Creating Daily Schedule Reports using SharpShooter Reports (Part II)

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 started from building a simple Windows Forms application that displays the reports we create. We also created our first report - a schedule of appointments for every department. The appointments will be displayed one by one and sorted by time.

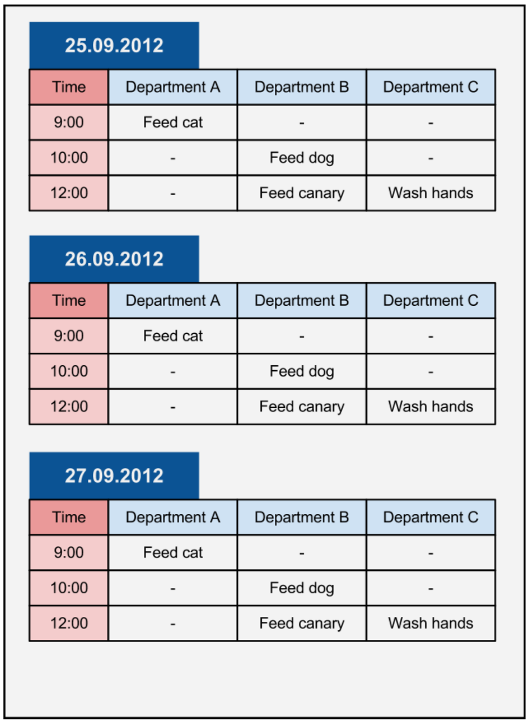
In this tutorial we will create more advanced schedule: the appointments will be displayed by time and in parallel for every department. This way it will look more like a real schedule, more visual and handy.

Prerequisites:

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

Schedule №2

Let’s now display the appointments by time and in parallel for every department:



25.09.2012

Time	Department A	Department B	Department C
9:00	Feed cat	-	-
10:00	-	Feed dog	-
12:00	-	Feed canary	Wash hands

26.09.2012

Time	Department A	Department B	Department C
9:00	Feed cat	-	-
10:00	-	Feed dog	-
12:00	-	Feed canary	Wash hands

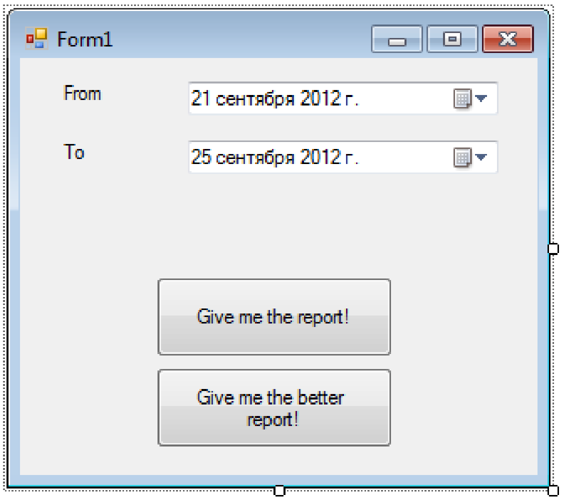
27.09.2012

Time	Department A	Department B	Department C
9:00	Feed cat	-	-
10:00	-	Feed dog	-
12:00	-	Feed canary	Wash hands

Let’s formalize the task:

1. We need to display all days within the indicated period.
2. For every day we need to display the date and also:
   a. The «Time» header and a time column in ascending order.
   b. Headers with names for every department and the appointments assigned to that department for a particular time.

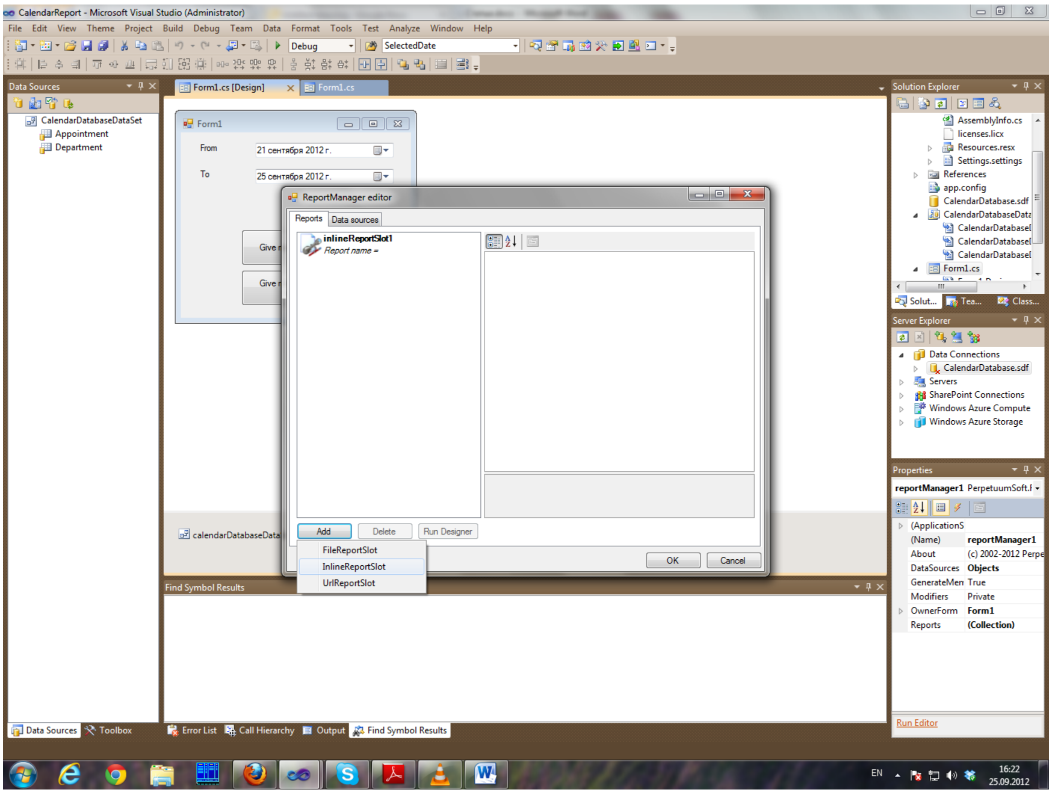
Let’s add one more button to the form:



From 21 сентября 2012 г.

To 25 сентября 2012 г.

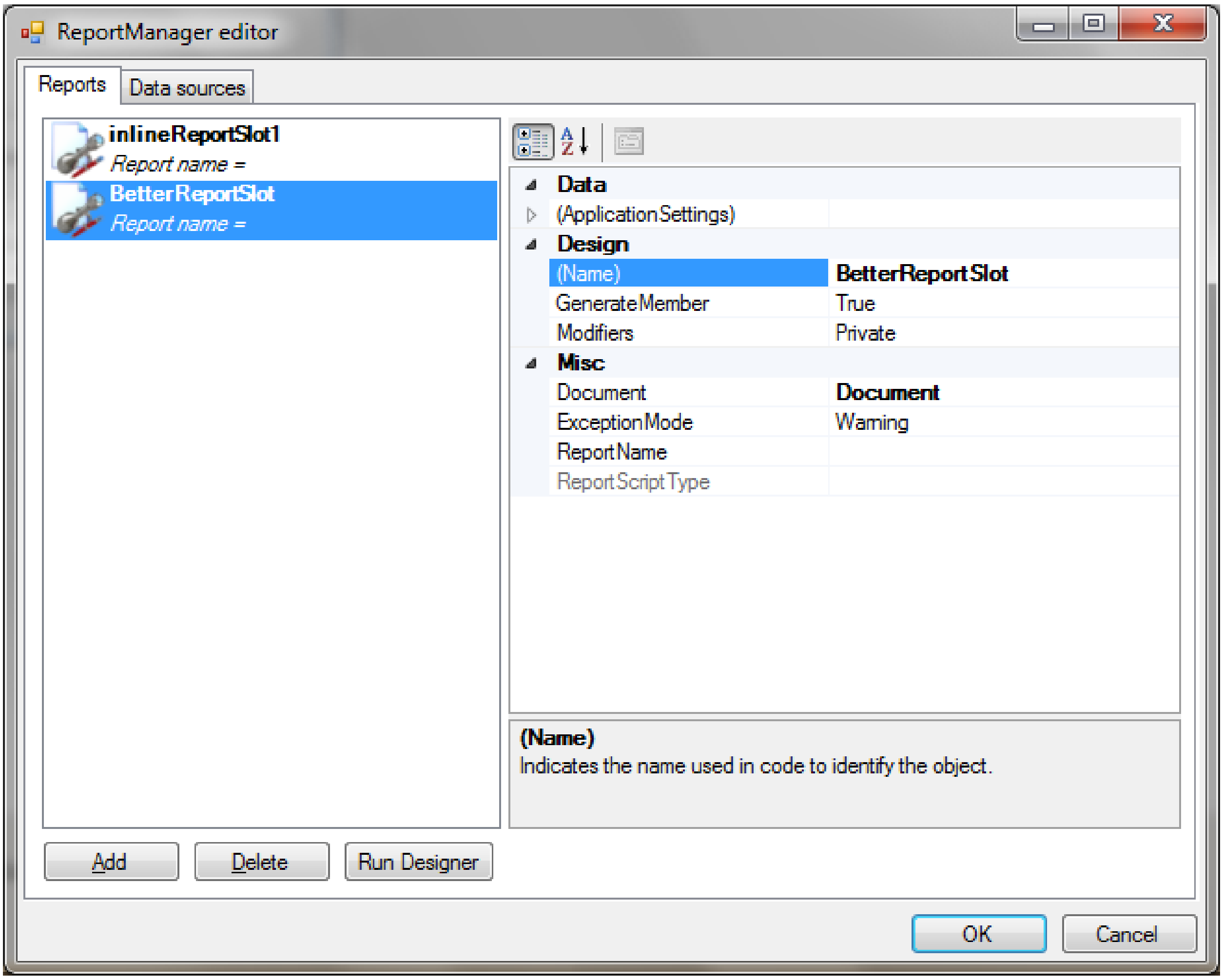
Add one more Inline report.



reportManager1 PerpetuumSoft.I

(ApplicationS	
(Name)	reportManager1
About	(c) 2002-2012 Perpe
DataSources	Objects
GenerateMen	True
Modifiers	Private
OwnerForm	Form1
Reports	(Collection)

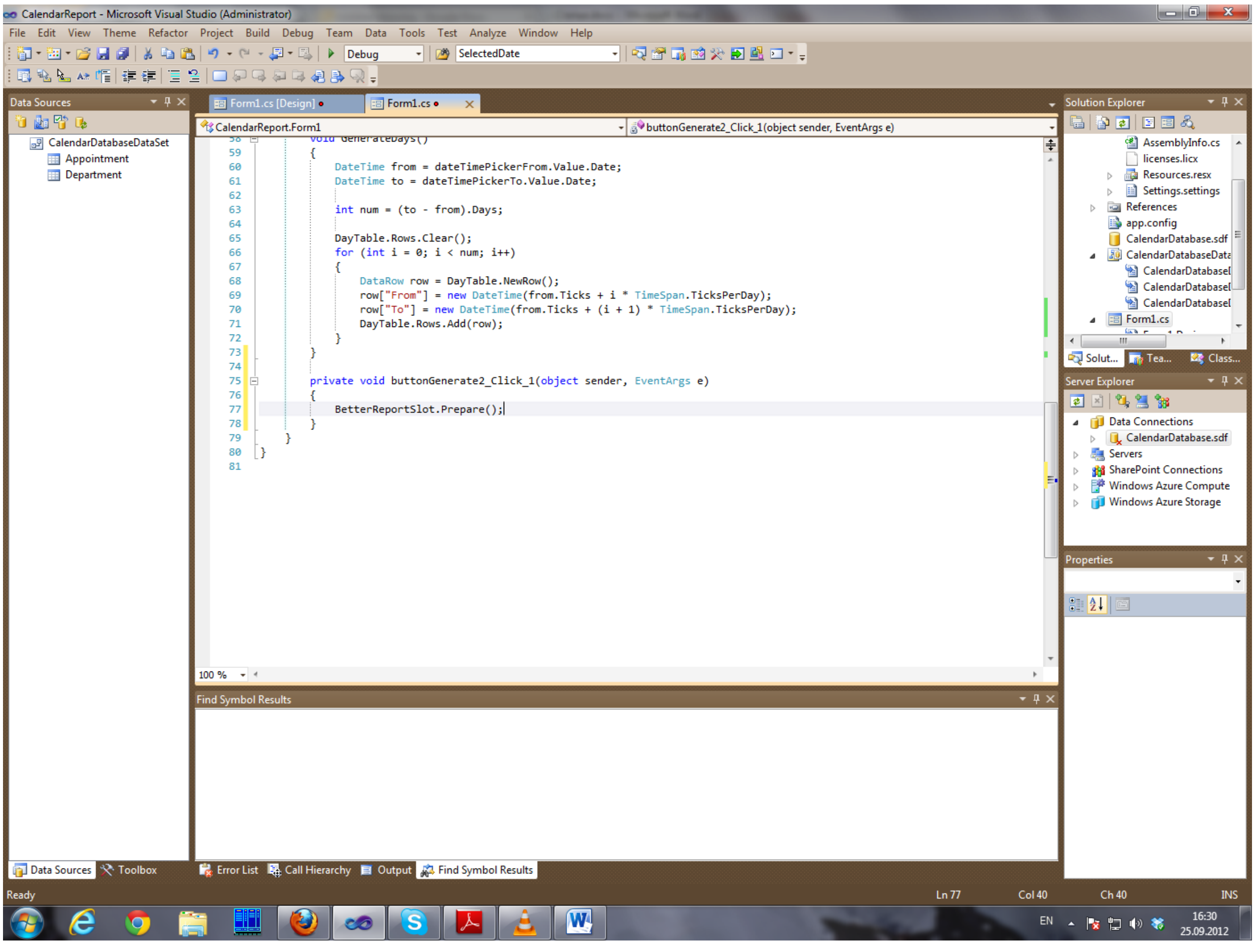
Our second report also needs a name. Let it be BetterReportSlot.



inlineReportSlot1
Report name =
BetterReportSlot
Report name =

Data
(ApplicationSettings)
Design
(Name) BetterReportSlot
GenerateMember True
Modifiers Private
Misc
Document **Document**
ExceptionMode Warning
ReportName
ReportScriptType

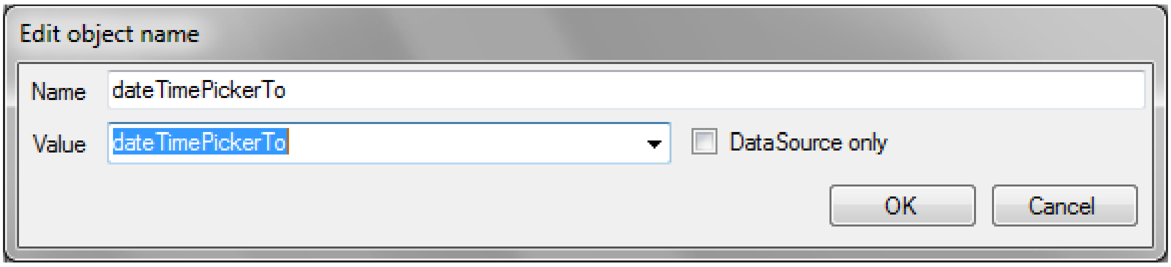
(Name)
Indicates the name used in code to identify the object.



```
        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);
            }
        }

        private void buttonGenerate2_Click_1(object sender, EventArgs e)
        {
            BetterReportSlot.Prepare();
        }
    }
}
```

Subscribe for events and create a new template.

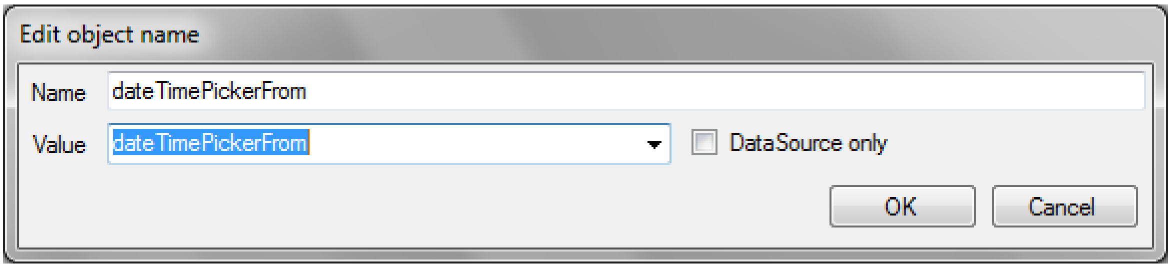
This time we will try not to use the Days table. But anyway we will need some start and end date. We can take them directly from the DateTimePicker controls.



Edit object name

Name dateTimePickerTo

Value dateTimePickerTo ☐ DataSource only



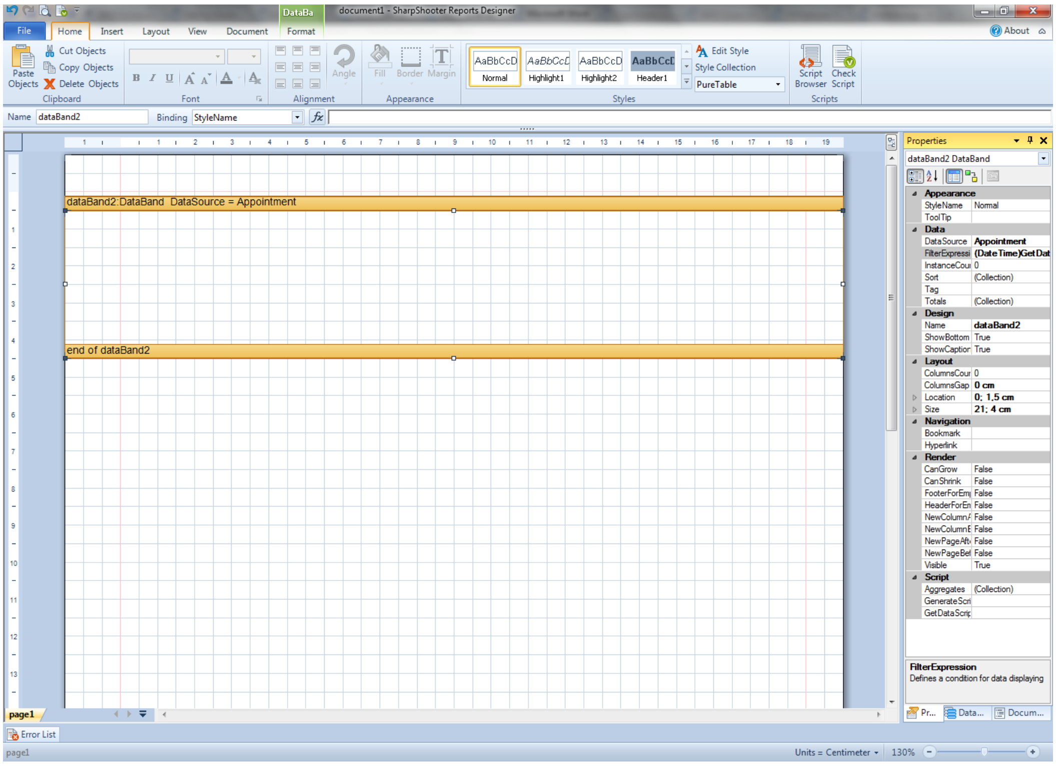
Edit object name

Name dateTimePickerFrom

Value dateTimePickerFrom ☐ DataSource only

Come back to a template.

The first DataBand will go through all Appointment records which are in the specified time period.



Name dataBand2 Binding StyleName

dataBand2:DataBand DataSource = Appointment

end of dataBand2

Properties

dataBand2 DataBand

Appearance	
StyleName	Normal
ToolTip	
Data	
DataSource	**Appointment**
FilterExpressi	**(DateTime)GetDat**
InstanceCour	0
Sort	(Collection)
Tag	
Totals	(Collection)
Design	
Name	**dataBand2**
ShowBottom	True
ShowCaption	True
Layout	
ColumnsCour	0
ColumnsGap	**0 cm**
Location	**0; 1,5 cm**
Size	**21; 4 cm**
Navigation	
Bookmark	
Hyperlink	
Render	
CanGrow	False
CanShrink	False
FooterForEm	False
HeaderForEn	False
NewColumnA	False
NewColumnE	False
NewPageAft	False
NewPageBef	False
Visible	True
Script	
Aggregates	(Collection)
GenerateScri	
GetDataScrip	

FilterExpression
Defines a condition for data displaying

DataSource :

Appointment;

FilterExpression :

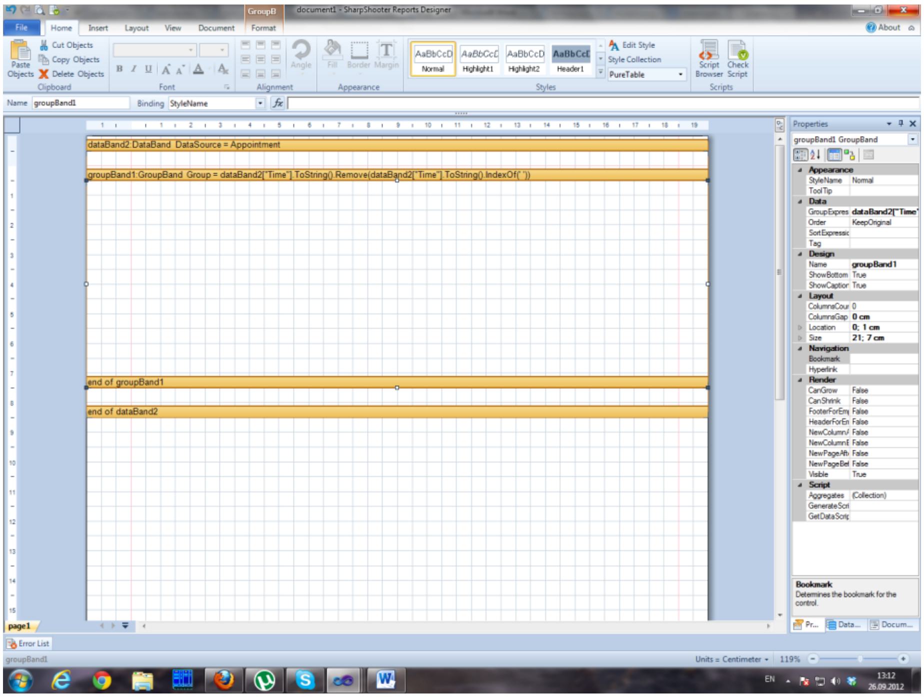
```
(DateTime)GetData("Appointment.Time") >
(DateTime)GetData("dateTimePickerFrom.Value.Date") &&
(DateTime)GetData("Appointment.Time") <
(DateTime)GetData("dateTimePickerTo.Value.Date")
```

Don’t forget to sort records by time.

Sort->Expression :

```
GetData("Appointment.Time")
```

Now we can group appointments by dates using the GroupBand.



dataBand2:DataBand DataSource = Appointment

groupBand1:GroupBand Group = dataBand2["Time"].ToString().Remove(dataBand2["Time"].ToString().IndexOf(' '))

end of groupBand1

end of dataBand2

Name: groupBand1

Binding: StyleName

Properties: groupBand1 GroupBand

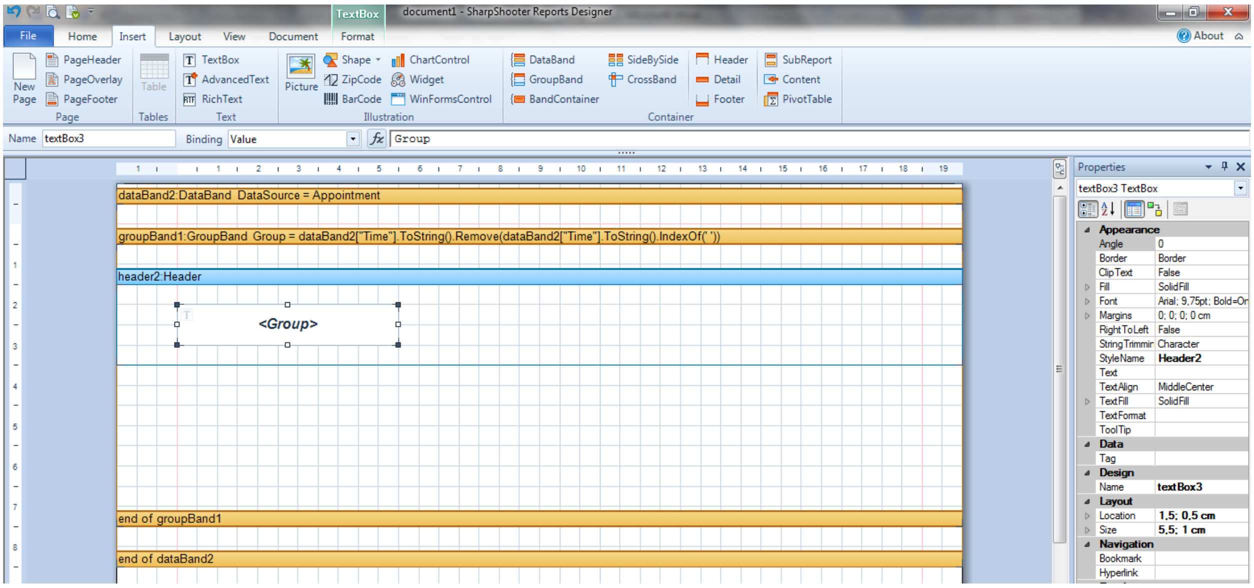
Appearance	
StyleName	Normal
ToolTip	
Data	
GroupExpres	dataBand2["Time"
Order	KeepOriginal
SortExpressic	
Tag	
Design	
Name	groupBand1
ShowBottom	True
ShowCaptior	True
Layout	
ColumnsCour	0
ColumnsGap	0 cm
Location	0; 1 cm
Size	21; 7 cm
Navigation	
Bookmark	
Hyperlink	
Render	
CanGrow	False
CanShrink	False
FooterForEm	False
HeaderForEn	False
NewColumnA	False
NewColumnE	False
NewPageAft	False
NewPageBel	False
Visible	True
Script	
Aggregates	(Collection)
GenerateScri	
GetDataScrip	

Bookmark
Determines the bookmark for the control.

GroupExpression:

```
dataBand2["Time"].ToString().Remove(dataBand2["Time"].ToString().IndexOf(' '))
```

To be simpler, we take text presentation of date as an criterion where we take off the part after space, that is Time.



dataBand2:DataBand DataSource = Appointment

groupBand1:GroupBand Group = dataBand2["Time"].ToString().Remove(dataBand2["Time"].ToString().IndexOf(' '))

header2:Header

<Group>

end of groupBand1

end of dataBand2

Name: textBox3

Binding: Value

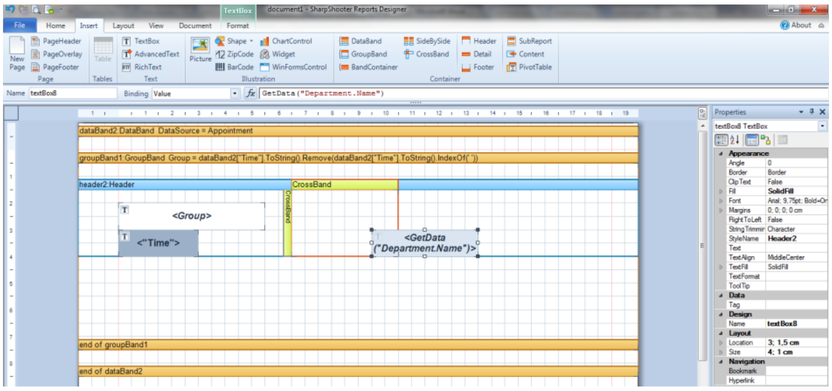
fx: Group

Properties: textBox3 TextBox

Appearance	
Angle	0
Border	Border
ClipText	False
Fill	SolidFill
Font	Arial; 9,75pt; Bold=Or
Margins	0; 0; 0; 0 cm
RightToLeft	False
StringTrimmir	Character
StyleName	Header2
Text	
TextAlign	MiddleCenter
TextFill	SolidFill
TextFormat	
ToolTip	
Data	
Tag	
Design	
Name	textBox3
Layout	
Location	1,5; 0,5 cm
Size	5,5; 1 cm
Navigation	
Bookmark	
Hyperlink	

In the header of the group, we display the criterion itself – the date. The keyword of the current criterion is the Group. And not to disparate the groups, we need to set the Order property to Ascending.

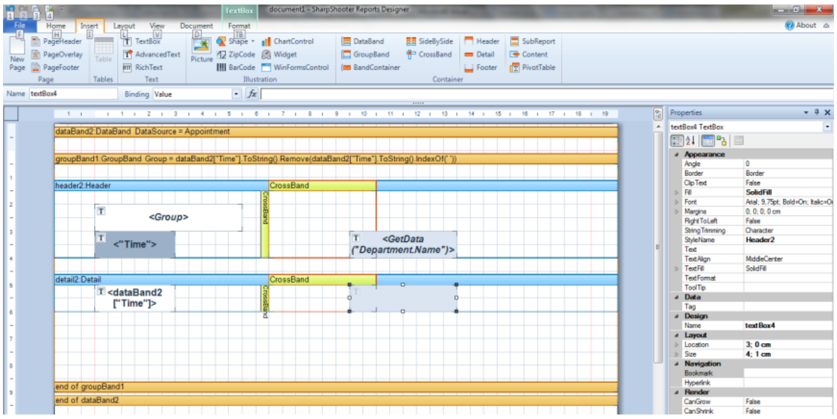
We also put the “Time” header and the CrossBand element that will display names of departments in a line. The DataSourse for CrossBand will be Department. We put TextBox bound to the department name inside CrossBand.


dataBand2:DataBand DataSource = Appointment
groupBand1:GroupBand Group = dataBand2["Time"].ToString().Remove(dataBand2["Time"].ToString().IndexOf(' '))
header2:Header
CrossBand
<Group>
<"Time">
<GetData ("Department.Name")>
end of groupBand1
end of dataBand2
Properties
textBox8 TextBox

The trick of such position of TextBox is that the width of the CrossBand should be equal to TextBox, but at the same time there should be some space for the «Time» title. And location of CrossBand horizontally does not affect anything.

Now we just need to display the record itself.

Add the Detail element and put TextBox to display time of the current record and CrossBand to display text of an appointment inside the Detail.


dataBand2:DataBand DataSource = Appointment
groupBand1:GroupBand Group = dataBand2["Time"].ToString().Remove(dataBand2["Time"].ToString().IndexOf(' '))
header2:Header
CrossBand
<Group>
<"Time">
<GetData ("Department.Name")>
detail2:Detail
CrossBand
<dataBand2 ["Time"]>
end of groupBand1
end of dataBand2
Properties
textBox4 TextBox

To display time, we bind TextBox to “dataBand2["Time"]” value.

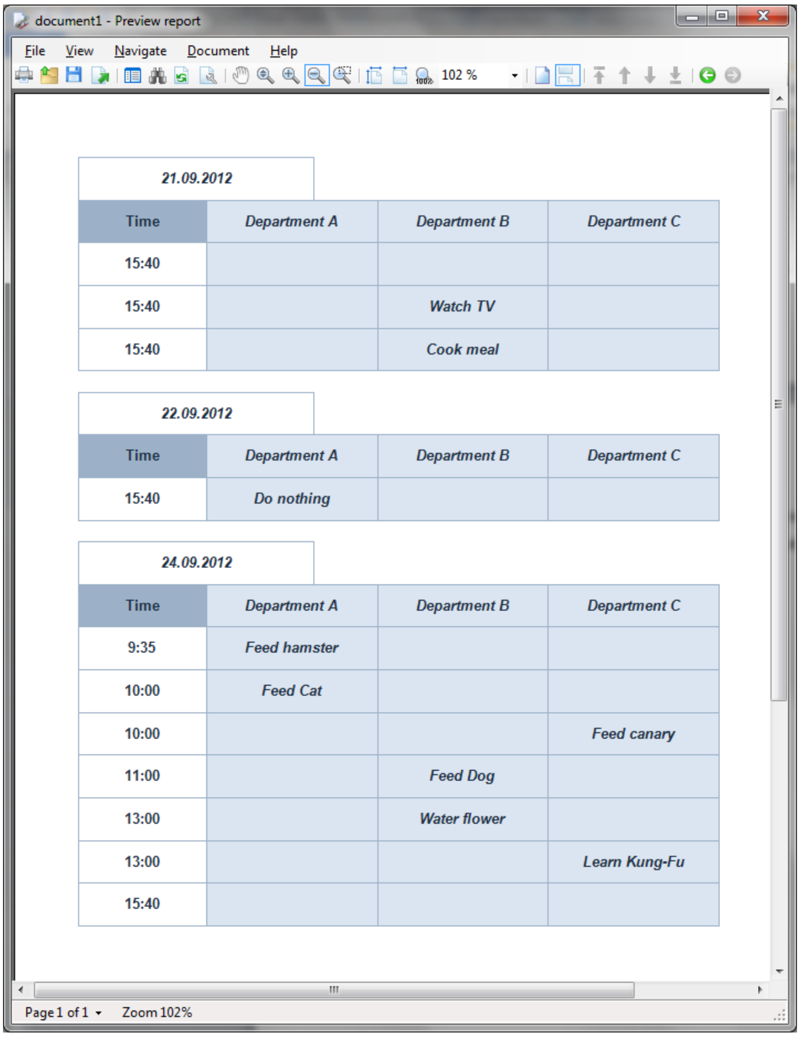
To display appointment text we need to set DataSourse of a new CrossBand toDepartment, add TextBox into it, and then we don’t need to bind it but to set script in the GenerateScript property.

GenerateScript:

```
if(
   textBox4.Value == "" &&
(int) dataBand2["Department"] == (int)
GetData("Department.DepartmentID")
){
   textBox4.Value = dataBand2["Text"];
}else{
   textBox4.Value = "";
}
```

This script sets appointment text only in a column that belongs to the corresponding department. In opposite case, we keep the line empty.

That’s all! Run!



21.09.2012

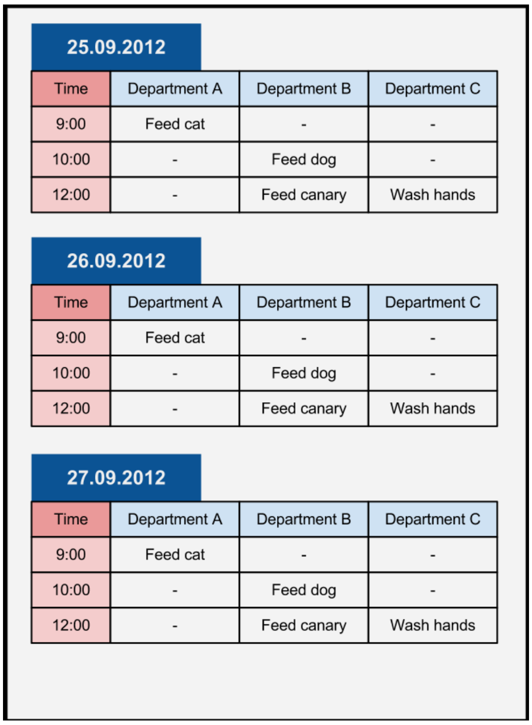
Time	Department A	Department B	Department C
15:40			
15:40		Watch TV	
15:40		Cook meal	

22.09.2012

Time	Department A	Department B	Department C
15:40	Do nothing		

24.09.2012

Time	Department A	Department B	Department C
9:35	Feed hamster		
10:00	Feed Cat		
10:00			Feed canary
11:00		Feed Dog	
13:00		Water flower	
13:00			Learn Kung-Fu
15:40			



25.09.2012

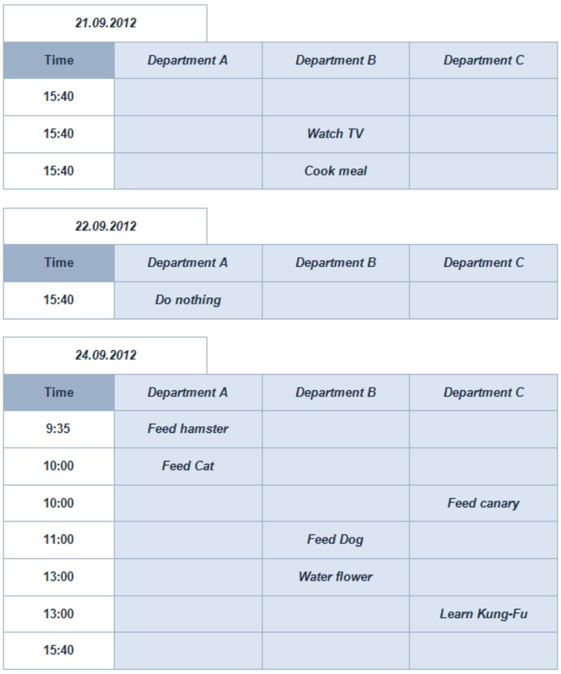
Time	Department A	Department B	Department C
9:00	Feed cat	-	-
10:00	-	Feed dog	-
12:00	-	Feed canary	Wash hands

26.09.2012

Time	Department A	Department B	Department C
9:00	Feed cat	-	-
10:00	-	Feed dog	-
12:00	-	Feed canary	Wash hands

27.09.2012

Time	Department A	Department B	Department C
9:00	Feed cat	-	-
10:00	-	Feed dog	-
12:00	-	Feed canary	Wash hands



21.09.2012

Time	Department A	Department B	Department C
15:40			
15:40		Watch TV	
15:40		Cook meal	

22.09.2012

Time	Department A	Department B	Department C
15:40	Do nothing		

24.09.2012

Time	Department A	Department B	Department C
9:35	Feed hamster		
10:00	Feed Cat		
10:00			Feed canary
11:00		Feed Dog	
13:00		Water flower	
13:00			Learn Kung-Fu
15:40			

The final report is quite close to what we planned to do.

But would be nice to put appointments of different departments with the same tome into one line.

Besides, several appointments of one department assigned at the same time could look better if we put them into one field.

How to do it? You will know in the next part.

To be continued…