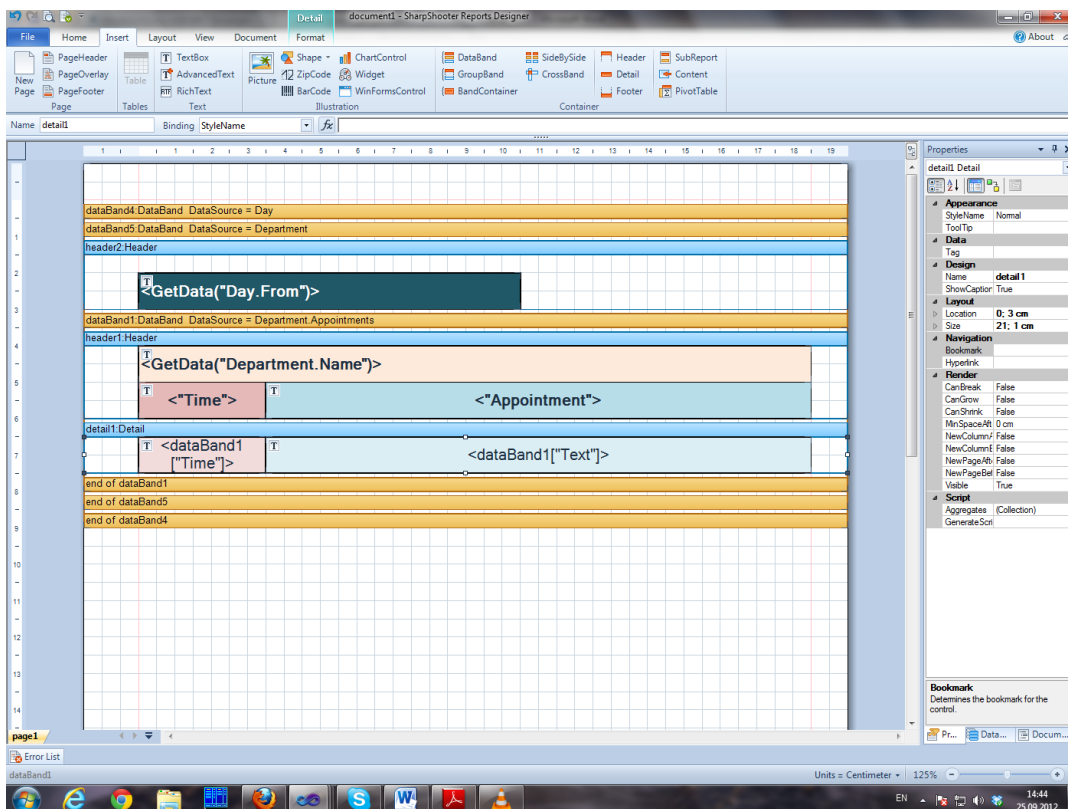
The records must be also sorted by time. To do this we need to add the  
`GetData ("Department.Appointments.Time")` criterion to the Sort property.



Now we need to add the Header element to dataBand1 to display the name of the current department and headers of time and description of the current appointment.



Now we need just to display the record of the appointment itself. We will use the detail element for this purpose.

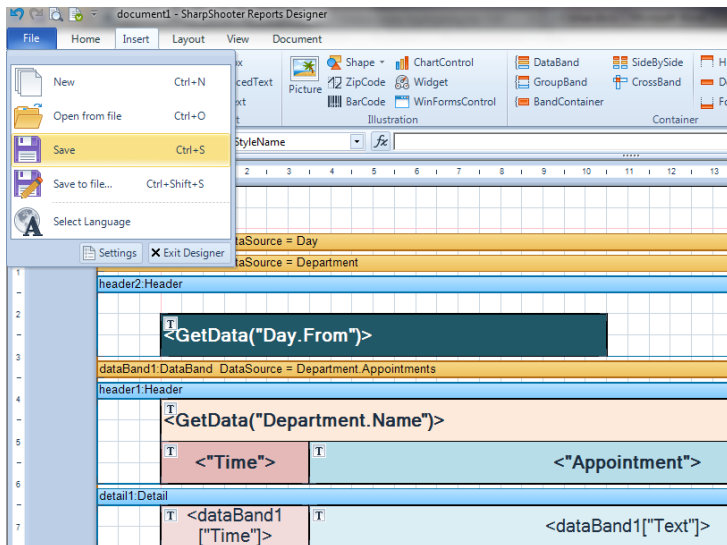


Lets add 2 TextBox elements to detail1. These TextBoxes are bound to the time fields of time and description of the current record.

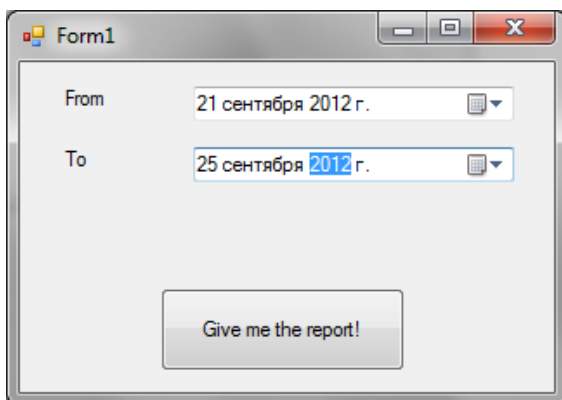
The script for the first case is dataBand1["Time"];

The script for the second case is dataBand1["Text"];

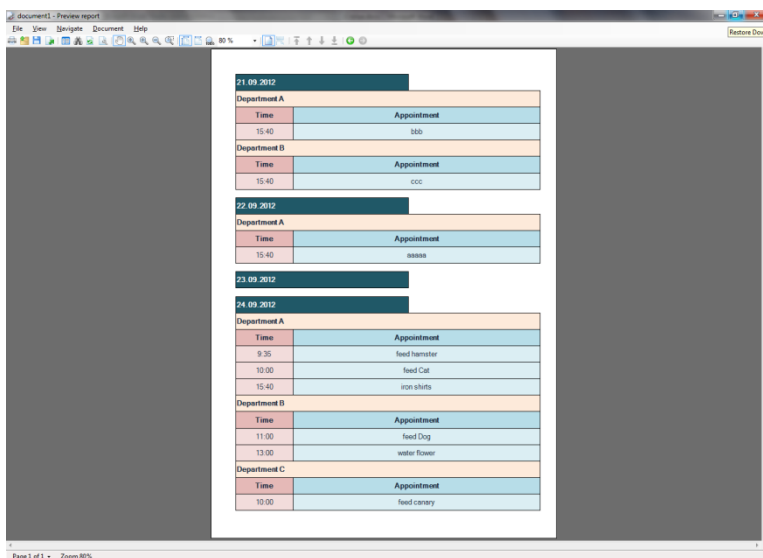
## Save the project



## Run it



## Press the big button



18.09.2012	
Department A	
Time	Appointment
9:00	feed cat
10:00	feed dog
12:00	feed hamster
Department B	
Time	Appointment
9:00	feed other cat
10:00	feed other dog
12:00	feed other hamster
Department C	
Time	Appointment
9:00	sleep

21.09.2012	
Department A	
Time	Appointment
15:40	bbb
Department B	
Time	Appointment
15:40	ccc
22.09.2012	
Department A	
Time	Appointment
15:40	aaaaa
23.09.2012	
24.09.2012	
Department A	
Time	Appointment
9:35	feed hamster
10:00	feed Cat
15:40	iron shirts
Department B	
Time	Appointment
11:00	feed Dog
13:00	water flower
Department C	
Time	Appointment
10:00	feed canary

Congratulations! The resulting report is exactly what we wanted to do!

*To be continued...*