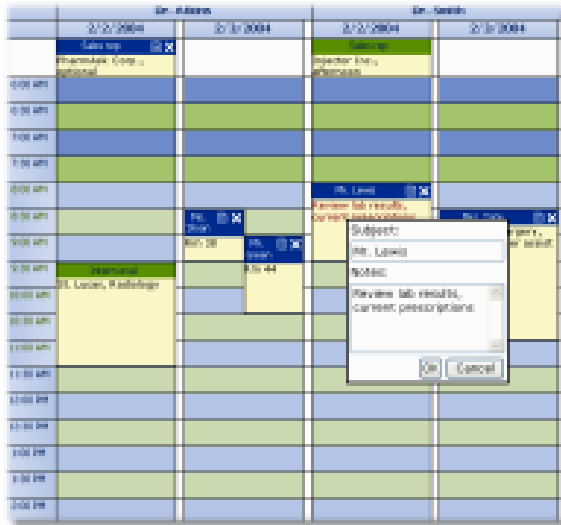


How to Create a Web-based Calendaring Solution using Visual Web Developer

Calendars are extremely useful for scenarios that include date picking and the organized display of events. ASP.NET provides a Calendar control that you can use to display the days of a month and select individual days, weeks, or the entire month at once. The Calendar control does not, however, provide an easy, elegant, or customizable way to create a calendar that is capable of displaying and editing events. Nor does it allow you to easily bind to a data source. In this article, we will take a look at an alternative solution offered by [TMSsoftware](http://www.tmssoftware.com) called WebPlanner.



Screenshot - WebPlanner control in action

From the WebPlanner Developers Guide:

“WebPlanner provides a first-class ASP.NET control for implementing a broad range of planning and scheduling solutions. Whether a project requires the creation of a single-user Personal Information Management (PIM) application or time planning for multiple resources such as hotel rooms, rental cars, and university courses, WebPlanner provides an open, highly-configurable interface that will suit the project's needs.”

Contents

[WebPlanner setup and configuration](#)

[Introduction to WebPlanner](#)

[Configuring and controlling the planner](#)

[Working with data](#)

[References and additional resources](#)

WebPlanner setup and configuration

Your first task is to download and install the WebPlanner software which is made available to you when you register your Visual Studio Express product. The registration process can be initiated by selecting **Help | Register Product** from the main menu of your Express product. After you register, you will receive an email from Microsoft with a link to the registration benefits portal site where you will find the WebPlanner download. Downloading and installing the WebPlanner software is quick and easy.

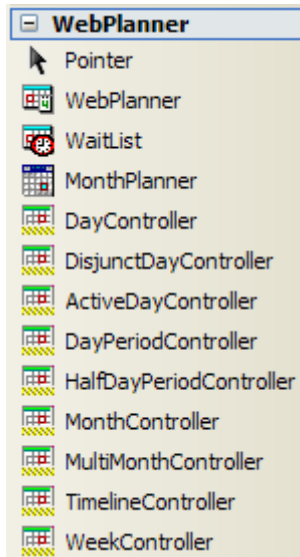


TMSsoftware — An ASP.NET 2.0 Web scheduling component offering a fully functional control for day, week, month and timeline rendering.

Screenshot - The Registration Benefits Portal site lists your benefits and provides download links

After WebPlanner has been installed, you can add the associated components to your Visual Web Developer toolbox. Start by loading a new instance of Visual Web Developer 2005 Express.

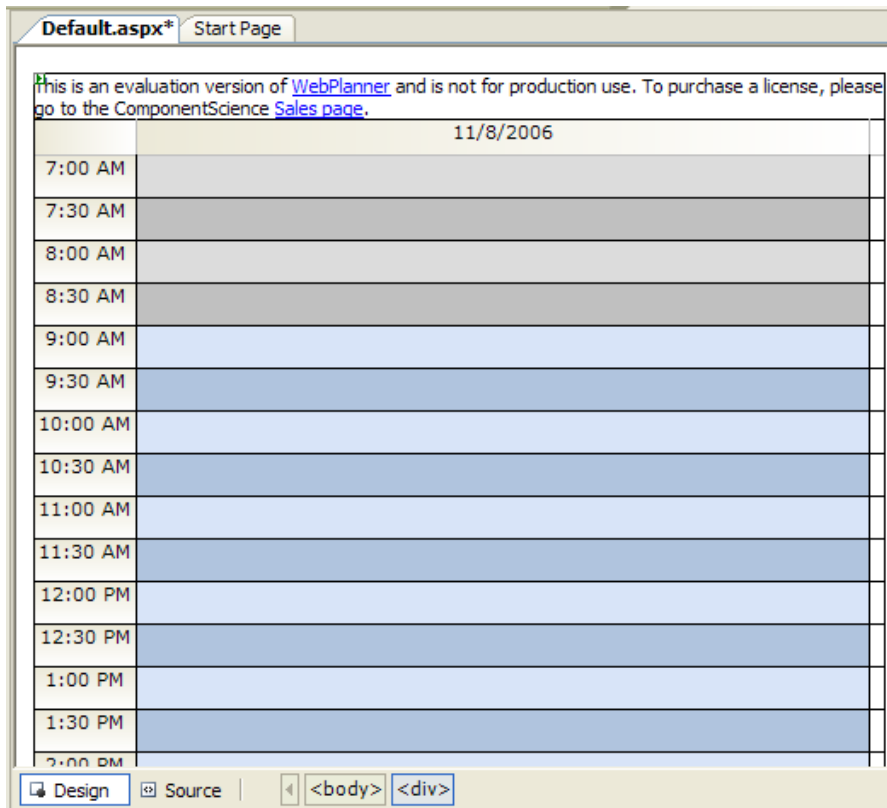
1. Select **File | New Web Site** from the main menu.
2. Select the **ASP.NET Web Site** template followed by the **OK** button.
3. Right-click within the **Toolbox** and select **Add Tab** from the context menu.
4. Name the new tab **WebPlanner**.
5. Right-click within the new **WebPlanner tab** and select **Choose Items**.
6. Click the **Browse** button and navigate to the location of **WebPlanner2005.dll**.
7. Click the **Open** button to add WebPlanner components to the list of available .NET Framework components.
8. Select all components that are part of the **TMS.WebPlanner** namespace followed by the **OK** button.



Screenshot - WebPlanner toolbox

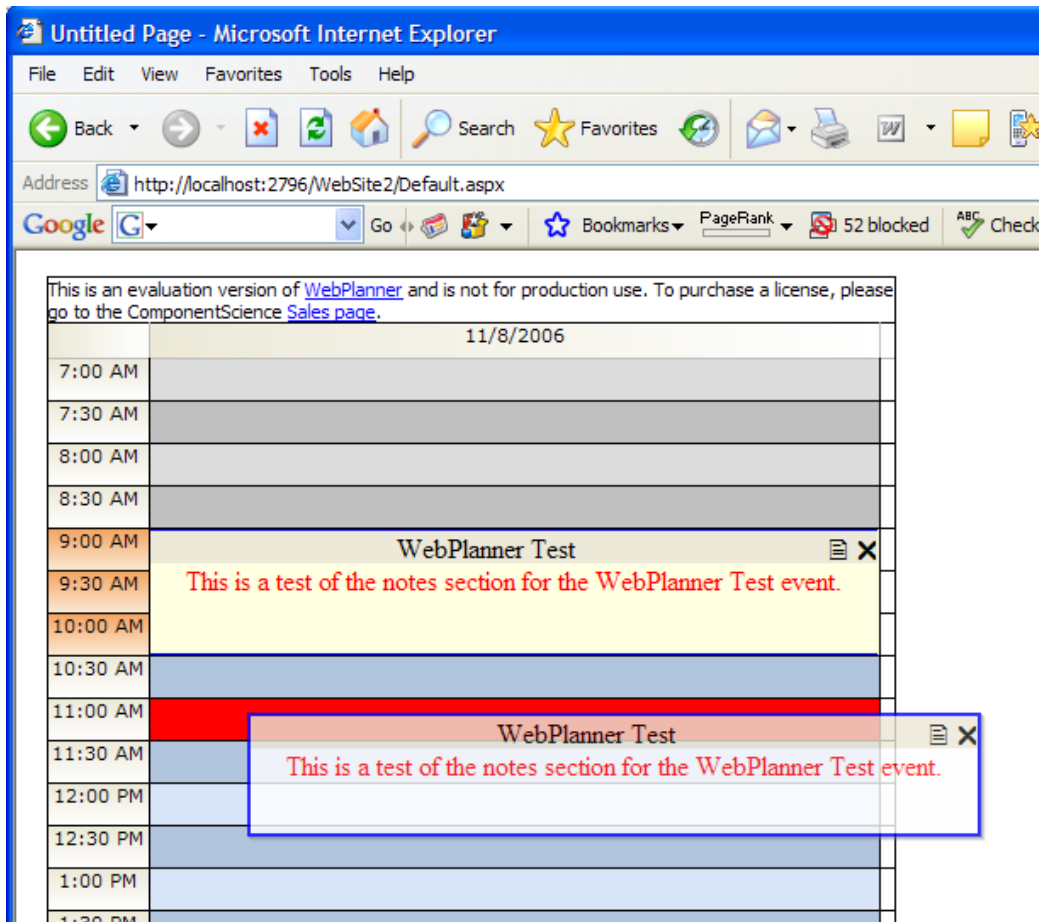
Introduction to WebPlanner

Let's start by looking at the default functionality provided by the WebPlanner control. In Visual Web Developer, switch to the Design view for the default page and drag an instance of the WebPlanner control from the toolbox onto the blank design surface. This will create a planner that shows the current date in a single column with rows representing half-hour blocks of time.



Screenshot - Default view for WebPlanner control

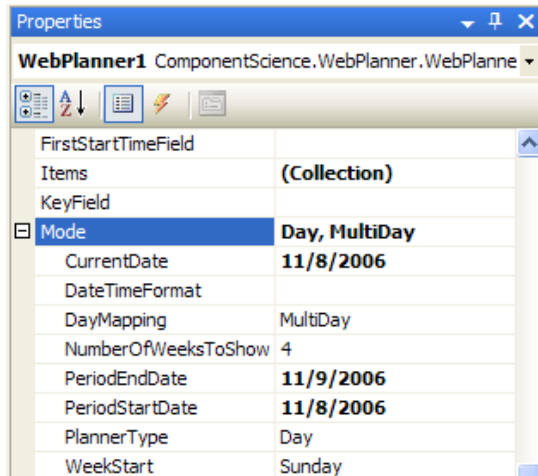
Select **Debug | Start Debugging** to take a look at the default WebPlanner control in action. Take a few moments to add events to the calendar and interactively resize and reorganize them by dragging and dropping with the mouse. The screenshot below shows an entry being moved from one time slot to another.



Screenshot - WebPlanner control supports drag and drop

Configuring and controlling the planner

Close Internet Explorer and return to Visual Web Designer. If you select the instance of WebPlanner and then view the Properties window, you will see a large number of configurable options that allow you to define the appearance, behavior, and underlying data used. Find and expand the Mode node located in the Data section of the Properties window. This node allows you to configure dates, formatting, and the planner type.



Screenshot - WebPlanner control properties

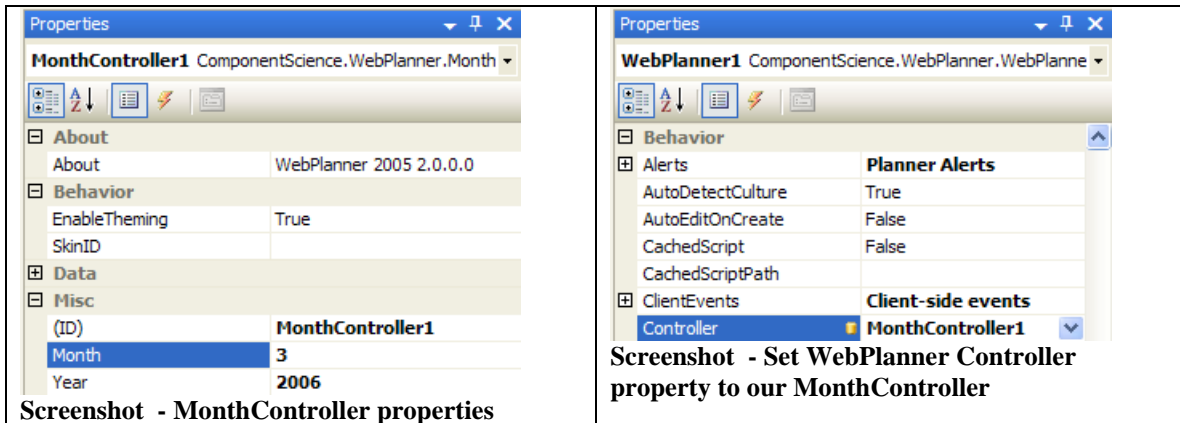
Change the PlannerType property to Month to make each of the rows represent individual days of a month. Feel free to experiment with other available property options to get a better idea what you can customize.

This is an evaluation version of [WebPlanner](#) and is not for production use. To purchase a license, please go to the ComponentScience [Sales page](#).

	Default resource
11/1	
11/2	
11/3	
11/4	
11/5	

Screenshot - WebPlanner in Month mode

To make the configuration of the WebPlanner Mode easier, a number of Controller components can be used. These controllers match up with each of the PlannerType property options that you just saw – for example there is a DayController and a MonthController. Go ahead and drag and drop a MonthController from your toolbox to your default form and change the current month. To connect the controller to the planner, you will need to set the Controller property for the planner.



Working with data

A calendar is not very useful unless the events on it can be saved and loaded. In this section, we will show how easy it is to hook a WebPlanner up to a database to allow a user to view, insert, and delete event records.

From the WebPlanner Developers Guide:

“For ASP.NET 2.0, the WebPlanner utilizes the new DataSource components for seamless databinding. It is sufficient to create an OleDbDataSource or SqlClientDataSource with defined INSERT, UPDATE, DELETE commands and the WebPlanner can use it without code to read, create & update events from a database.”

In order to use your database with a WebPlanner control, the following requirements must be satisfied:

- The scheduled events must be stored in a single table.
- The scheduled events must be uniquely identified via some type of key field.
- Each scheduled event must have a start time and an end time.

Create a new database:

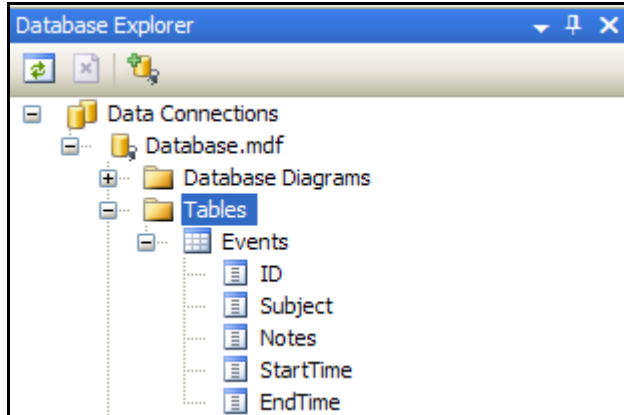
1. Right-click on the **App_Data** folder and select **Add New Item**.
2. Select the **SQL Database** template followed by the **Add** button.
3. In **Database Explorer**, right-click the **Tables** node for the new database and select **Add New Table**.
4. Add the columns with data types as shown in the following screenshot:

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Subject	text	<input checked="" type="checkbox"/>
Notes	text	<input checked="" type="checkbox"/>
StartTime	datetime	<input type="checkbox"/>
EndTime	datetime	<input type="checkbox"/>

Note: set the ID column as the primary key by right-clicking on it and selecting **Set Primary Key**.

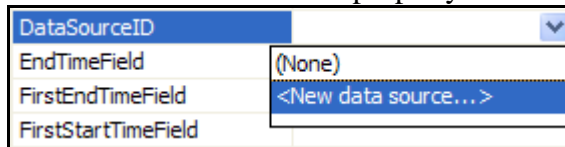
5. Save the new table by selecting **File | Save Table1** from the main menu.

6. When you are asked to choose a name for the table, enter **Events**.
7. You should now have a table named Events as shown below:

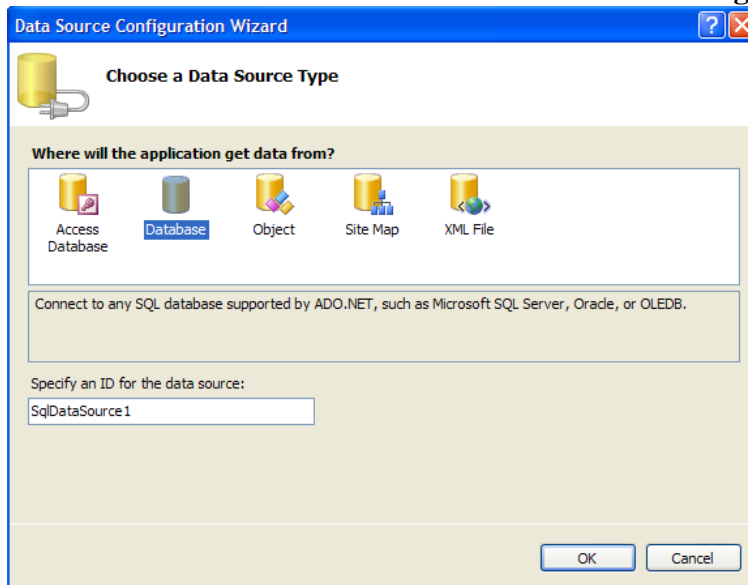


Create a new data source and link it to the WebPlanner:

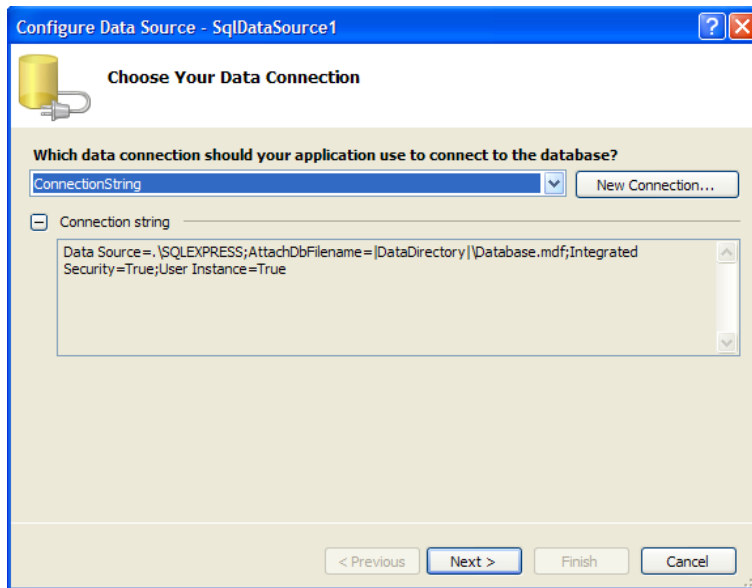
1. Go to the **Properties** window for your WebPlanner instance.
2. Locate the **DataSourceID** property and click the drop down box.



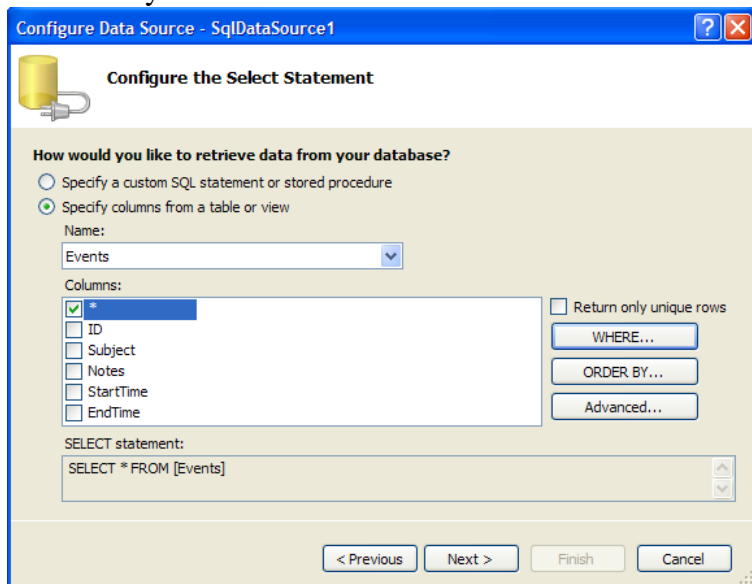
3. Select **New Data Source** to load the **Data Source Configuration Wizard**.



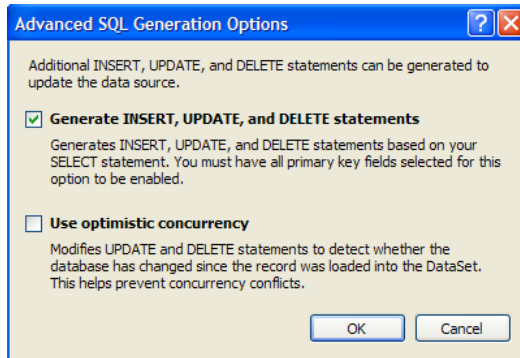
4. Select **Database** for the source type and select **OK** to continue.



5. Select the **ConnectionString** connection from the provided drop down box followed by **Next**.



6. Select the “*” column so that the SELECT statement reads “SELECT * FROM [Events]”.
7. Click the **Advanced** button to load the **Advanced SQL Generation Options** window.



8. Select the **“Generate INSERT, UPDATE, and DELETE statements”** check box followed by the **OK** button to return to the Configure Data Source wizard.
9. Click the **Next** button to continue followed by the **Finish** button to complete the configuration wizard.

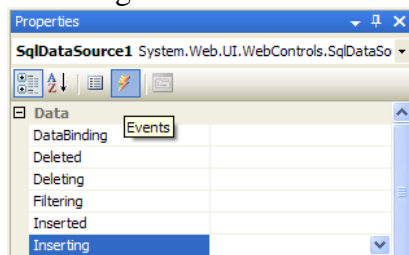
Note: You may notice that the WebPlanner control indicates that there is an error with the new database. This problem only affects the design-time view and will not be an issue when you actually run the Web site.

Set the necessary data field mappings for WebPlanner:

1. Go to the **Properties** window for your **WebPlanner** instance.
2. Set each of the following WebPlanner properties to map to the appropriate database column:
 - **KeyField** maps to **ID**
 - **NotesField** maps to **Notes**
 - **SubjectField** maps to **Subject**
 - **StartTimeField** maps to **StartTime**
 - **EndTimeField** maps to **EndTime**

Set the ID database parameter to a unique value during inserts:

1. Go to the **Properties** window for your **SqlDataSource** instance.
2. Click the yellow lightning icon to view the available events as shown in the following screenshot.



3. Double-click the **Inserting** event to have Visual Web Designer generate a method to handle the event. This will load the code-behind file for your default page.
4. For C#, add the following field to the page class:

```
static int currentID = 0;
```

For VB.NET, add the following field to the **Inserting** event handler method rather than at the class level:

```
Static Dim currentID As Integer = 0
```

Note: we need a way to generate a unique number to identify each event that is stored in our database. The *currentID* field will be incremented each time an event is added to the database.

5. Add the following line of code to the **Inserting** event handler:

C#:

```
e.Command.Parameters["@ID"].Value = currentID++;
```

VB.NET:

```
e.Command.Parameters("@ID").Value = currentID
currentID += 1
```

Note: this solution is meant to provide a quick demonstration and will only work when the database is blank and from a single browser instance. In order to create a more realistic solution you would need to assign the ID value within the database itself, possibly by using a stored procedure for insertions.

Now that you have finished setting up the demonstration Web site, go ahead and run the application to see how things work. The following screenshot shows an instance of the application with two events. If you close the browser and start a new instance you will see that the events have been persisted to the database.

This is an evaluation version of [WebPlanner](#) and is not for production use. To purchase a license, please go to the [ComponentScience Sales page](#).

	Default resource	
3/1		
3/2		
3/3	Found an alien in my bathtub	☰ ✕
3/4		
3/5	Lost my left shoe while riding the bus	☰ ✕
3/6		
3/7		
3/8		
3/9		
3/10		

Screenshot - Example application using WebPlanner data bound to a database file

References and additional resources

For further documentation, sample code, and product information visit:

<http://www.tmssoftware.com>

Note: you need to register (for free) for an account before you can download the sample code. Also be aware that not all of the sample code is meant for use with WebPlanner 2005 – most of it was created for the previous version. See the support forums for help converting existing code to the new version.