

Creating Daily Schedule Reports using SharpShooter Reports (Part III)

By Maxim Edapin, a software developer at Perpetuum Software LLC



Introduction:

In the first two parts of the articles series devoted to the creation of various daily schedules we start from building a simple Windows Forms application and created two schedules:

- 1. A simple schedule of appointments for every department (Part 1)
- 2. A schedule of appointments grouped by date and displayed in parallel for every department (Part II).

But there is always room for perfection – now let's create a really professional schedule where the appointments are grouped by time and departments and displayed in one field if department and time coincide.

Prerequisites:

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

Schedule Nº3

Let's build an advanced schedule where all appointments for the same date, time and department will be displayed in one field.

25.09.2012							
Time	Department A	Department B	Department C				
9:00	Feed cat	-	Sleep / Snort				
10:00	-	Feed dog	-				
12:00	-	Feed canary	Wash hands				
26.09	9.2012						
Time	Department A	Department B	Department C				
9:00	Feed cat	-	-				
10:00	-	Feed dog	-				
12:00	Sleep / Snort	Feed canary	Wash hands				
27.09.2012							
Time	Department A	Department B	Department C				
9:00	Feed cat	-	-				
9:00 10:00	Feed cat	- Feed dog	-				

Let's formalize the task:

- 1. We need to display all days within some period.
- 2. For every day we need to display the date and:



- a. «Time» title and a column with time sorted in ascending order.
- b. Titles with names for every department and appointments assigned to them by time.
- 3. Appointments text is displayed in line for every unique time record.
- 4. Appointments with the same time and department are displayed in one field.

To solve this task the existing data sources are not enough, at least as they are. We won't change database structure, but will add some tables into the supplementary DataSet instead.

Members:		Tim	nePointTable <u>P</u> roperti	es:		
0 DayTable / Data					*	
1 TimePointTable		⊳	(ApplicationSetting:			
2 MessageTable			Columns	(Collection)		
			Constraints	(Collection)		
			DisplayExpression			
			MinimumCapacity	50		
			Namespace		Ε	
			Prefix			
			PrimaryKey	DataColumn[]	-	
			TableName	TimePointTable		
		⊿	Design			
x			(Name)	TimePointTable		
			GenerateMember	True		
			Modifiers	Private		
<u>A</u> dd		۵	Misc		-	
					_	

TimePointTable - will include unique values of appointments time.

Columns Collection Editor		-	_		? ×		
Members: Time Properties:							
0 TimePointID		⊿ Da	ata		•		
1 Time		⊳ (A	pplicationSetting:				
		AI	lowDBNull	True			
		Au	utoIncrement	False			
		Au	utoIncrementSeed	0			
		Au	utoIncrementStep	1	=		
		Ca	aption	Time			
		Co	olumnName	Time			
		Da	ataType	System.DateTime			
		Da	ateTimeMode	UnspecifiedLocal			
		De	efaultValue	<dbnull></dbnull>			
		Ex	pression				
	X	M	laxLength	-1			
		Na	amespace				
Add		Pr	efix		-		
				CI	ose		

MessageTable – will include field values for every department and unique time value.



Columns Collection Editor			and the Party of the	?	x
Members:	N	Mes	sageID <u>P</u> roperties:		
0 MessageID		۵	Data		*
1 Text		⊳	(ApplicationSetting:		
2 Department			AllowDBNull	True	
3 TimePoint			AutoIncrement	False	
			AutoIncrementSeed	0	
			AutoIncrementStep	1	=
			Caption	MessageID	
			ColumnName	MessageID	
			DataType	System.String	
			DateTimeMode	UnspecifiedLocal	
			DefaultValue	<dbnull></dbnull>	
			Expression		
×	<		MaxLength	-1	
	<u> </u>		Namespace		
Add			Prefix		-
				Close	

Don't forget to set the AutoIncrement property for the primary keys!

The relation of the Message to TimePoint will also be useful. Let's create it and call the Messages.



To fill these tables I created the following function:

private v	<pre>roid GenerateTimePoints()</pre>
í	<pre>//Group all tasks by time var TimeGroups = from app in calendarDatabaseDataSet1.Appointment group app by app.Time;</pre>
	<pre>foreach (var TimeGroup in TimeGroups) {</pre>
	<pre>//Create the TimePoint for every group DataRow tp_row = TimePointTable.NewRow(); //New record tp_row["Time"] = TimeGroup.Key; //Unique time - grouping criterion TimePointTable.Rows.Add(tp_row); //Adding to a table</pre>
	<pre>//Group appointments by departments var DepGroups = from app in TimeGroup group app by app.Department;</pre>
	<pre>//Go through all existing departments for (int i = 0; i < calendarDatabaseDataSet1.Department.Count; i++) {</pre>
department	<pre>int DepartmentID = calendarDatabaseDataSet1.Department[i].DepartmentID;//Primary key of the current</pre>
	<pre>//Create the Message record for every department DataRow msg_row = MessageTable.NewRow(); msg_row["Department"] = DepartmentID; //Current department msg_row["TimePoint"] = tp_row["TimePointID"]; //Current unique time</pre>
	<pre>//Check if there is an appointment for current department at current unique time if (DepGroups.Any(p => p.Key == DepartmentID)) { //There is! //Take a group of appointments var DepGroup = DepGroups.First(p => p.Key == DepartmentID); //Unite all appointment descriptions into one field DepGroup.ToList().ForEach(p => msg_row["Text"] += p.Text + ". "); } else { //No. //Keep the dash msg_row["Text"] = "-"; } //Add the resulting Message record to a table MessageTable.Rows.Add(msg_row);</pre>
}	} }

So, now we just need to invoke this function before report generation and new data sources will be built.

Add them to the report.

Edit obj	ect name				
Name	TimePoint				
Value	TimePointTable	•	DataSource	e only	
			[ОК	Cancel
Edit obj	ect name				
Edit obj	ect name Message				
Edit obj Name Value	ect name <mark>Message</mark> MessageTable	- I	Data Source	e only	

Now, let's move to a report template:



The first DataBand(dataBand1) goes through Days values.







The second DataBand (dataBand2) gies through TimePoint records.

For the dataBand2, let's put a Header with the current date and titles.

The titles are – the "Time" TextBox and the CrossBand that outputs departments names.



And finally the Detail element (detail1).

The detail1 contains:

- 1. TextBox that is bound to the «dataBand2["Time"]» value
- 2. CrossBand that includes the TextBox bound to the «GetData("TimePoint.Messages.Text")» value.



TextBox will bring a time value of the current TimePoint, and CrossBand will bring all messages which belong to the current TimePoint.

Ok, now we can run it!

				25.0	9.2012		
21 00 20	12			Time	Department A	Department B	Department
21.09.20				9:00	Feed cat	•	Sleep / Sno
Time	Department A	Department B	Department C	10:00		Feed dog	
15:40	Play golf.	Watch TV. Cook meal.	-	12:00		Feed canary	Wash hand
22.09.20	12			26.0	9.2012		
Time	Department A	Department B	Department C	Time	Department A	Department B	Department
15:40	Do nothina.			9:00	Feed cat		
	2 c richning.			10:00		Feed dog	•
24.09.20	12			12:00	Sleep / Snort	Feed canary	Wash hand
Time	Department A	Department B	Department C	27.0	9.2012		
9:35	Feed hamster.	-	-	Time	Department A	Department B	Department
10:00	Feed Cat.	-	Feed canary.	9:00	Feed cat		
11:00		Feed Dog.	-	10:00		Feed dog	-
13:00	-	Water flower.	Learn Kung-Fu.	12:00		Feed canary	Wash hand
15:40	Iron shirts.		-				

Not bad!

As you can see, all records with the same time are grouped and displayed in one line and even in one field when and if needed!

21.09.2	012		
Time	Department A	Department B	Department C
<mark>1</mark> 5:40	Play golf.	Watch TV. Cook meal.	-
22.09.2	012		
Time	Department A	Department B	Department C
<mark>1</mark> 5:40	Do nothing.	-	-
24.09.20	012		
Time	Department A	Department B	Department C
<mark>9:35</mark>	Feed hamster.	-	-
10:00	Feed Cat.	-	Feed canary.
11:00	-	Feed Dog.	-
13:00	-	Water flower.	Learn Kung-Fu.
15:40	Iron shirts.	-	-

This is a really good schedule that we can print and use every day!