
ComponentOne

Scheduler for WinForms

ComponentOne, a division of GrapeCity

201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206 USA

Website: <http://www.componentone.com>

Sales: sales@componentone.com

Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Trademarks

The ComponentOne product name is a trademark and ComponentOne is a registered trademark of GrapeCity, Inc. All other trademarks used herein are the properties of their respective owners.

Warranty

ComponentOne warrants that the media on which the software is delivered is free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective media to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for the defective media by sending it and a check for \$25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original media on which the software is delivered is set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. ComponentOne is not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only.

Table of Contents

Scheduler for WinForms Overview	4
Help with WinForms Edition	4
Key Features	5-7
Scheduler for WinForms Quick Start	8
Step 1 of 3: Creating the Scheduling Application	8-10
Step 2 of 3: Binding to a Data Source	10-16
Step 3 of 3: Customizing the Calendar Settings	16-20
Scheduler for WinForms Top Tips	21-24
Design-Time Support	25
Scheduler for WinForms Smart Tags	25
C1Schedule Tasks Menu	25-32
C1Calendar Tasks Menu	32-34
Scheduler for WinForms Smart Designers	34-35
C1Schedule Smart Designer	35-39
C1Calendar Smart Designer	39-41
Working with Scheduler for WinForms	42
Appointments	42-45
Labels	45
Assigning Predefined Labels to an Appointment	45-47
Assigning Custom Labels to an Appointment	47-48
Availability	48-49
Assigning Predefined Availabilities to an Appointment	49-50
Reminders	50-51
Reminder Properties	52-53
Contacts	53
Assigning Contacts to an Appointment	53-54
Adding Contacts to the Master Contact List	54-55
Categories	55
Assigning Predefined Categories to an Appointment	55-57
Adding Custom Categories to the Master Category List	57-58
Resources	59
Assigning Resources to an Appointment	59-61
Adding Resources to the Master Resource List	61-62
Appointment Actions	62-63

Data Views	63-68
Changing the Default Data View	68-69
Data Mappings	69-71
Localization	71-72
Localize Dialog Box	72-73
Localization Toolbar	73-77
Setting the Current Culture	77-78
Switching Resources at Run Time	78
Grouping	78-79
Showing Working Time From One Day to Another	79
Customizing Scheduler for WinForms' Appearance	80
Visual Styles	80-88
Setting a Predefined Visual Style	88-90
C1Calendar Visual Style Settings	90
Common Node	90-91
Navigation Buttons Node	91-92
Month Header Node	92-94
Day Header Node	94-95
Days Node	95-97
C1Schedule Visual Style Settings	97
Common Node	97-98
Title Node	98-100
Navigation Buttons Node	100-101
Appointment Node	101-102
Palette Node	102-103
Navigation Panel Node	103-104
Current Date Node	104
Day View Node	104-106
Time Ruler Node	106-108
Current Time Node	108-109
All-Day Area Node	109
Day Header Node (Day View)	109-110
Month View Node	110-112
Grid Header Node	112-113
Week Tab Node	113-114
Day Header Node	114-115

Run-Time Interaction	116
C1Schedule Run-Time Context Menu	116-119
Working with Appointments	119-122
Opening an Appointment	122
Saving an Appointment	122-123
Editing an Appointment	123
Deleting an Appointment	123-124
Recurring Appointments	124-126
Mouse and Keyboard Navigation	126
Changing the Background Color	126-127
Navigating the Calendar	127-128
Printing and Previewing Data	128-129
Importing and Exporting Data	129
C1Calendar Interactivity	129
Changing the Month or Year	129-130
Changing the View	130-131
Using Mouse and Keyboard Navigation	131
Drag-and-Drop Support	131-133
Scheduler for WinForms Samples	134-136
Scheduler for WinForms Task-Based Help	137
Creating a Bindable Microsoft Access Database	137-138
Mapping the Microsoft Access Database to a C1Schedule Control	138-139
Creating a Custom Visual Style	139
Saving Custom Visual Style	139-140
Importing Saved Visual Style	140
Deleting a Custom Visual Style	140
Formatting the Border Style	140
Formatting the Border Style of the C1Schedule Control	140-142
Formatting the Border Style of the C1Calendar Control	142-144
Hiding the Week Numbers	144-146
Adding Importing and Exporting	146-148
Customizing Printing and Previewing	148
Disabling Printing	148-149
Disabling the Print Options Form	149-151
Disabling the Print Progress Form	151-152
Hiding Private Appointments	152

Scheduler for WinForms Overview

Easily integrate completely customized Microsoft Office style scheduling functionality into your applications with **Scheduler for WinForms**. Now you can build intuitive, polished-looking scheduling applications with ease using **Scheduler for WinForms'** built-in visual styles, data views, and appointment labels. With additional options like drag-and-drop support, customizable dialog boxes, and easy localization, designing a scheduling application to meet the needs of your end-users has never been easier.

Scheduler for WinForms includes two visual controls, [C1Schedule](#) and [C1Calendar](#), and a data component, [C1ScheduleStorage](#). C1Schedule is a fully functional Outlook-style schedule that allows users to add, edit, and manage their appointments easily. C1Calendar is a calendar control that enables users to select the current date or the date range to display appointment data in the C1Schedule control. And C1ScheduleStorage, created automatically by the C1Schedule control, handles all the behind the scenes data operations.



Getting Started

Get started with the following topics:

- [Key Features](#)
- [Quick Start](#)
- [Components](#)
- [Samples](#)

Help with WinForms Edition

Getting Started

For information on installing **ComponentOne Studio WinForms Edition**, licensing, technical support, namespaces and creating a project with the control, please visit [Getting Started with WinForms Edition](#).

Key Features

Scheduler for WinForms incorporates several key features, including the following:

- **Standard Data Binding or Built-in Data Source**

Scheduler for WinForms gives you the option of using standard ADO.NET data binding or using a built-in **DataSource**. Using the **Data Source settings**, which work with the [C1ScheduleStorage](#) component, you can attach a **DataSource** and map to each column in the table to save and load appointments, categories, contacts, labels, resources, and the status of appointments.

If you want to use a built-in **DataSource**, you can save or load data in any supported format (binary, XML, or iCal) when it is appropriate for your application. You can do it using the [C1ScheduleStorage](#) [Export](#) and [Import](#) methods. End-users can import and export data at any time by using the **Import/Export** items from the context menu of the [C1Schedule](#) control.

For additional information on binding [C1Schedule](#) to a data source or using a built-in data source, see [Binding to a Data Source](#) or [Importing and Exporting Data](#).

- **Appointments**

Users can easily add new and edit existing appointments within a [C1Schedule](#). Appointments can occur one-time or recur over a set amount of time, and reminders can be set so no appointment is missed. Additionally, [C1Schedule](#) provides twelve built-in labels and four availability options to help users manage each appointment, as well as the ability to create custom labels. Appointments can be organized within categories, and resources and contacts for each appointment can be specified. For more information on appointments, see [Appointments](#).

- **Built-in Data Views**

The [C1Schedule](#) control includes five data built-in data views, allowing you to offer a variety of ways for users to view their schedules. Having built-in data views simplifies development time by allowing you to set the view using one property, [ViewType](#), rather than having to use multiple controls to show each different view. The data views include the following:

View	Description
DayView	Displays a detailed view showing appointments for a particular day.
MonthView	Displays appointments for one or more months.
TimeLineView	Displays appointments in a horizontal time line.
WeekView	Displays appointments for specified work days.
WorkWeekView	Displays appointments for any given weekly period. The default is Monday through

View	Description
	Friday.

For more information on each of the data views, see [Data Views](#).

• Visual Styles

Scheduler for WinForms includes twelve built-in Visual Styles, as well as the capability to create customized Visual Styles at design time and save them to an XML file:

Visual Style	Description
Aero	Uses a light yellow palette.
Office 2007 Black	Mimics the Microsoft Office 2007 Black theme.
Office 2007 Blue	Mimics the Microsoft Office 2007 Blue theme.
Office 2007 Silver	Mimics the Microsoft Office 2007 Silver theme.
Royale	Mimics the Microsoft Windows Media Center Royale theme.
System	Uses the current desktop settings.
Windows XP Blue	Mimics the Microsoft Windows XP Blue theme.
Windows XP Olive	Mimics the Microsoft Windows XP Olive Green theme.
Windows XP Silver	Mimics the Microsoft Window XP Silver theme.
Yahoo	Uses a grey palette.

For more information on the built-in Visual Styles, see [Visual Styles](#).



Please note that Visual Styles - Aero, Royal, System, Yahoo, and Windows XP are marked with obsolete attribute. You are advised to use other modern visual styles available. You can also use themes such as Office 2010, Office 2013, etc. included into C1Themes assembly, see [Themes for WinForms](#) for more information.

• Office 2007 Support

Scheduler for WinForms includes three Office 2007-style Visual Styles, which by default have the BubbleLook property set to **True**:

Visual Style	Description
Office 2007 Black	Mimics the Microsoft Office 2007 Black theme.

Visual Style	Description
Office 2007 Blue	Mimics the Microsoft Office 2007 Blue theme.
Office 2007 Silver	Mimics the Microsoft Office 2007 Silver theme.

Additionally, [C1Schedule](#) also supports the Week View for Office 2007.

Scheduler for WinForms Quick Start

In this section you'll learn how to use the basic **Scheduler for WinForms** functionality to create a simple scheduling application. This section is not intended to be a comprehensive tutorial of all features of **Scheduler for WinForms**, but rather provide a quick introduction and highlight some general approaches to using the product.

In the following quick start guide, you'll create a simple scheduling application and customize its appearance, bind your scheduling application to a data source, customize the calendar, and explore some of the run-time functionality **Scheduler for WinForms** provides.

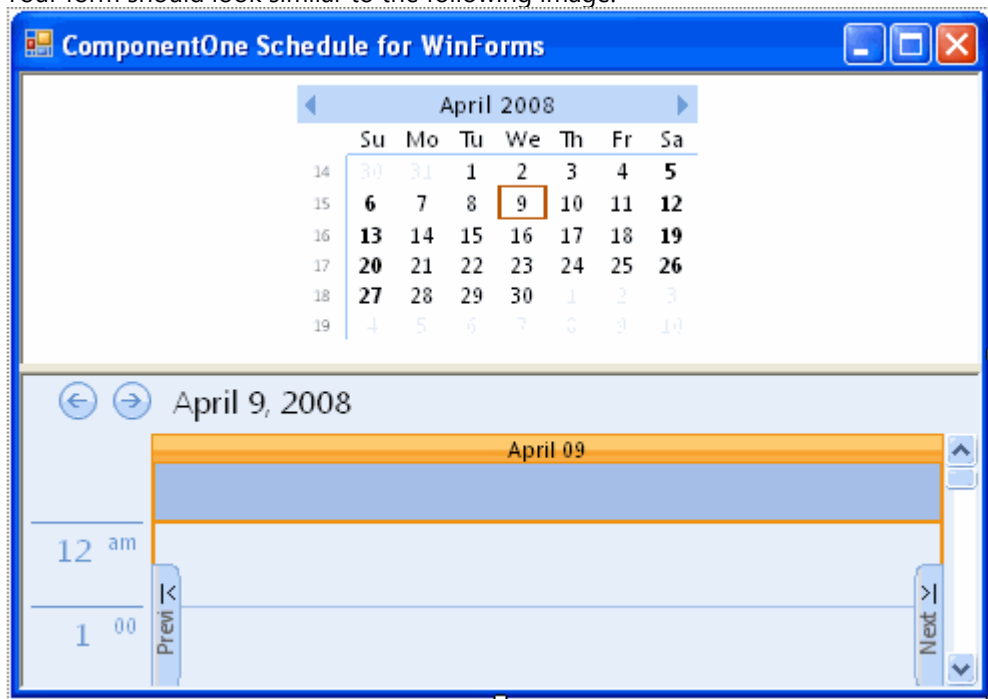
Step 1 of 3: Creating the Scheduling Application

In this step you'll create a simple scheduling application using the [C1Calendar](#) and [C1Schedule](#) controls. You'll then customize the appearance of your scheduling application using the incorporated Smart Designers and without adding any code to your project.

To begin, complete the following steps:

1. Create a new .NET project.
2. Add a **SplitContainer** control to the form and set its **Orientation** property to **Horizontal**.
3. If the **SplitContainer** is not docked on the form, set its **Dock** property to **Fill**.
4. Select **Panel1**, the top pane of the **SplitContainer**, and add a **C1Calendar** control to the form within the pane.
5. Set **C1Calendar1's Dock** property to **Fill**.
6. Select **Panel2**, the bottom pane of the **SplitContainer**, and add a **C1Schedule** control to the form within the pane.
7. Set **C1Schedule1's Dock** property to **Fill**.

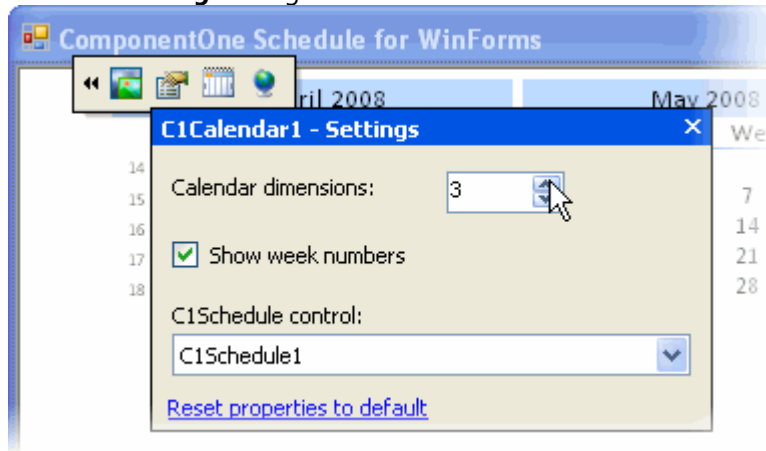
Your form should look similar to the following image:



8. Open the **C1Calendar Smart Designer** located near the top left of the **C1Calendar** control to set some of the **C1Calendar** settings. For more information on the **C1Calendar Smart Designer**, see [C1Calendar Smart Designer](#).
9. Click the **Settings** button on the **C1Calendar Smart Designer** to open the **Settings** dialog box and then complete the following:
 - Select **C1Schedule1** from the **C1Schedule control** drop-down list to link the **C1Calendar** control to the

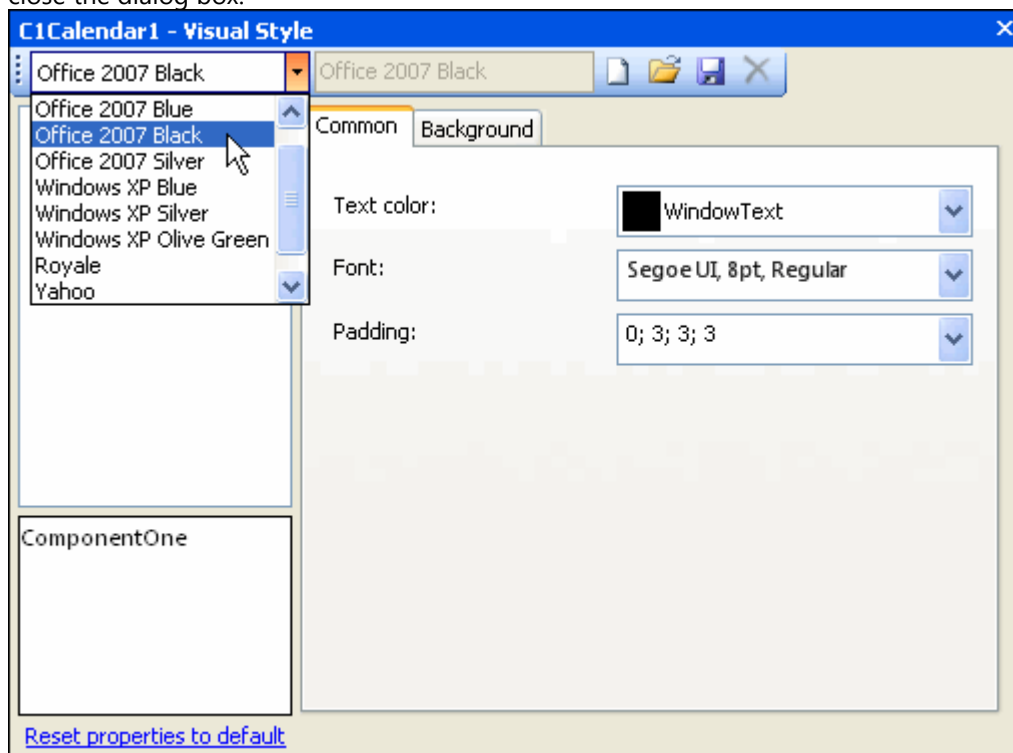
C1Schedule control.

- Set the **Calendar dimensions** box to **3**.
- Close the **Settings** dialog box.



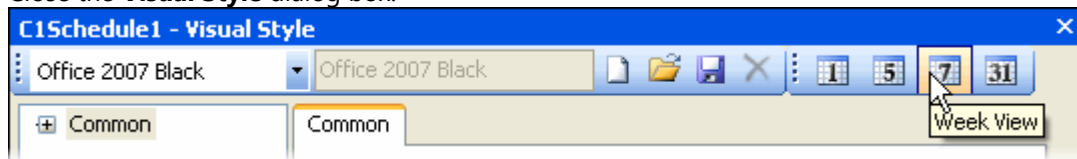
Note: The [CalendarDimensions](#) property can display only the number of months that it can fit into the available space.

10. Resize the form and the **SplitContainer** to accommodate the new **C1Calendar** dimensions.
11. Click the **Visual Style** button on the **C1Calendar Smart Designer** to open the **Visual Style** dialog box.
12. In the **Visual Style** dialog box, select **Office 2007 Black** from the **Current Visual Style** drop-down list, and close the dialog box.



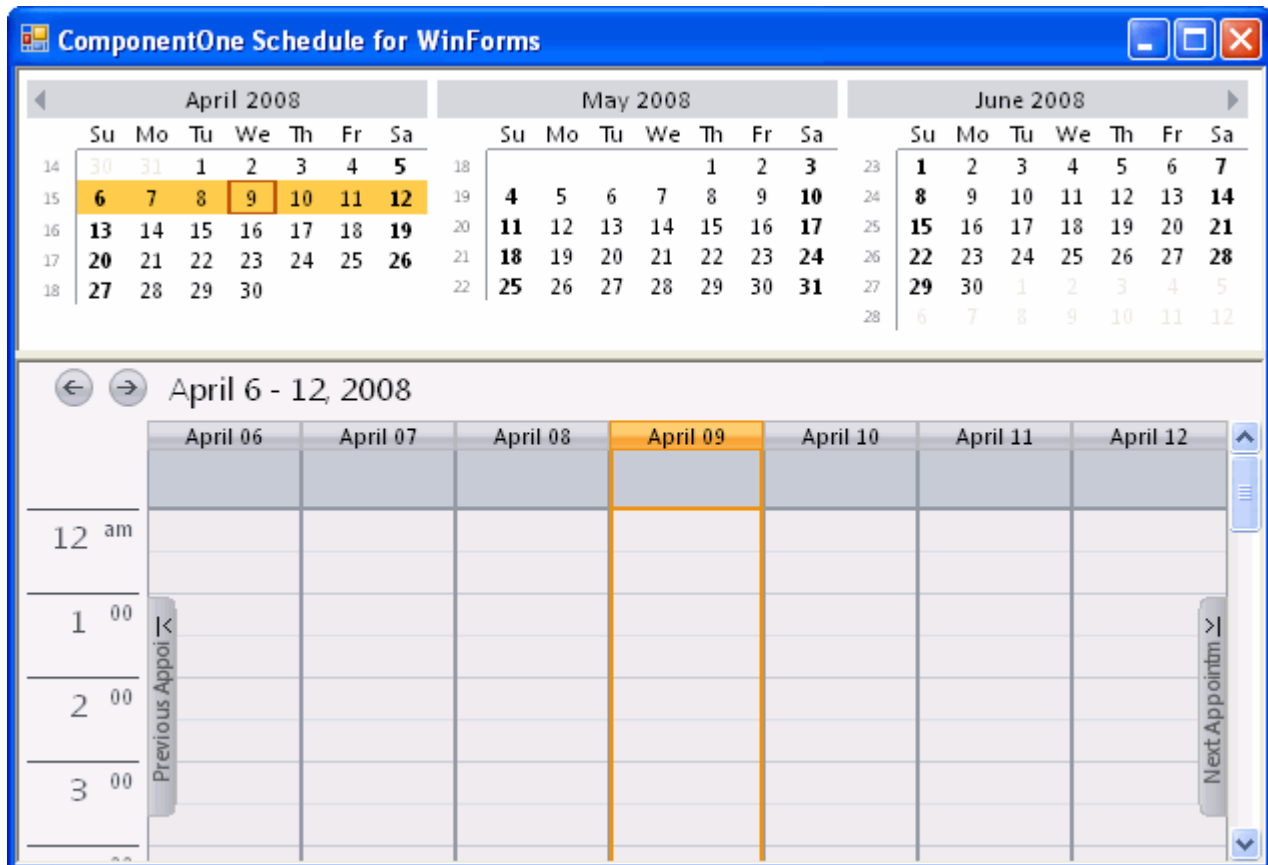
13. Open the **C1Schedule Smart Designer** located near the top left of the C1Schedule control to set some of the C1Schedule settings. For more information on accessing the **C1Schedule Smart Designer**, see [C1Schedule Smart Designer](#).
14. Click the **Calendar Settings** button and in the **Calendar Settings** dialog box change the year in the **First date** drop-down box to **2005**.
15. Click the **Visual Style** button on the **C1Schedule Smart Designer** to open the **Visual Style** dialog box and complete the following:
 - Select **Office 2007 Black** from the **Current Visual Style** drop-down list.

- Click the **Week View** button
- Close the **Visual Style** dialog box.



Run the program and observe the following:

A simple schedule application in week view with a three-month calendar using a predefined Visual Style will appear.



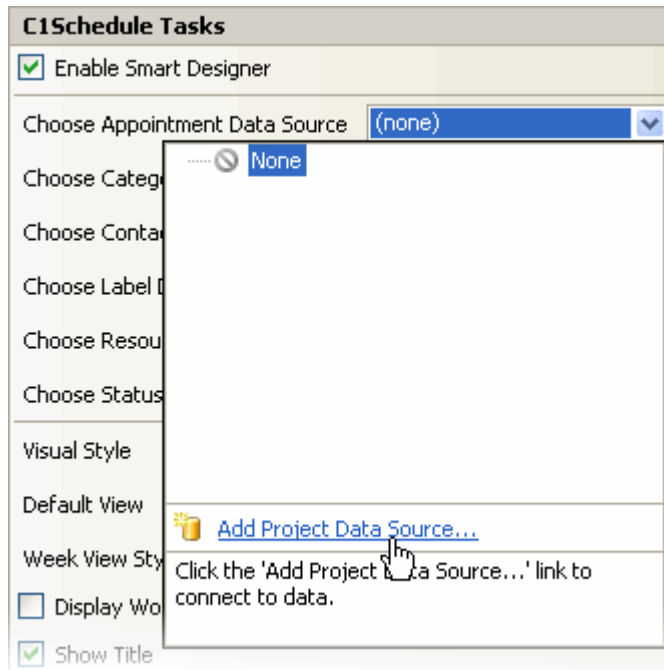
You have successfully created a simple scheduling application without writing a single line of code. In the next step, you'll learn how to bind the **C1Schedule** control to a data source.

Step 2 of 3: Binding to a Data Source

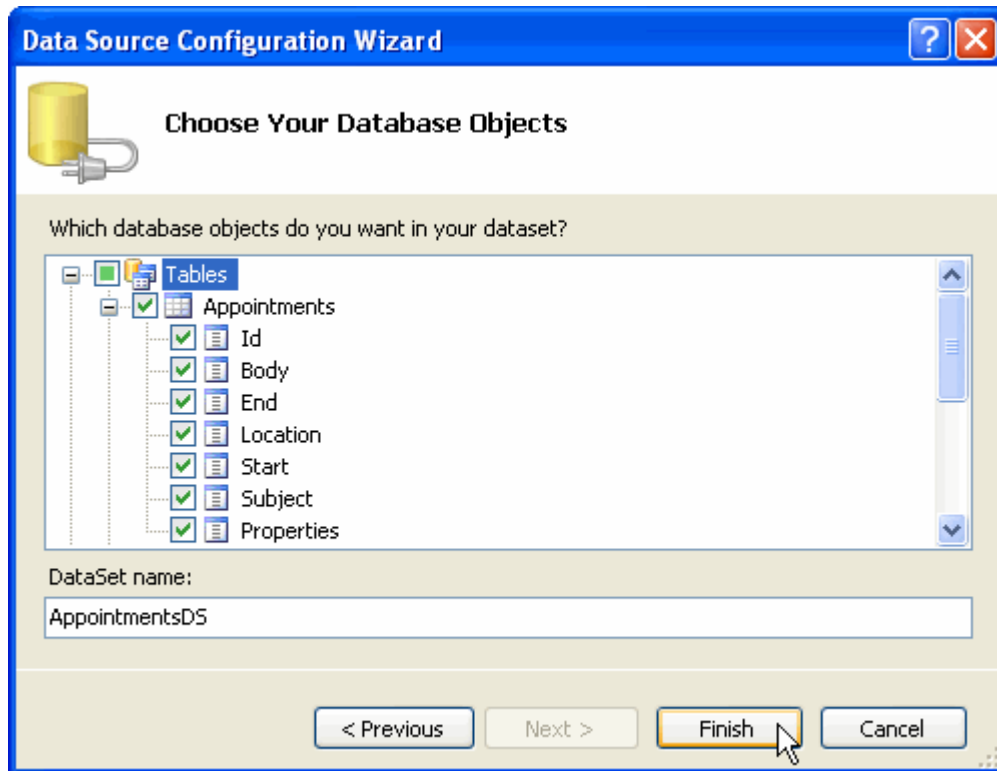
In the last step you created a simple scheduling application. In this step you'll walk through binding a data source to the scheduling application you created in the [Creating the Scheduling Application](#) topic.

To bind the **C1Schedule** control to a data source, complete the following steps:

1. Click on the **C1Schedule1**'s smart tag to open the **C1Schedule Tasks** menu. For more information on accessing the **C1Schedule Tasks** menu, see [C1Schedule Tasks Menu](#).
2. In the **C1Schedule Tasks** menu, click the **Choose Appointment Data Source** drop-down arrow and select the **Add Project Data Source** link from the drop-down box.



3. The **Data Source Configuration Wizard** appears. Leave the default setting, **Database**, selected on the **Choose a Data Source Type** page, and click **Next**.
4. Click the **New Connection** button to create a new connection or choose one from the drop-down list. The **Add Connection** dialog box appears.
5. Click **Browse**. In the **Select Microsoft Access Database File** dialog box, browse to the NWind.mdb database included with the **C1ScheduleDemo** sample. Select the **C1NWind.mdb** file and click **OK**.
6. In the **Add Connection** dialog box, click the **Test Connection** button to make sure that you have successfully connected to the database or server and click **OK** to close the dialog box that appears.
7. Click **OK** to close the **Add Connection** dialog box. The new string appears in the on the **Choose Your Data Connection** page.
8. Click the **Next** button to continue. A dialog box will appear asking if you would like to add the data file to your project and modify the connection string. Since it is not necessary to copy the database to your project, click **No**.
9. Save the connection string in the application configuration file by checking the **Yes, save the connection as** box and leave the default name, **C1NWindConnectionString**. Click the **Next** button to continue.
10. On the **Choose Your Database Objects** page, select the **Appointments** table and include all of the fields. Enter **AppointmentsDS** in the **DataSet name** box and click **Finish** to exit the wizard.



11. Switch to Code view and notice that a DataSet and connection string are added to your project. Additionally, Visual Studio automatically creates the following code to fill the DataSet:

To write code in Visual Basic

Visual Basic

```
Me.AppointmentsTableAdapter.Fill(Me.AppointmentsDS.Appointments)
```

To write code in C#

C#

```
this.appointmentsTableAdapter.Fill(this.appointmentsDS.Appointments);
```

12. Add the following code to the **Form_FormClosing** event to update the database when the form closes:

To write code in Visual Basic

Visual Basic

```
Me.AppointmentsTableAdapter.Update(Me.AppointmentsDS.Appointments)
```

To write code in C#

C#

```
this.appointmentsTableAdapter.Update(this.appointmentsDS.Appointments);
```

13. Return to Design view; in the **C1Schedule Tasks** menu, a **Set Appointment Mappings** link will appear after the data source is set. Click the **Set Appointment Mappings** link to open the **Appointment Storage Properties** dialog box.

C1Schedule Tasks

☒ Enable Smart Designer

Choose Appointment Data Source: AppointmentsBindingSour ▼

[Set Appointment Mappings...](#)


Choose Category Data Source: (none) ▼

Choose Contact Data Source: (none) ▼

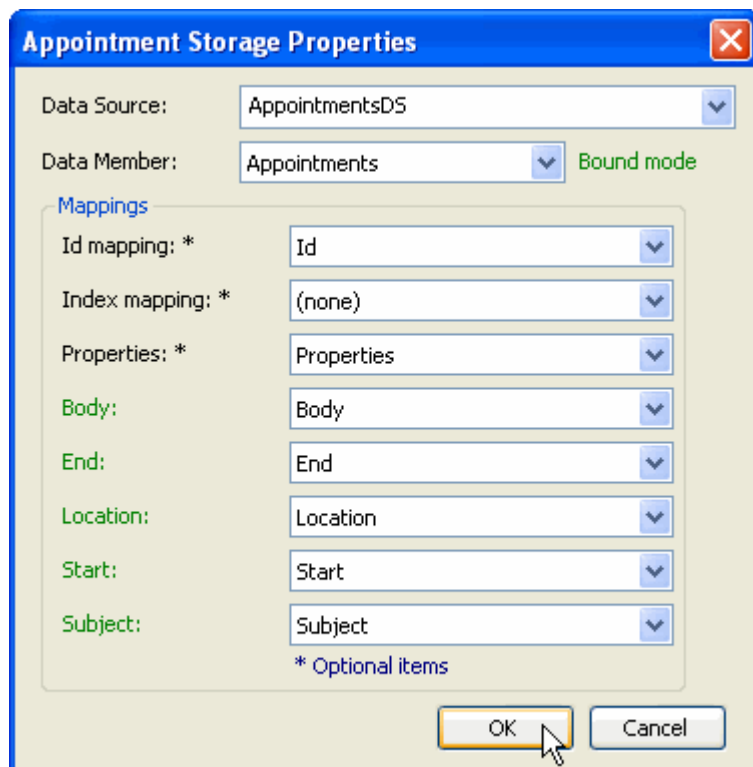
Choose Label Data Source: (none) ▼

14. In the **Appointment Storage Properties** dialog box, click the **Data Source** drop-down arrow and select **AppointmentDS** from the drop-down list.
15. Click the **Data Member** drop-down arrow and select **Appointments** from the drop-down list.
16. Under **Mappings**, set the following properties:

Mapping	Property
Id mapping	Id
Properties	Properties
Body	Body
End	End
Location	Location
Start	Start
Subject	Subject

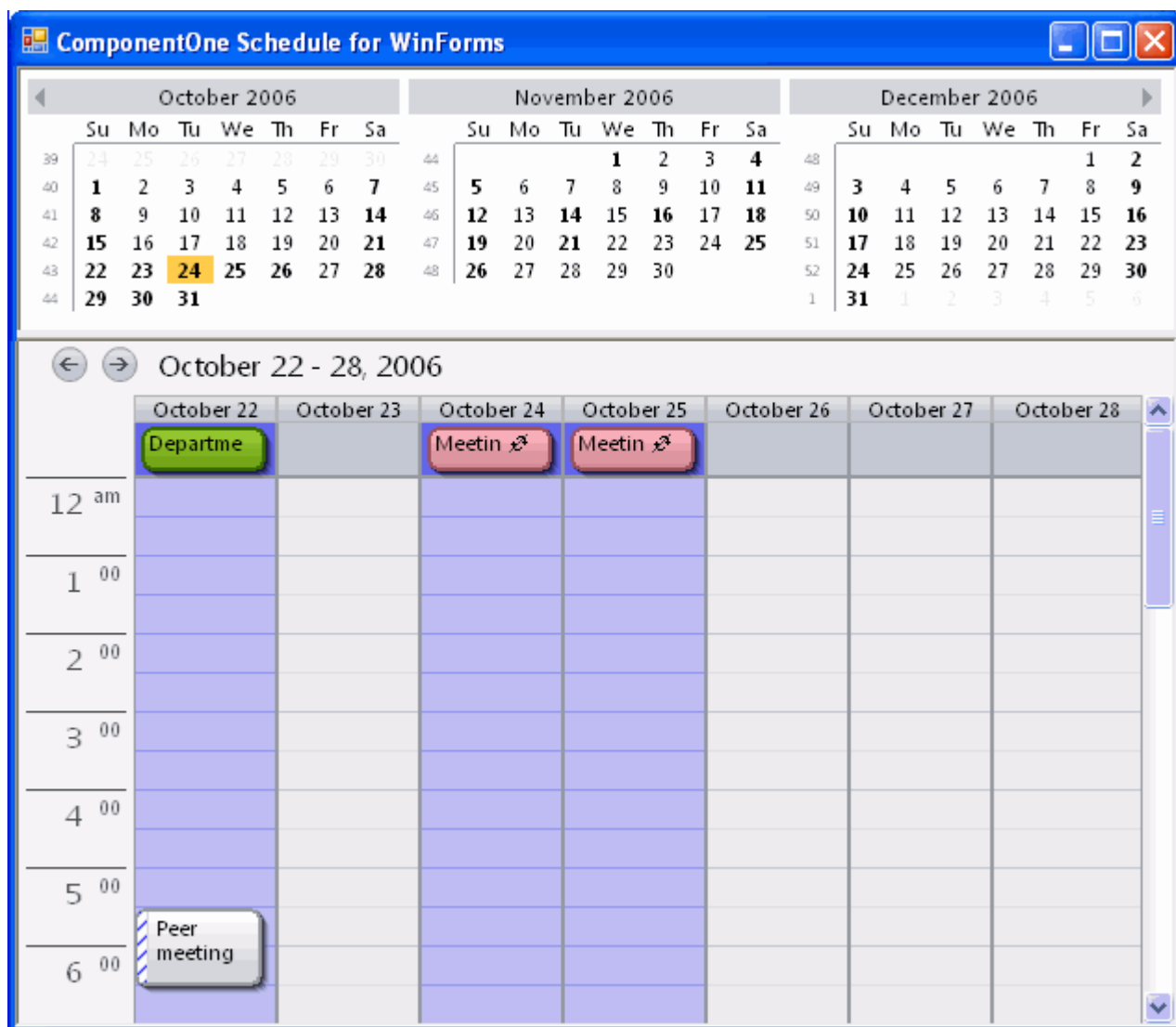
 **Note:** Required properties will change from red to green when they are properly set. To ensure that all mappings are set, there should not be any red properties; otherwise, a run-time error will occur.

17. Click **OK** to close the **Appointment Storage Properties** dialog box.

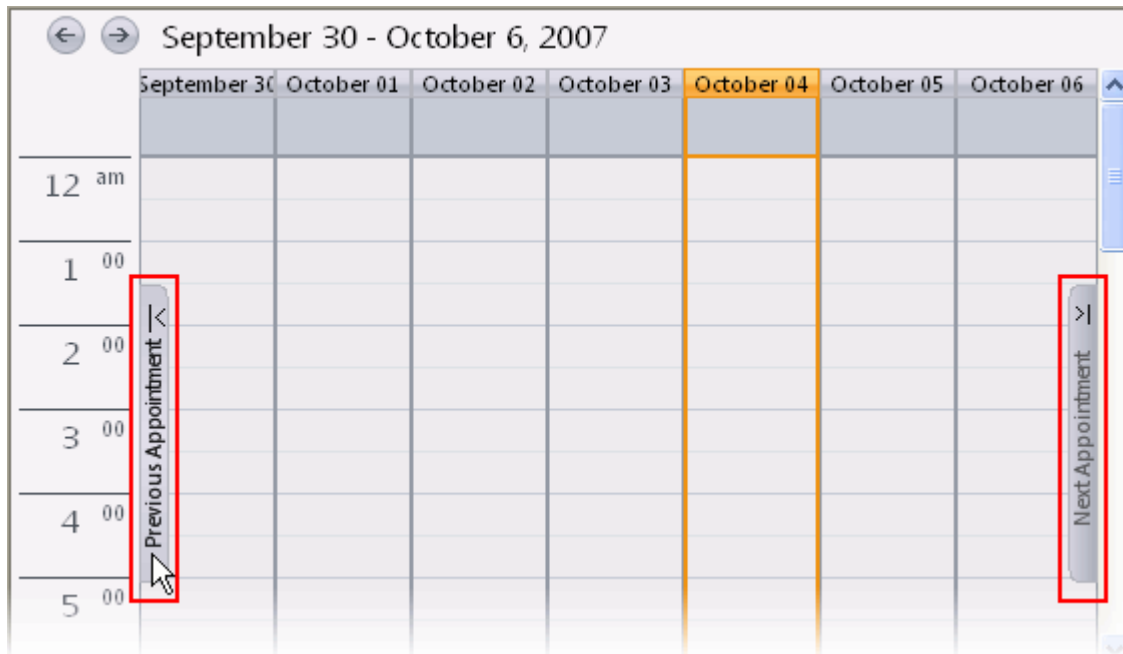


Run the application and observe the following:

Notice that the data from the **Appointments** table is reflected in both the [C1Calendar](#) and [C1Schedule](#) controls.



By default, the Office 2007 Visual Style gives you added navigation options by enabling the navigation panels and the browse buttons. If an appointment is not present in the visible range of dates, a navigation panel appears on the left and right sides of the schedule. Clicking on one of the panels changes the current view to the previous or next date with appointments. If there is not a previous or next appointment, the corresponding panel is disabled. In the image below the **Previous Appointment** panel is enabled and the **Next Appointment** panel is disabled.



You can also use the navigation buttons to navigate through the schedule. The navigation buttons appear on the Title bar of the [C1Schedule](#) control. Clicking on one of the navigation buttons changes the current view to the next or previous range of dates.

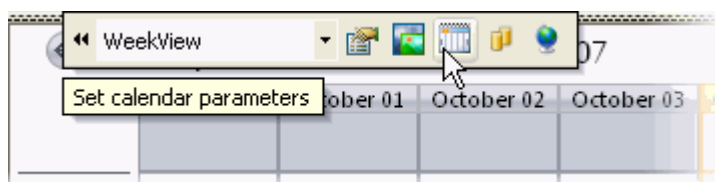


You have successfully bound the scheduling application to a data source. In the next topic, you will learn how to customize calendar settings for a 10-hour day, four-day work week.

Step 3 of 3: Customizing the Calendar Settings

The following steps will walk you through setting up the calendar for a 10-hour day, four-day work week, including setting the start day time, end day time, the time scale, and week start properties. Complete the following to customize the calendar:

1. On the **C1Schedule Smart Designer**, click the **Calendar Settings** button. For more information on accessing the **C1Schedule Smart Designer**, see [C1Schedule Smart Designer](#).



2. Set the start and end day times by setting the `StartDayTime` and `EndDayTime` properties in the Smart Designer or in code.

In the Designer

- In the **Calendar Settings** dialog box, set the Start day time box to **10:00** (AM) and set the End day time box to **21:00** (9:00 PM).

- Don't close the dialog box.

In Code

- Add the following code to the **Form_Load** event to set the **StartDayTime** property to **10:00 AM** and the **EndDayTime** property to **21:00** (9:00 PM):

To write code in Visual Basic

Visual Basic

```
Me.C1Schedule1.CalendarInfo.StartDayTime = New TimeSpan(10, 0, 0)
Me.C1Schedule1.CalendarInfo.EndDayTime = New TimeSpan(21, 0, 0)
```

To write code in C#

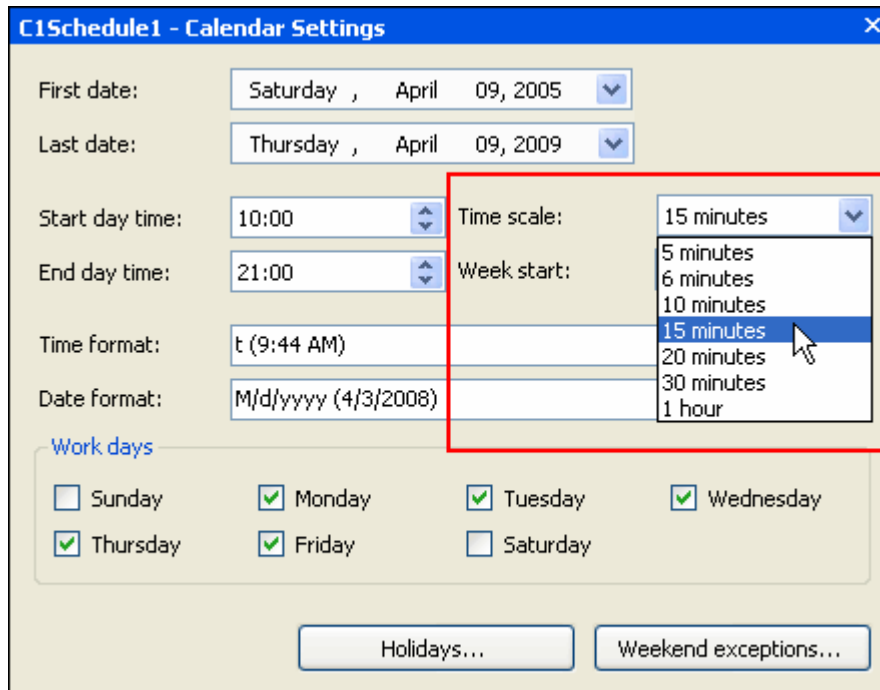
C#

```
this.c1Schedule1.CalendarInfo.StartDayTime = new TimeSpan(10, 0, 0);
this.c1Schedule1.CalendarInfo.EndDayTime = new TimeSpan(21, 0, 0);
```

3. Set the time scale by setting the **TimeScale** property in the Smart Designer or in code.

In the Designer

- In the **Calendar Settings** dialog box, set the **Time scale** drop-down to **15 minutes**.



- Don't close the dialog box.

In Code

- Add the following code to the **Form_Load** event to set the **TimeInterval** property to **15 minutes** using the **TimeScaleEnum** enumeration:

To write code in Visual Basic

Visual Basic

```
Me.C1Schedule1.CalendarInfo.TimeInterval =  
C1.C1Schedule.TimeScaleEnum.FifteenMinutes
```

To write code in C#

C#

```
this.c1Schedule1.CalendarInfo.TimeInterval =  
C1.C1Schedule.TimeScaleEnum.FifteenMinutes;
```

OR

- Add the following code to the **Form_Load** event to set the **TimeScale** property to **15 minutes** by assigning a value:

To write code in Visual Basic

Visual Basic

```
Me.C1Schedule1.CalendarInfo.TimeScale = New TimeSpan(90000000000)
```

To write code in C#

C#

```
this.c1Schedule1.CalendarInfo.TimeScale = new TimeSpan(90000000000);
```

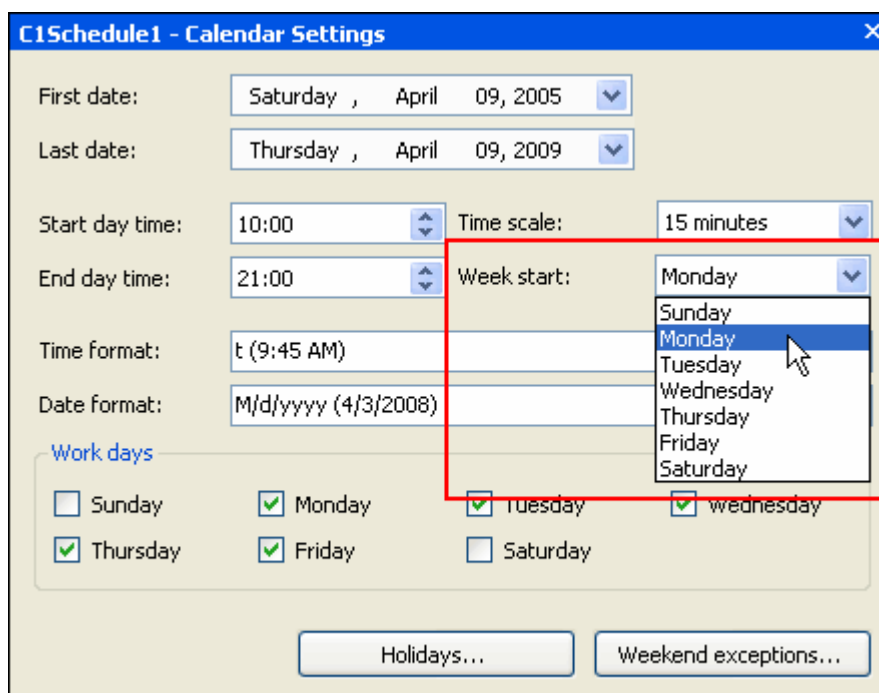


Note: The TimeScale property cannot be set to any arbitrary value. When assigning a value to this property it is automatically calculated so as to be equal to the nearest TimeScaleEnum enumeration value.

4. Set the week start day by setting the WeekStart property in the Smart Designer or in code.

In the Designer

- In the **Calendar Settings** dialog box, set the **Week start** drop-down to **Monday**.



- Don't close the dialog box.

In Code

- Add the following code to the **Form_Load** event to set the **WeekStart** property to **Monday**:

To write code in Visual Basic

Visual Basic

```
Me.C1Schedule1.CalendarInfo.WeekStart = DayOfWeek.Monday
```

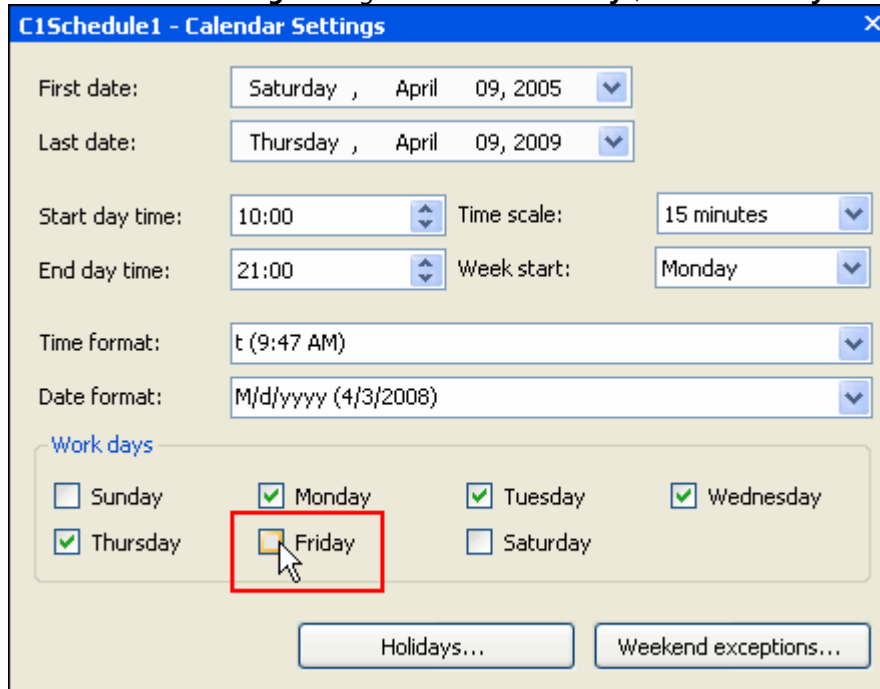
To write code in C#

C#

```
this.c1Schedule1.CalendarInfo.WeekStart = DayOfWeek.Monday;
```

5. Set the work days by setting the **WorkDays** property in the Smart Designer:

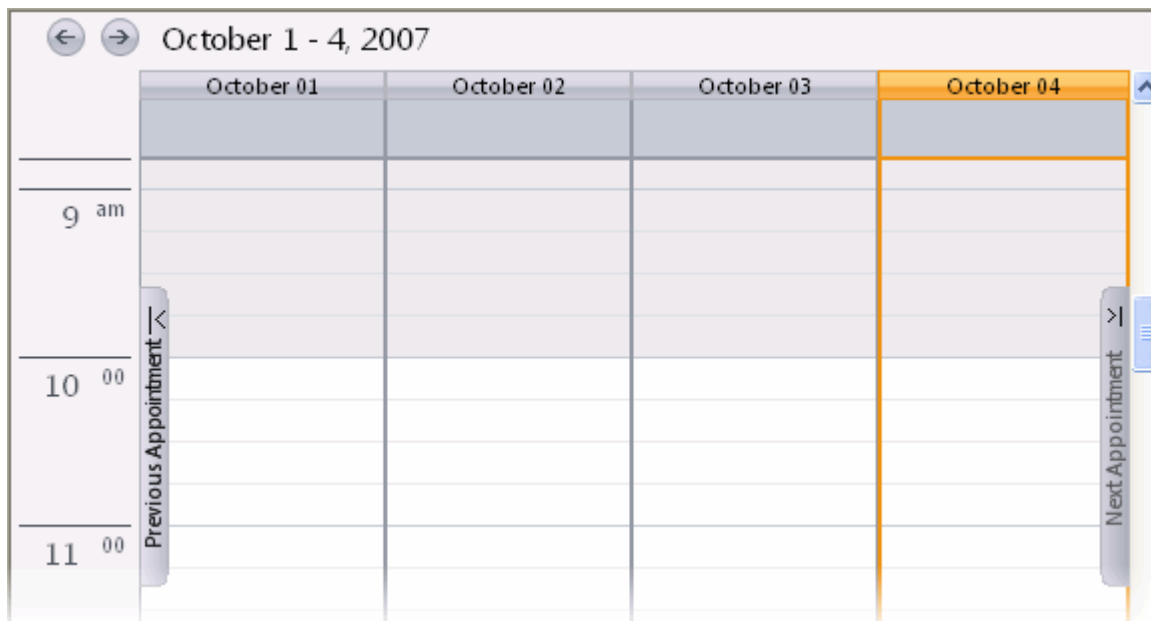
- o In the **Calendar Settings** dialog box under **Work days**, uncheck **Friday**.



6. Close the **Calendar Settings** dialog box.

Run the application and observe the following:

The scheduling application displays a 4-day work week when in the Work Week view.



Congratulations! You have successfully set the calendar settings for a 10-hour day, four-day work week. This concludes the Quick Start.

Scheduler for WinForms Top Tips

The following tips were compiled from frequently asked user questions posted in the [Scheduler for WinForms online forum](#).

Tip 1: Disable Updating C1Schedule while Filling the Underlying Data Source

When you bind the [C1Schedule](#) control to a data source, every change in the underlying data source forces the C1Schedule control to update. This could affect performance, especially when loading a large amount of data. To disable updating, you can call the [BeginUpdate](#) method before data operations and call the [EndUpdate](#) when the operations are complete. For example:

To write code in Visual Basic

Visual Basic

```
' Disable updating
Me.C1Schedule1.BeginUpdate()
' Fill datasets with data
Me.ProductsTableAdapter.Fill(Me.NwindDataSet1.Products)
Me.EmployeesTableAdapter.Fill(NwindDataSet.Employees)
Me.AppointmentsTableAdapter.Fill(NwindDataSet.Appointments)
' Enable updating
Me.C1Schedule1.EndUpdate()
```

To write code in C#

C#

```
// Disable updating
this.c1Schedule1.BeginUpdate();
// Fill datasets with data
this.productsTableAdapter.Fill(this.nwindDataSet1.Products);
this.employeesTableAdapter.Fill(nwindDataSet.Employees);
this.appointmentsTableAdapter.Fill(nwindDataSet.Appointments);
// Enable updating
this.c1Schedule1.EndUpdate();
```

Tip 2: Use the CustomData and Tag Properties to Keep Additional Information

If needed, you can store additional appointment information in the [C1Schedule](#) control. The [Appointment](#) class has two properties which can be used to keep custom information:

- The [Tag](#) property can be used at run time for storing any information. This property value is not used for appointment serialization, so, for example, it won't be saved at export or via [AppointmentStorage](#) binding.
- The [CustomData](#) property is serialized with other appointment properties. It isn't used by the C1Schedule control; you can use it to keep your data associated with appointment. This property is serialized into an XML format and it is saved into underlying data source if mapping for Appointment properties is set. It is not retained when exporting to or importing from iCal format files.

Tip 3: Extracting All Appointments for Particular Period of Time

To get all Appointment objects for particular time period, use appropriate overload of [GetOccurrences](#) method. This method returns all events in the specified period, including recurring events occurrences. For example:

To write code in Visual Basic

Visual Basic

```
Dim todayAppointments As AppointmentList =
C1Schedule1.DataStorage.AppointmentStorage.Appointments.GetOccurrences(DateTime.Today,
DateTime.Today.AddDays(1))
```

To write code in C#

C#

```
AppointmentList todayAppointments =
c1Schedule1.DataStorage.AppointmentStorage.Appointments.GetOccurrences(DateTime.Today,
DateTime.Today.AddDays(1));
```

Tip 4: Customize Appointment Appearance

You can change appointment appearance in the [BeforeAppointmentFormat](#) event's event handler. The [Text](#) property accepts HTML to customize the display in the control. Or example, you can use the following code to insert a custom image:

To write code in Visual Basic

Visual Basic

```
e.Text = "<div><img src=res://myIcon.gif/></div>"
```

To write code in C#

C#

```
e.Text = "<div><img src=res://myIcon.gif/></div>";
```

In the above code the myIcon.gif image will be extracted from your application resources and inserted into appointment. You can change other [BeforeAppointmentFormatEventArgs](#) properties as well. Note that this information won't be saved to underlying datasource, you should handle appointment information and apply formatting from your code at run time. For example:

To write code in Visual Basic

Visual Basic

```
Private Sub c1Schedule1_BeforeAppointmentFormat(sender As Object, e As
BeforeAppointmentFormatEventArgs)
    If e.Appointment.ReminderSet Then
        e.Icons = e.Icons Or AppointmentIcons.Reminder
    End If
    If Not [String].IsNullOrEmpty(e.Text) AndAlso e.Appointment.Importance =
ImportanceEnum.High Then
        e.Text = "<div style=""color:red""><b>" + e.Text + "</b></div>"
    End If
End Sub
```



```

        End If
        If [String].IsNullOrEmpty(e.Text) AndAlso Not
[String].IsNullOrEmpty(e.Appointment.Body) Then
            e.Text = "<a href=more>More info...</a>"
        Else
            e.Text = "<div><img src=res://myIcon.gif/></div>"
        End If
    End Sub
End Sub

```

To write code in C#

C#

```

private void c1Schedule1_BeforeAppointmentFormat(object sender,
BeforeAppointmentFormatEventArgs e)
{
    if (e.Appointment.ReminderSet)
    {
        e.Icons |= AppointmentIcons.Reminder;
    }
    if (!String.IsNullOrEmpty(e.Text) &&
        e.Appointment.Importance == ImportanceEnum.High)
    {
        e.Text = "<div style=\"color:red\"><b>" + e.Text + "</b></div>";
    }
    if (String.IsNullOrEmpty(e.Text) &&
        !String.IsNullOrEmpty(e.Appointment.Body))
    {
        e.Text = "<a href=more>More info...</a>";
    }
    else
    {
        e.Text = "<div><img src=res://myIcon.gif/></div>";
    }
}

```

Tip 5: Customize Appointment ToolTips

Appointment ToolTips can include HTML markup, allowing you to customize ToolTips easily. You can change a Tooltip in the [BeforeAppointmentTooltipShow](#) event's event handler. For example:

To write code in Visual Basic

Visual Basic

```

Private Sub C1Schedule1_BeforeAppointmentTooltipShow(sender As Object, e As
BeforeAppointmentTooltipShowEventArgs)
    e.Text = "<img src='res://clicon_bottom3.gif'>" + "<br/>" +
e.Appointment.Start.ToShortTimeString() + " <b>" + e.Appointment.Subject + "</b><br/>"
+ e.Appointment.Body
End Sub

```

To write code in C#

C#

```
private void c1Schedule1_BeforeAppointmentTooltipShow(object sender,
BeforeAppointmentTooltipShowEventArgs e)
{
    e.Text = "<img src='res://clicon_bottom3.gif'>" + "<br/>" +
    e.Appointment.Start.ToShortTimeString() +
    " <b>" + e.Appointment.Subject + "</b><br/>" + e.Appointment.Body;
}
```

Design-Time Support

Scheduler for WinForms provides visual editing to make it easier to create a schedule application. The following sections describe how to use **Scheduler for WinForms**' design-time environment to configure the **Scheduler for WinForms** controls:

Smart Tags and Tasks Menus

You can invoke each control's tasks menu by clicking on the smart tag (📌) in the upper-right corner of the control. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each control. For more information on how to use the tasks menu for each control in **Scheduler for WinForms**, see [C1Schedule Tasks Menu](#) and [C1Calendar Tasks Menu](#).

Smart Designers

You can easily configure the **Scheduler for WinForms** components at design time by using the associated smart designers. For more information on the **Scheduler for WinForms** smart designers, see the [C1Schedule Smart Designer](#) and [C1Calendar Smart Designer](#).

Properties Window

You can also easily configure **Scheduler for WinForms** at design time using the Properties window in Visual Studio. You can access the Properties window by right-clicking the control and selecting **Properties**.

Scheduler for WinForms Smart Tags

In Visual Studio, the [C1Schedule](#) and [C1Calendar](#) controls include a smart tag. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each control. You can invoke the tasks menu by clicking on the smart tag (📌) in the upper-right corner of each control. For more information on how to use the tasks menu for each control in **C1Schedule**, see [C1Schedule Tasks Menu](#) and [C1Calendar Tasks Menu](#).

C1Schedule Tasks Menu

In the **C1Schedule Tasks** menu you can quickly and easily enable the Smart Designer, connect to and choose a data source, set properties, and dock the control.

To access the **C1Schedule Tasks** menu, click on the smart tag (📌) in the upper right corner of the control. This will open the **C1Schedule Tasks** menu.

C1Schedule Tasks

☒ Enable Smart Designer

Choose Appointment Data Source (none)

Choose Category Data Source (none)

Choose Contact Data Source (none)

Choose Owner Data Source (none)

Choose Label Data Source (none)

Choose Resource Data Source (none)

Choose Status Data Source (none)

Visual Style Office 2007 Blue

Default View DayView

Week View Style Office2007

Time Line Style Hours

☐ Display Work Time Only

☒ Display All-Day Area

☒ Show Title

Time Interval 30 minutes

Week Start Sunday

Edit Options All

☒ Show Reminder Form

☒ Show Context Menu

☒ Show Appointment ToolTip

[Reset Appearance](#)

[About C1Schedule](#)

[Localize...](#)

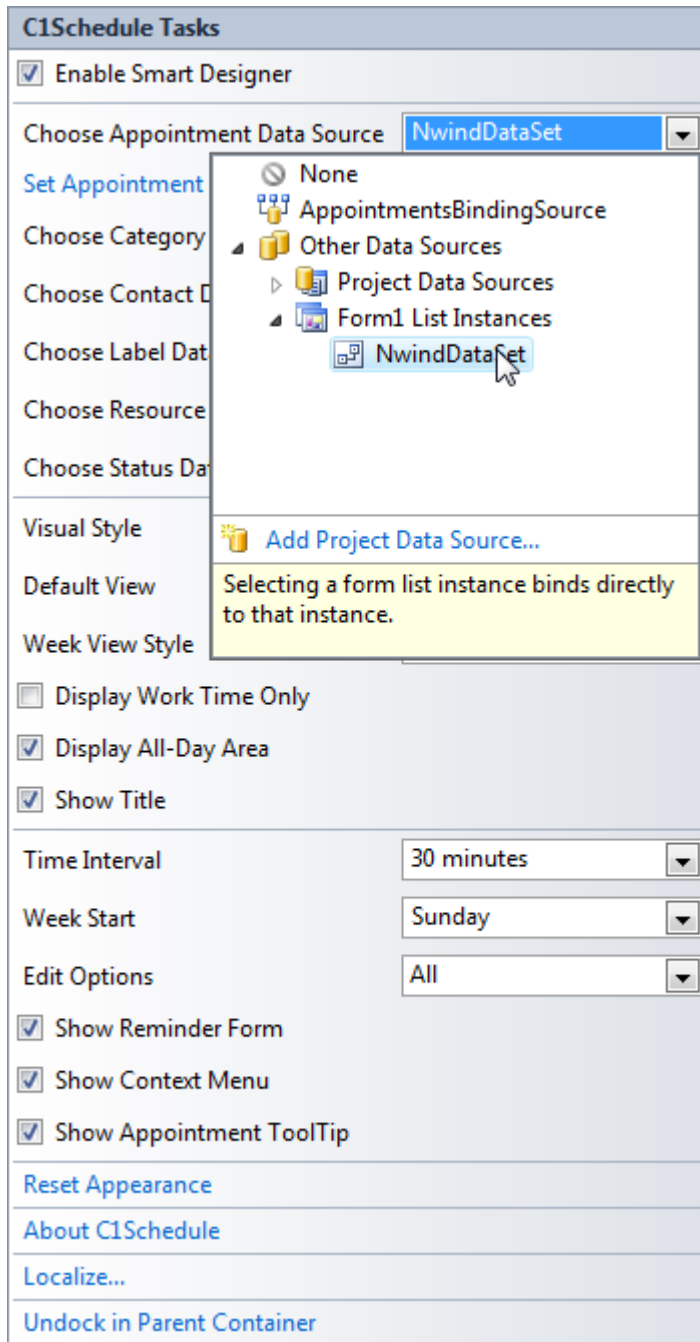
[Dock in Parent Container](#)

Enable Smart Designer

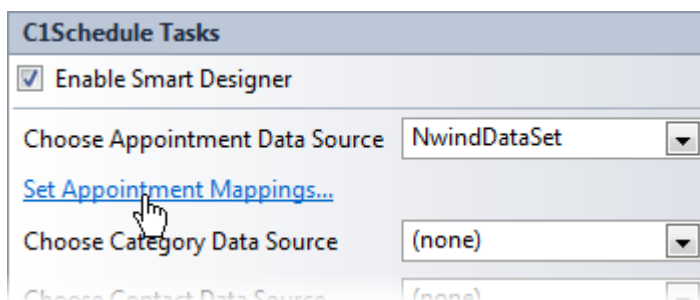
Selecting the **Enable Smart Designer** check box activates the Smart Designer on the [C1Schedule](#) control for greater design-time interaction. The default is checked. For more information on the Smart Designer, see [C1Schedule Smart Designer](#).

Choose Appointment Data Source

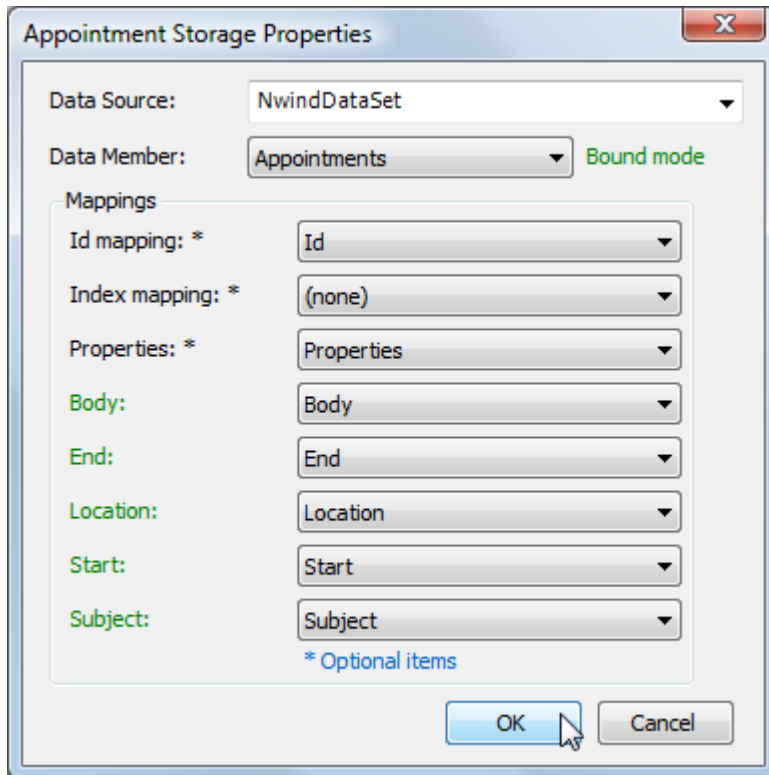
Clicking the drop-down arrow in the **Choose Appointment Data Source** box opens a list of available data sources for the [AppointmentStorage](#) and allows you to add a new data source. To add a new data source to the project, click **Add Project Data Source** to open the **Data Source Configuration Wizard**. For more information on adding a new data source to the project, see [Binding to a Data Source](#).



After binding to a data source, the option for data mappings will appear.



Clicking the **Set Appointment Mappings** link will open the **Appointment Storage Properties** dialog box where you can set the data mappings for the appointment data source.



Choose Category Data Source

Clicking the drop-down arrow in the **Choose Category Data Source** box opens a list of available data sources for the [CategoryStorage](#) and allows you to add a new data source. After binding to a data source, the option for data mappings will appear.

Choose Contact Data Source

Clicking the drop-down arrow in the **Choose Contact Data Source** box opens a list of available data sources for the [ContactStorage](#) and allows you to add a new data source. After binding to a data source, the option for data mappings will appear.

Choose Owner Data Source

Clicking the drop-down arrow in the **Choose Owner Data Source** box opens a list of available data sources for the **OwnerData** and allows you to add a new data source. After binding to a data source, the option for data mappings will appear.

Choose Label Data Source

Clicking the drop-down arrow in the **Choose Label Data Source** box opens a list of available data sources for the [LabelStorage](#) and allows you to add a new data source. After binding to a data source, the option for data mappings will appear.

Choose Resource Data Source

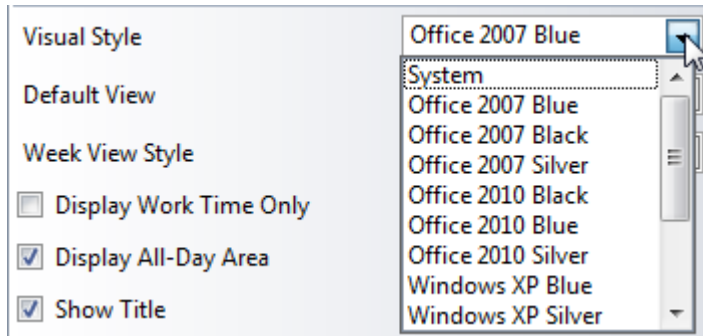
Clicking the drop-down arrow in the **Choose Resource Data Source** box opens a list of available data sources for the [ResourceStorage](#) and allows you to add a new data source. After binding to a data source, the option for data mappings will appear.

Choose Status Data Source

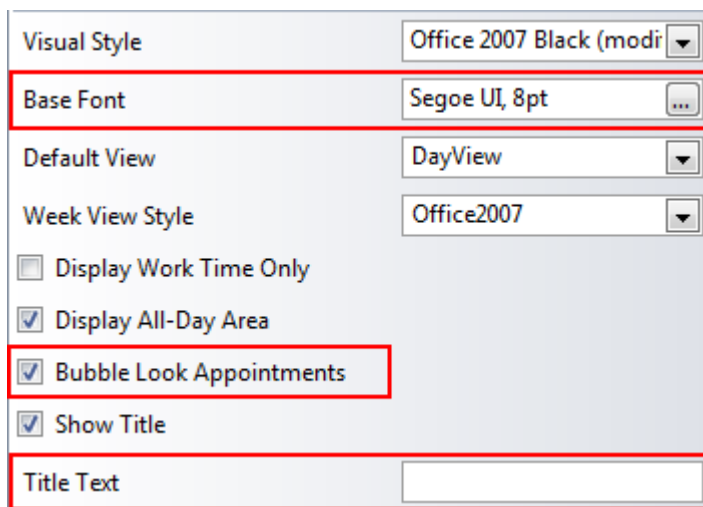
Clicking the drop-down arrow in the **Choose Status Data Source** box opens a list of available data sources for the [StatusStorage](#) and allows you to add a new data source. After binding to a data source, the option for data mappings will appear.

Visual Style

Clicking the drop-down arrow in the **Visual Style** drop-down opens a list of different [VisualStyle](#) enumeration options, such as System, Office 2007 Blue, Office 2007 Black, Office 2007 Silver, Office 2010 Black, Office 2010 Silver, Office 2010 Silver, Windows XP Blue, Windows XP Silver, Windows XP Olive, Royale, Yahoo, Aero, and Custom. The default value is **Office 2007 Blue**. For more information on the different visual styles, see [Visual Styles](#).



When the **Visual Style** drop-down is set to a modified version of one of the predefined visual styles, additional options appear in the tasks menu, such as **Base Font**, **Bubble Look Appointments**, and **Title Text**.



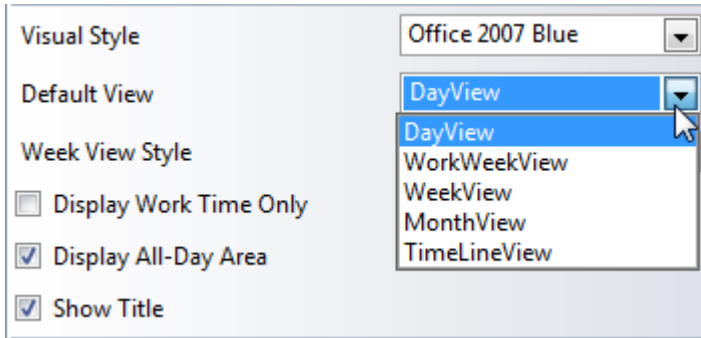
The **Base Font** box will open a **Font** dialog box where you can change the appearance of the font for the entire [C1Schedule](#) control.

The **Bubble Look Appointments** check box sets the BubbleLook property to show bubble appointments with a gradient background. By default, this property is set to **True** for Office 2007-style Visual Styles and **False** for other Visual Styles.

The **Title Text** text box sets the TitleText property, which displays a string in the title of the [C1Schedule](#) control.

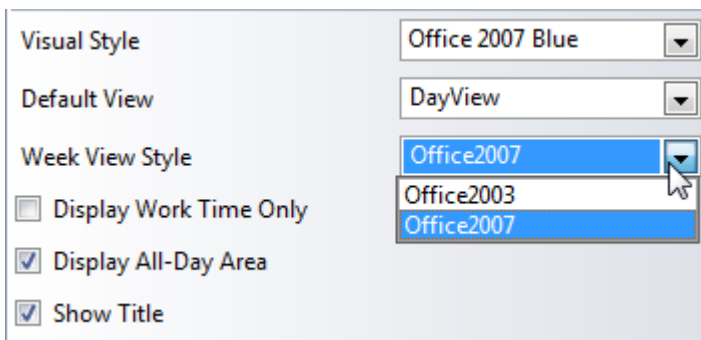
Default View

Clicking the drop-down arrow in the **Default View** box opens a list of different [ScheduleViewEnum](#) enumeration options, such as DayView, WeekView, WorkWeekView, and MonthView. The default value is **DayView**. For more information on the different data views, see [Data Views](#).



Week View Style

Clicking the drop-down arrow in the Week View Style box opens a list of different [WeekViewStyleEnum](#) enumeration options, such as Office2003 and Office2007. The default value is **Office2007**.



Time Line Style

Clicking the drop-down arrow in the **Time Line Style** box opens a list of different [TimeLineStyleEnum](#) enumeration options, such as Hours and Days. The default value is **Hours**.

Display Work Time Only

Checking the **Display Work Time Only** check box displays only the time set as work time when in [DayView](#), [WeekView](#), and [WorkWeekView](#) views. The default value is unchecked.

Display All-Day Area

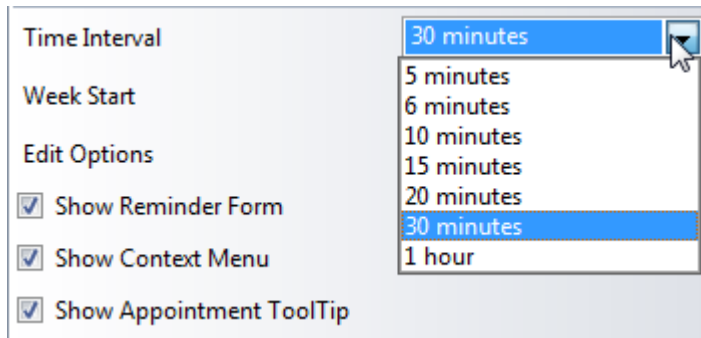
Checking the **Display All-Day Area** check box displays the all-day area in the appointment. The default value is checked.

Show Title

Checking the **Show Title** check box sets the [ShowTitle](#) property to **True** and displays the title panel. The default value is checked.

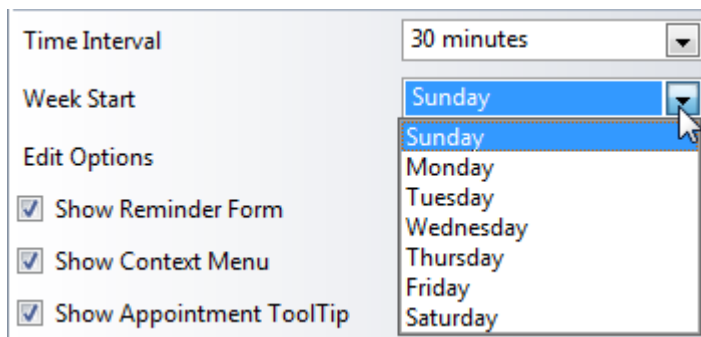
Time Interval

Clicking the drop-down arrow in the **Time Interval** box opens a list of different [TimeScaleEnum](#) enumeration options, such as [FiveMinutes](#), [SixMinutes](#), [TenMinutes](#), [FifteenMinutes](#), [TwentyMinutes](#), [ThirtyMinutes](#), and [OneHour](#). The default value is **ThirtyMinutes**. The TimeScaleEnum enumeration determines the time interval used for displaying time slots in the DayView, WeekView, and WorkWeekView views.



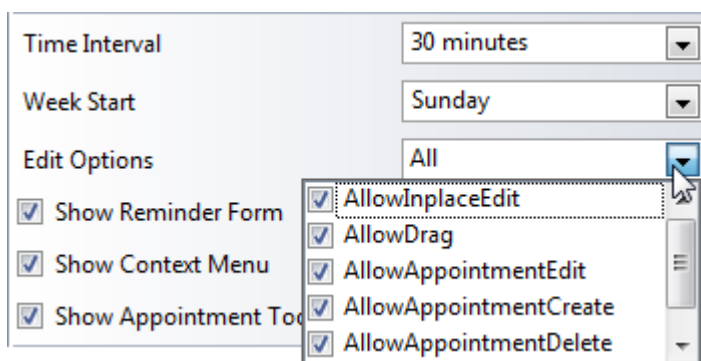
Week start

Clicking the drop-down arrow in the **Week start** box opens a list of different [WeekStart](#) property options, such as **Sunday**, **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, and **Saturday**. The default value uses the system settings. The WeekStart property determines the first day of the week.



Edit Options

Clicking the drop-down arrow next to the **Edit Options** box opens a list of different [EditOptions](#) enumeration options, such as [AllowInplaceEdit](#), [AllowDrag](#), [AllowAppointmentEdit](#), [AllowAppointmentCreate](#), [AllowAppointmentDelete](#), and [All](#). The default value is **All**.



Show Reminder Form

Checking the **Show Reminder Form** check box sets the [ShowReminderForm](#) property to **True** and displays the **Reminder** dialog box when a reminder alerts. The default value is checked.

Show Context Menu

Checking the **Show Context Menu** check box sets the [ShowContextMenu](#) property to **True** and displays the context menu when the mouse is right-clicked. The default value is checked.

Show Appointment ToolTip

Checking the **Show Appointment Tooltip** check box sets the [ShowAppointmentToolTip](#) property to **True** and displays a ToolTip when the mouse is over an appointment. The default value is checked.

Reset Appearance

Clicking **Reset Appearance** resets all of the settings to the default values.

About C1Schedule

Clicking **About** displays the [C1Schedule](#) control's **About** dialog box, which is helpful in finding the build number of the control.

Localize

Clicking the **Localize** button opens the **Localize** dialog box. In the **Localize** dialog box, you can customize your localization settings. For more information, see [Localization](#).

Dock in parent container/Undock in parent container

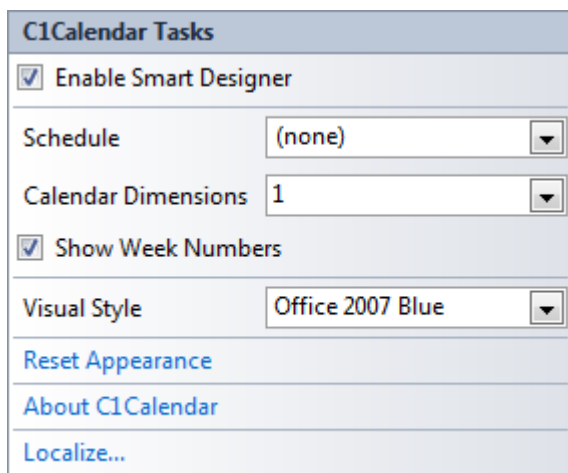
Clicking **Dock in parent container** sets the **Dock** property for [C1Schedule](#) to **Fill**.

If [C1Schedule](#) is docked in the parent container, the option to undock [C1Schedule](#) from the parent container will be available. Clicking **Undock in parent container** sets the **Dock** property for [C1Schedule](#) to **None**.

C1Calendar Tasks Menu

In the **C1Calendar Tasks** menu, you can quickly and easily enable the Smart Designer, link the [C1Calendar](#) to a [C1Schedule](#) control, and set calendar properties.

To access the **C1Calendar Tasks** menu, click on the smart tag (🔗) in the upper right corner of the control. This will open the **C1Calendar Tasks** menu.



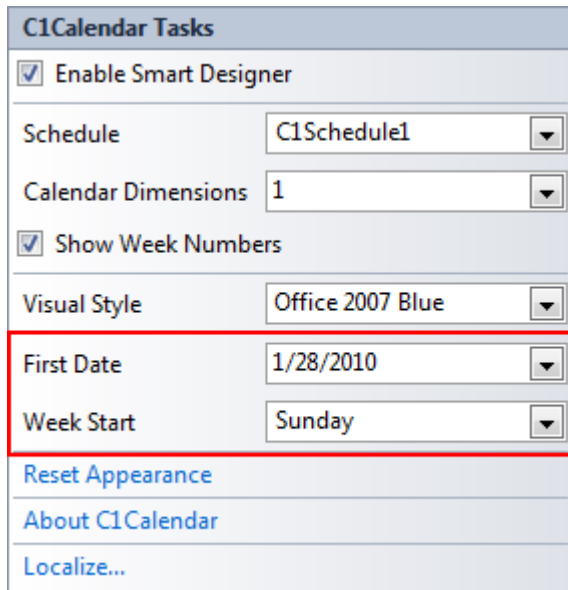
Enable Smart Designer

Selecting the **Enable Smart Designer** check box, activates the Smart Designer on the [C1Calendar](#) control for greater design-time interaction. The default is checked. For more information on the Smart Designer, see [C1Calendar Smart Designer](#).

Schedule

Clicking the drop-down arrow in the **Schedule** box allows you to link the [C1Calendar](#) control to a [C1Schedule](#) control.

When the **Schedule** drop-down box is set to a C1Schedule control, the **First Date** and **Week Start** options appear on the **C1Calendar Tasks** menu.



Calendar Dimensions

Clicking the drop-down arrow in the **Calendar Dimensions** box opens a list containing the number of months to appear in the C1Calendar control. The default value for the [CalendarDimensions](#) property is **1**.

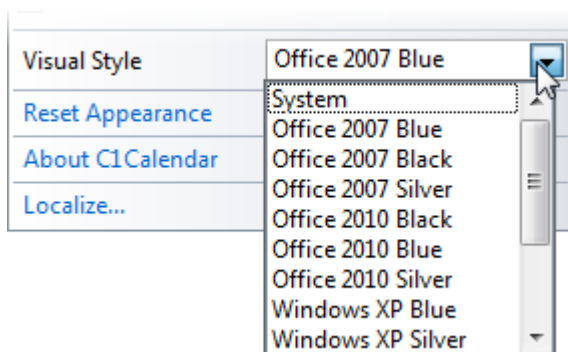
Note: The CalendarDimensions property accepts values from 1 to 12, but will only display the number of months that it can fit into the available space.

Show Week Numbers

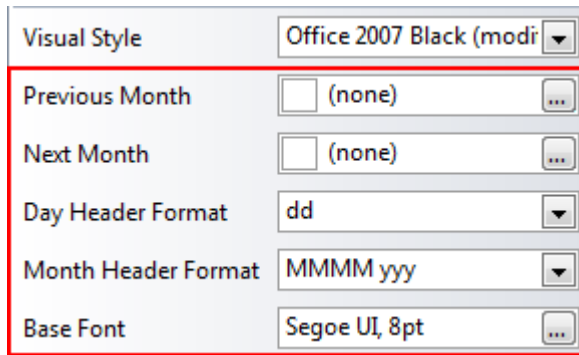
Checking the **Show Week Numbers** check box displays week number in the [C1Calendar](#) control. The default value is checked.

Visual Style

Clicking the drop-down arrow in the **Visual Style** drop-down opens a list of different [VisualStyle](#) enumeration options, such as System, Office 2007 Blue, Office 2007 Black, Office 2007 Silver, Windows XP Blue, Windows XP Silver, Windows XP Olive, Royale, Yahoo, Aero, and Custom. The default value is **Office 2007 Blue**. For more information on the different visual styles, see [Visual Styles](#).



When the **Visual Style** drop-down is set to a modified version of one of the predefined visual styles, additional options appear in the tasks menu, such as **Previous Month**, **Next Month**, **Day Header Format**, **Month Header Format**, and **Base Font**.



The **Previous Month** box will open the **Select Resource** dialog box where you can choose an image for the **Previous** button.

The **Next Month** box will open the **Select Resource** dialog where you can choose an image for the **Next** button.

The **Day Header Format** drop-down sets the format of the day header text. The default is **dd** (for example, **Mo**). Available formats include:

- **d** - Displays the 1-letter abbreviated name for each day. For example, **M**.
- **dd** - Displays the 2-letter abbreviated name for each day. For example, **Mo**.
- **ddd** - Displays the 3-letter abbreviated name for each day. For example, **Mon**.

The **Month Header Format** drop-down sets the format of the month header text. The default is **MMMM yyyy** (for example, February 2007). Available formats include:

- **MMMM yyyy** - Displays the full name of the month and the year. For example, **February 2007**.
- **MMM yyyy** - Displays the abbreviated name of the month and the year. For example, **Feb 2007**.

The **Base Font** box will open a **Font** dialog box where you can change the appearance of the font for the entire [C1Calendar](#) control.

First Date (Only available when [C1Calendar](#) is linked to a [C1Schedule](#) control)

Clicking the drop-down arrow in the **First Date** box opens a [DateTimePicker](#) to select the first date that will appear in the [C1Schedule](#) control.

Week Start (Only available when [C1Calendar](#) is linked to a [C1Schedule](#) control)

Clicking the drop-down arrow in the **Week Start** box opens a list of different [WeekStart](#) property options, such as **Sunday**, **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, and **Saturday**. The default value is determined by the system settings. The [WeekStart](#) property determines the first day of the week.

Reset Appearance

Clicking **Reset Appearance** resets all of the settings to the default values.


About C1Schedule

Clicking **About** displays the [C1Calendar](#) control's **About** dialog box, which is helpful in finding the build number of the control.

Localize


Clicking the **Localize** button opens the **Localize** dialog box. In the **Localize** dialog box, you can customize your localization settings. For more information, see [Localization](#).

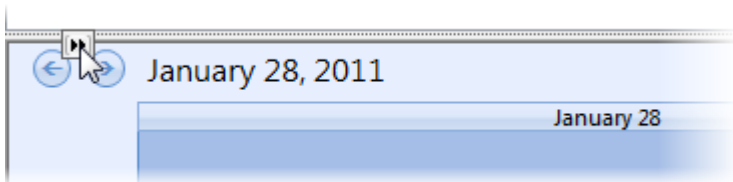
Scheduler for WinForms Smart Designers

The [C1Schedule](#) and [C1Calendar](#) Smart Designers allow you to quickly and easily configure the **Scheduler for WinForms** components at design time. The Smart Designer floating toolbars can be accessed by clicking the  button in the upper left of each control. For more information on the **Scheduler for WinForms** smart designers, see the [C1Schedule Smart Designer](#) and [C1Calendar Smart Designer](#).

C1Schedule Smart Designer

[C1Schedule](#) features a Smart Designer floating toolbar to enhance design-time interaction. Using the Smart Designer you can set properties directly on the form.

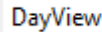





The Smart Designer's floating toolbar can be displayed by clicking the  button that appears in the upper left corner of the form.



To close the floating toolbar, click the  button.

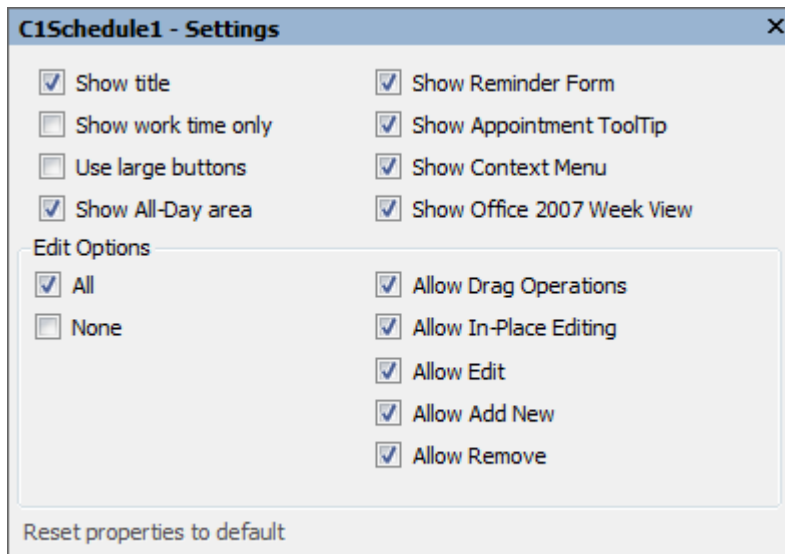


The floating toolbar for the [C1Schedule](#) control consists of the following toolbar buttons:

Button	Description
	Default View: Select the default view.
	Settings: Set the properties for the C1Schedule control.
	Visual Style: Edit the appearance of the C1Schedule control.
	Calendar Settings: Customize the calendar settings.
	Data Source Settings: Set the mappings for the data source.
	Localize: Localize the schedule.

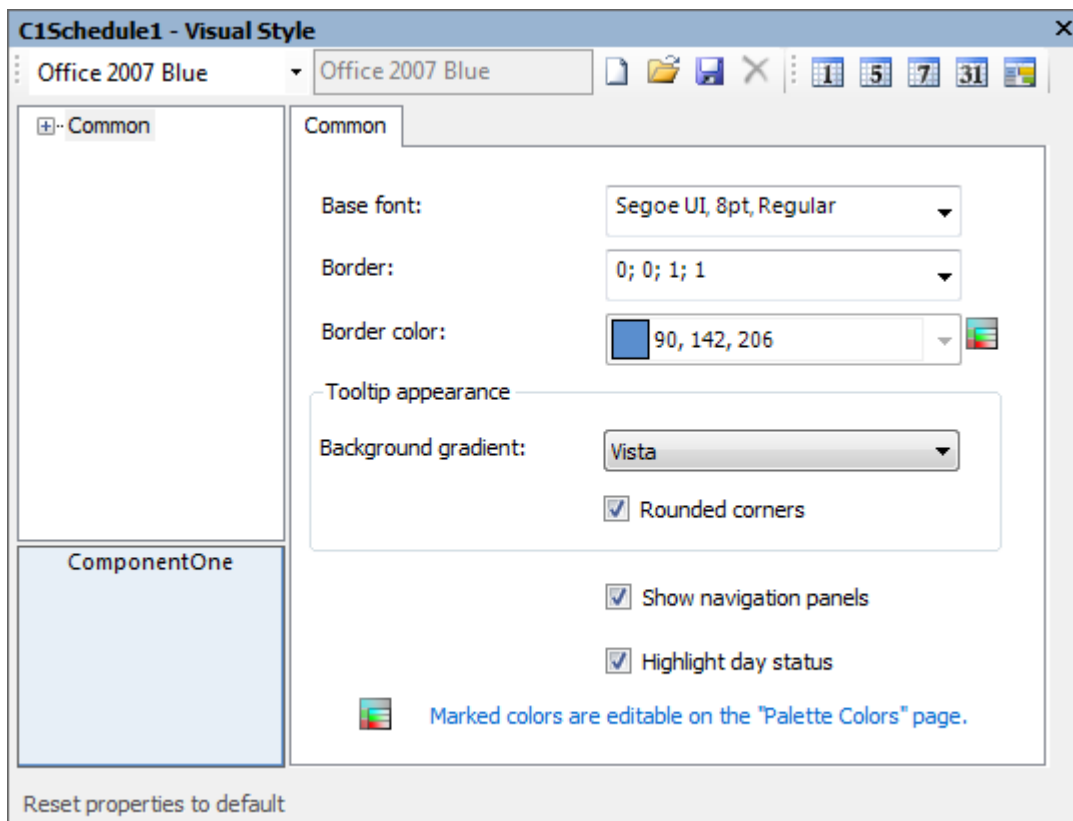
Settings

Clicking the **Settings** button opens the **Settings** dialog box where you can set properties for the C1Schedule control.



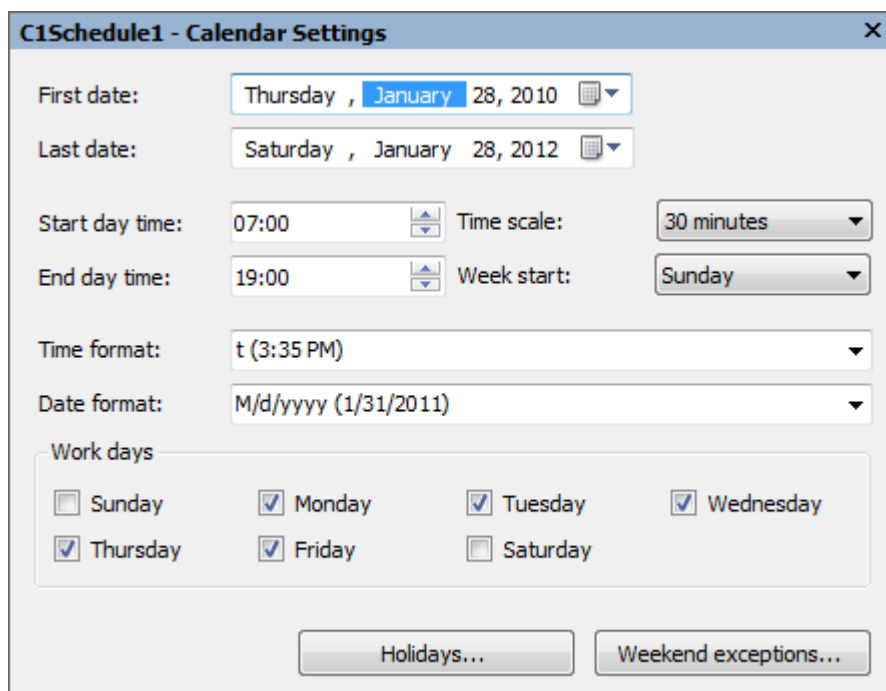
Visual Style

Clicking the **Visual Style** button opens the **Visual Style** dialog box where you can set layout properties and style options. For more information, see [Customizing Scheduler for WinForms' Appearance](#).



Calendar Settings

Clicking the **Calendar Settings** button opens the **Calendar Settings** dialog box where you can customize the settings for the calendar.



C1Schedule1 - Calendar Settings

First date: Thursday , January 28, 2010

Last date: Saturday , January 28, 2012

Start day time: 07:00 Time scale: 30 minutes

End day time: 19:00 Week start: Sunday

Time format: t (3:35 PM)

Date format: M/d/yyyy (1/31/2011)

Work days

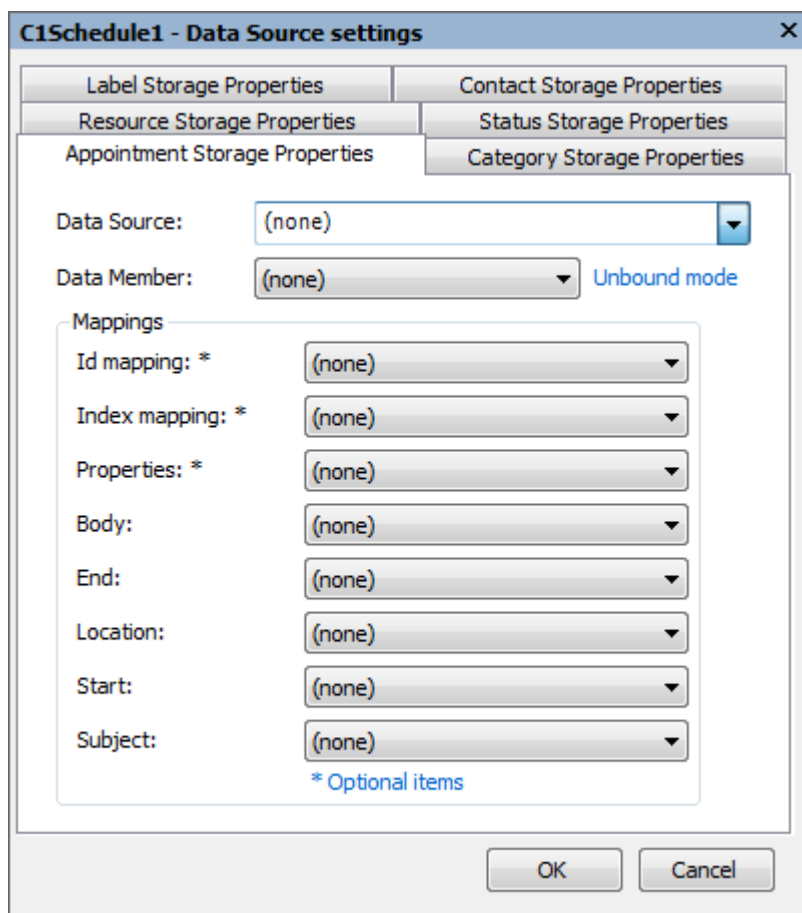
☐ Sunday ☒ Monday ☒ Tuesday ☒ Wednesday

☒ Thursday ☒ Friday ☐ Saturday

Holidays... Weekend exceptions...

Data Source Settings

Clicking the **Data Source Settings** button opens the **Data Source Settings** dialog box where you can set the mappings for the data source. By default, the **Data Member** property is in **Unbound mode**.



C1Schedule1 - Data Source settings

Label Storage Properties Contact Storage Properties

Resource Storage Properties Status Storage Properties

Appointment Storage Properties Category Storage Properties

Data Source: (none)

Data Member: (none) Unbound mode

Mappings

Id mapping: * (none)

Index mapping: * (none)

Properties: * (none)

Body: (none)

End: (none)

Location: (none)

Start: (none)

Subject: (none)

* Optional items

OK Cancel

Setting the data source to a bound data source changes the **Data Member** property to **Bound mode**. Mappings can be set similar to the image below. For more information on binding to a data source, see [Binding to a Data Source](#).

C1Schedule1 - Data Source settings

Label Storage Properties Contact Storage Properties

Resource Storage Properties Status Storage Properties

Appointment Storage Properties Category Storage Properties

Data Source: NwindDataSet

Data Member: Appointments **Bound mode**

Mappings

Id mapping: * Id

Index mapping: * (none)

Properties: * Properties

Body: Body

End: End

Location: Location

Start: Start

Subject: Subject

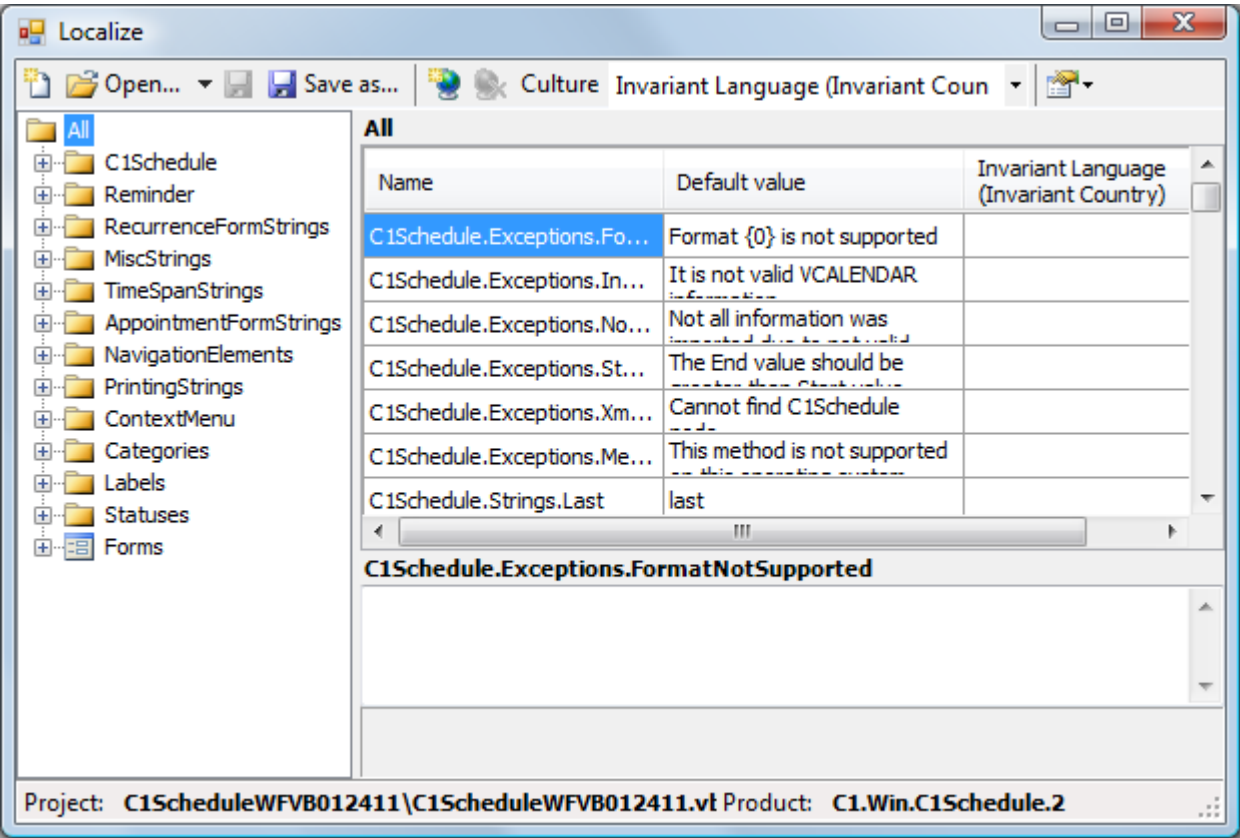
* Optional items

OK Cancel

For more information about mapping bound data to a [C1Schedule](#) control, see [Data Mappings](#).


Localize

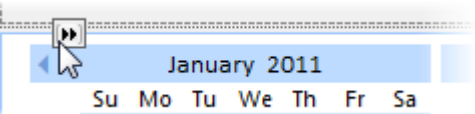
Clicking the **Localize** button opens the **Localize** dialog box where you customize your localization settings. For more information, see [Localization](#).



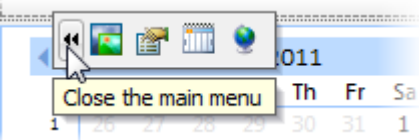
C1Calendar Smart Designer

C1Calendar features a Smart Designer floating toolbar to enhance design-time interaction. Using the Smart Designer, you can set properties directly on the form.



The Smart Designer's floating toolbar can be displayed by clicking the  button that appears in the upper left corner of the form.





To close the floating toolbar, click the  button.



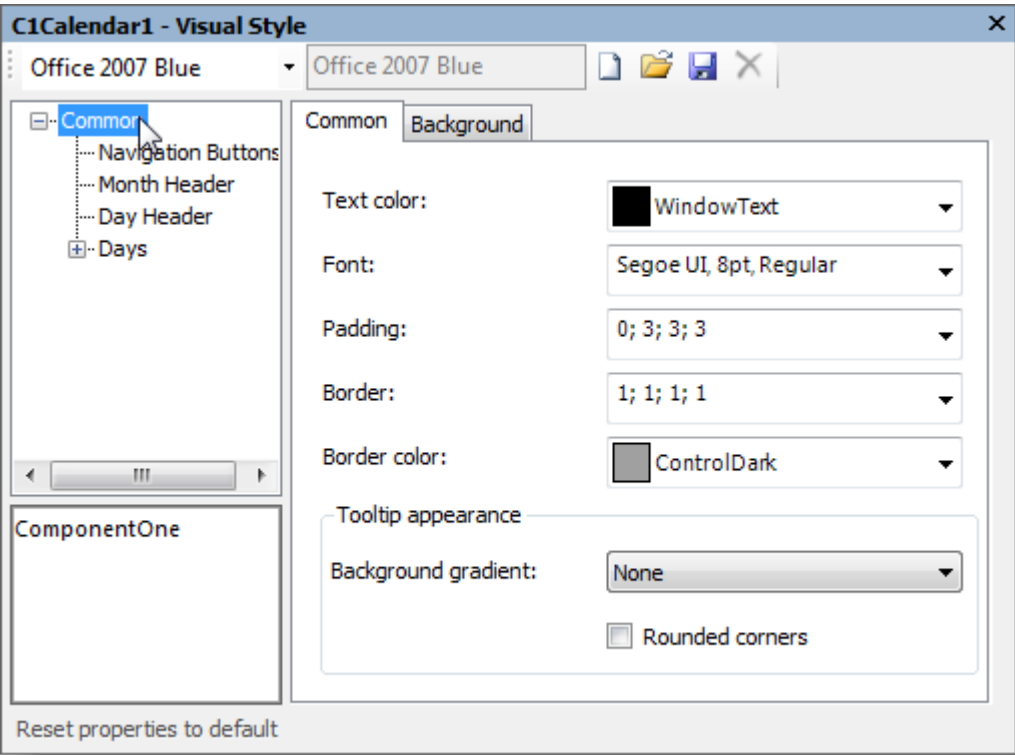
The floating toolbar for the C1Calendar control consists of the following toolbar buttons:

Button	Description
	Visual Style: Edit the appearance of the C1Calendar control.
	Settings: Set C1Calendar control properties.

Button	Description
	Calendar Settings: Customize the calendar settings.
	Localize: Localize the calendar.

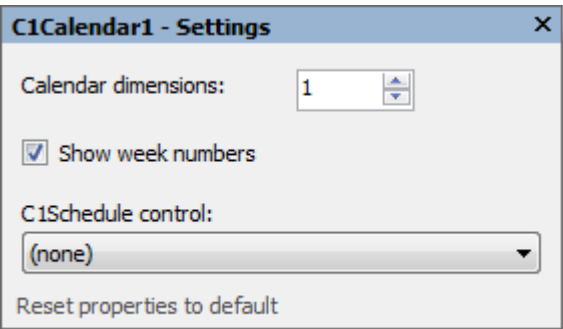
Visual Style

Clicking the **Visual Style** button opens the **Visual Style** dialog box where you can set layout properties and style options. For more information, see [Customizing Scheduler for WinForms' Appearance](#).



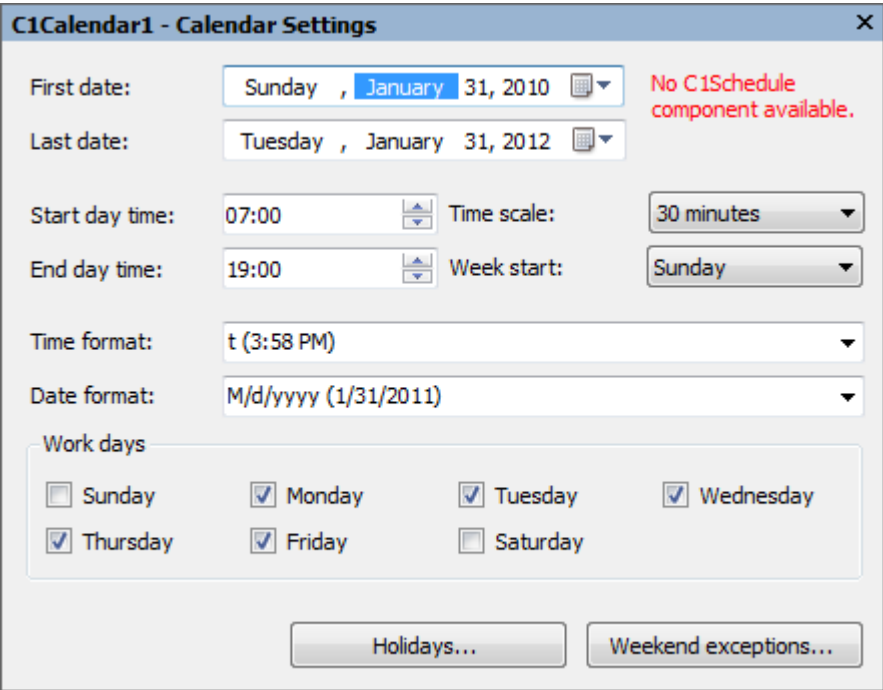
Settings

Clicking the **Settings** button opens the **Settings** dialog box where you can set calendar dimensions and link to a [C1Schedule](#) control.



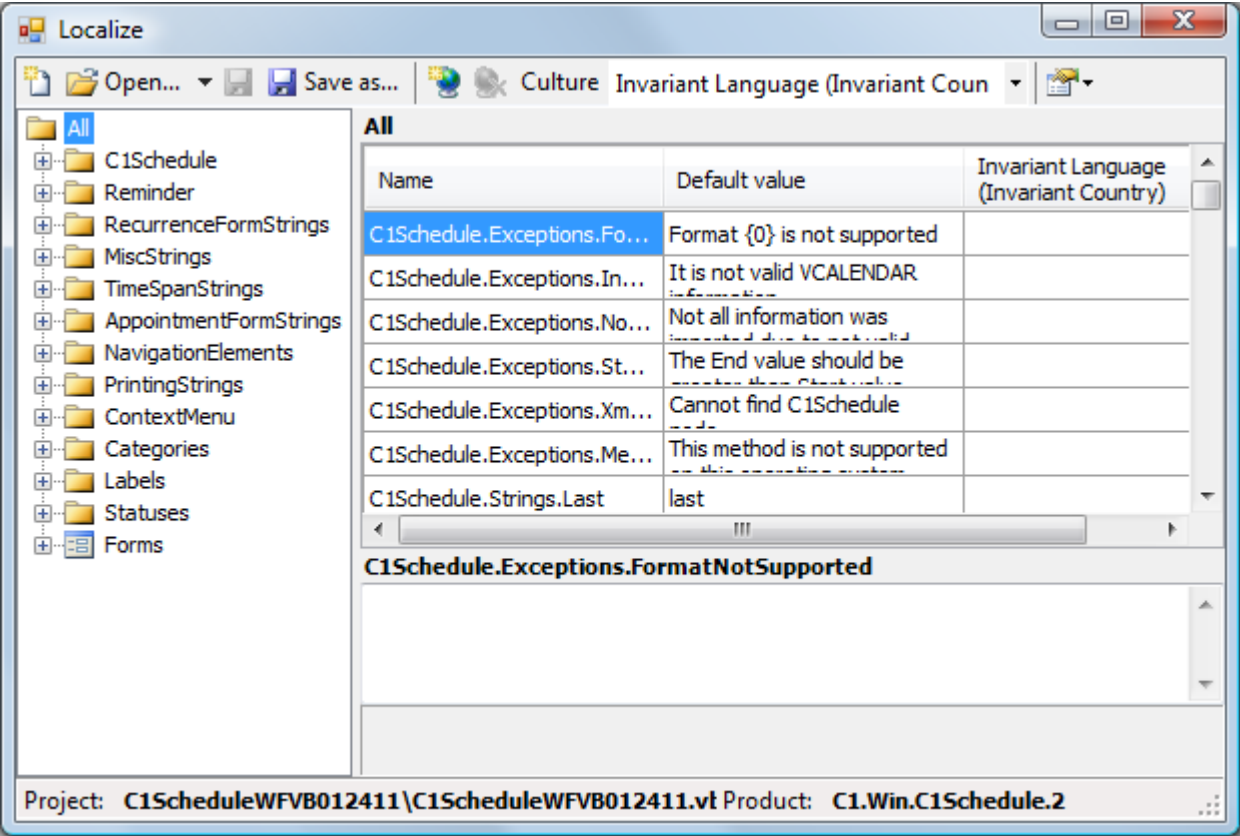
Calendar Settings (Available only when linked to a C1Schedule control)

Clicking the **Calendar Settings** button opens the **Calendar Settings** dialog box where you can customize the settings for the calendar.



Localize

Clicking the **Localize** button opens the **Localize** dialog box where you customize your localization settings. For more information, see [Localization](#).



Working with Scheduler for WinForms

Scheduler for WinForms provides full scheduling functionality with the familiar look of Microsoft Outlook, which can be easily bound to a database or a built-in `DataSource` and then customized to create schedules in your WinForms applications. The following topics explain the main aspects of **Scheduler for WinForms**:

Appointments

The `C1Schedule` control allows you to create, edit, and delete appointments at run time. Appointments can occur one-time or recur over a set amount of time, and reminders can be set so no appointment is missed. Additionally, `C1Schedule` provides [twelve built-in labels](#) and [four availability options](#) to help users manage each appointment, as well as the ability to create custom labels. Appointments can be organized within categories, and resources and contacts for each appointment can be specified.

Data Views

The `C1Schedule` control includes five built-in data views: `DayView`, `WeekView`, `WorkWeekView`, `MonthView`, and `TimeLineView`. Since the data views are built-in, you only have to set the `ViewType` property to change the date view.

Data Mappings

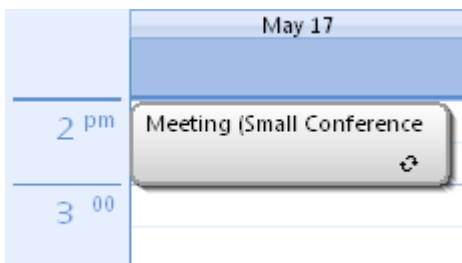
In the **Data Source settings** dialog box in the [C1Schedule Smart Designer](#), which works with the `C1ScheduleStorage` component, you can attach a `DataSource` to the `C1Schedule` control and map to each column in the data table.

Appointments

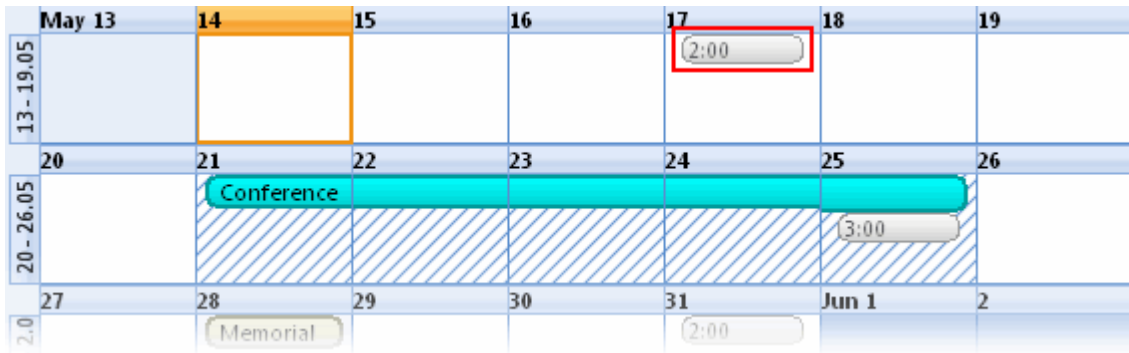
An appointment represents a period of time and detailed information about events that will take place during that period of time. Appointments can span from a specified duration, such as 30 minutes, to multi-day events, and can be set either in code or by binding to a data source, or at run time, either through the context menu or by clicking a specified time on the schedule.

Interval Appointments

Appointments that span a specific duration will appear on the schedule during that duration when in [DayView](#), [WeekView](#), or [WorkWeekView](#) views. For example, the appointment below starts at 2:00 PM and lasts 1 hour.



In the [MonthView](#) view, the appointment will appear on the calendar with the time and subject of the appointment.



Setting an Interval Appointment

Appointments can be created in code or at run time through the **Appointment** dialog box. The following code, added to the **Form_Load** event, creates a new interval appointment:

To write code in Visual Basic

Visual Basic

```
' Add a new appointment.
Dim app As Cl.C1Schedule.Appointment
app = Me.C1Schedule1.DataStorage.AppointmentStorage.Appointments.Add()

' Set some details for the appointment.
app.Subject = "Meeting"
app.Location = "Large Conference Room"
app.Duration = TimeSpan.FromMinutes(45)
app.Start = New Date(2007, 2, 28, 13, 30, 0)
```

To write code in C#

C#

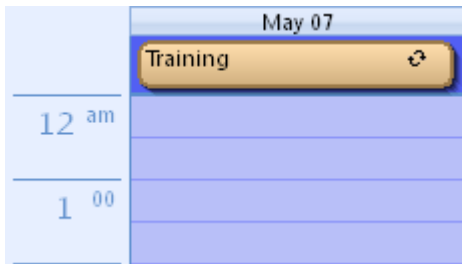
```
// Add a new appointment.
Cl.C1Schedule.Appointment app;
app = this.c1Schedule1.DataStorage.AppointmentStorage.Appointments.Add();

// Set some details for the appointment.
app.Subject = "Meeting";
app.Location = "Large Conference Room";
app.Duration = TimeSpan.FromMinutes(45);
app.Start = new DateTime(2007, 2, 28, 13, 30, 0);
```

At run time, an appointment can be created through the **Appointment** dialog box by setting the **Start time** and **End time** drop-downs to the duration of the appointment. For more information on creating appointments at run time, see [Working with Appointments](#).

All-Day or Multi-Day Events

Appointments that are either an all-day event or a multi-day event will appear on the schedule in the area above the schedule when in DayView, WeekView, or WorkWeekView view. For example, the appointment below represents an all-day event.



In the MonthView view, the appointment will appear on the calendar with the subject of the appointment within a box.



Setting an All-Day Appointment

Appointments can be created in code or at run time through the **Appointment** dialog box. The following code, added to the **Form_Load** event, creates a new all-day appointment:

To write code in Visual Basic

Visual Basic

```
' Create a new appointment.
Dim app As Cl.C1Schedule.Appointment
app = Me.C1Schedule1.DataStorage.AppointmentStorage.Appointments.Add()

' Make the appointment an all day event.
app.AllDayEvent = True

' Set some details for the appointment.
app.Subject = "Training"
app.Location = "Large Conference Room"
```

To write code in C#

C#

```
// Create a new appointment.
Cl.C1Schedule.Appointment app =
this.c1Schedule1.DataStorage.AppointmentStorage.Appointments.Add();

// Make the appointment an all day event.
app.AllDayEvent = true;

// Set some details for the appointment.
app.Subject = "Training";
app.Location = "Large Conference Room";
```

At run time, an appointment can be created through the **Appointment** dialog box by checking the **All day event**

check box. For more information on creating appointments at run time, see [Working with Appointments](#).













Labels

There are twelve predefined labels available in [C1Schedule](#) to assign to appointments. The color of the label is visible in every data view in the C1Schedule control. The following image displays labels in the month view of the C1Schedule control:

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	May 6	7	8	9	10	11	12
6 - 12.05		Training	Training	Training	Training	Training 3:00	
	13	14	15	16	17	18	19
13 - 19.05					2:00		
	20	21	22	23	24	25	26
20 - 26.05		Conference				3:00	
	27	28	29	30	31	Jun 1	2
27.05 - 2.06		Memorial			2:00		

Predefined Labels

The predefined labels include the following:

Label	Color	Index
None		0
Important		1
Business		2
Personal		3
Vacation		4
Deadline		5
Must Attend		6
Travel Required		7
Needs Preparation		8
Birthday		9
Anniversary		10
Phone Call		11

Assigning Predefined Labels to an Appointment

Labels can be assigned in code or at run time through the **Appointment** dialog box. The following code, added to the **Form_Load** event, assigns a **Must Attend** label to an appointment:

To write code in Visual Basic

Visual Basic

```
' Add a new appointment.
Dim app As Cl.ClSchedule.Appointment
app = Me.ClSchedule1.DataStorage.AppointmentStorage.Appointments.Add()

' Set some details for the appointment.
app.Subject = "Meeting"
app.Location = "Large Conference Room"
app.Duration = TimeSpan.FromMinutes(45)
app.Start = New Date(2007, 2, 28, 13, 30, 0)

' Assign a predefined label to the appointment.
app.Label = Me.ClSchedule1.DataStorage.LabelStorage.Labels.Item(6)
```

To write code in C#

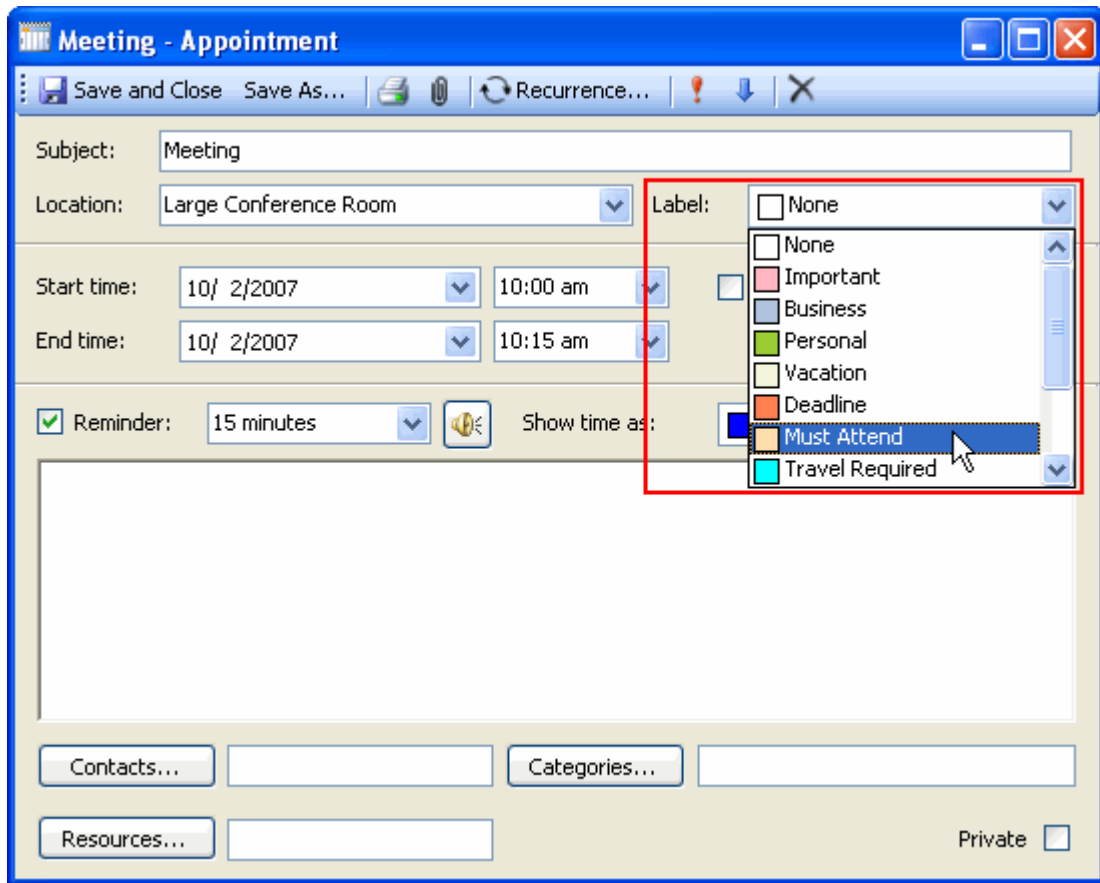
C#

```
// Add a new appointment.
Cl.ClSchedule.Appointment app;
app = this.ClSchedule1.DataStorage.AppointmentStorage.Appointments.Add();

// Set some details for the appointment.
app.Subject = "Meeting";
app.Location = "Large Conference Room";
app.Duration = TimeSpan.FromMinutes(45);
app.Start = new DateTime(2007, 2, 28, 13, 30, 0);

// Assign a predefined label to the appointment.
app.Label = this.ClSchedule1.DataStorage.LabelStorage.Labels[6];
```

At run time, the label can be set in the **Appointment** dialog box by setting the **Label** drop-down arrow and selecting one of the available options. In the image below the **Must Attend** label is selected:



Assigning Custom Labels to an Appointment

You can also add custom labels using the [Label](#) property. The following code, added to the **Form_Load** event, assigns a custom **Meeting** label to an appointment:

To write code in Visual Basic

Visual Basic

```
' Add a new appointment.
Dim app As Cl.C1Schedule.Appointment
app = Me.C1Schedule1.DataStorage.AppointmentStorage.Appointments.Add()

' Set some details for the appointment.
app.Subject = "Meeting"
app.Location = "Large Conference Room"
app.Duration = TimeSpan.FromMinutes(45)
app.Start = New Date(2007, 2, 28, 13, 30, 0)

' Assign a custom label to the appointment.
app.Label = New Cl.C1Schedule.Label(Color.DeepSkyBlue, "Meeting", "Meeting")
```

To write code in C#

C#

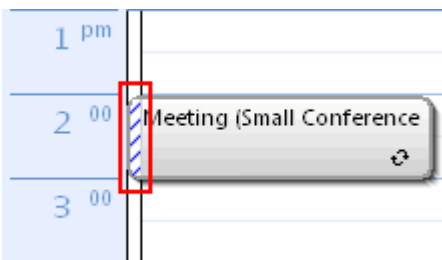
```
// Add a new appointment.
C1.C1Schedule.Appointment app;
app = this.c1Schedule1.DataStorage.AppointmentStorage.Appointments.Add();

// Set some details for the appointment.
app.Subject = "Meeting";
app.Location = "Large Conference Room";
app.Duration = TimeSpan.FromMinutes(45);
app.Start = new DateTime(2007, 2, 28, 13, 30, 0);

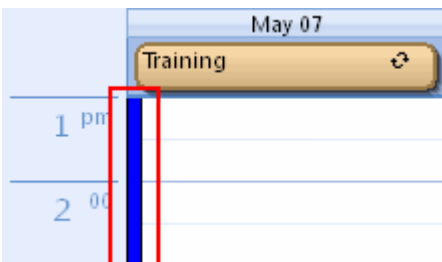
// Assign a custom label to the appointment.
app.Label = new C1.C1Schedule.Label(Color.DeepSkyBlue, "Meeting", "Meeting");
```


Availability

There are four predefined availabilities available in [C1Schedule](#) to assign to appointments: **Busy**, **Free**, **Out of Office**, and **Tentative**. The color of the availability is only visible in the [WorkWeekView](#), [WeekView](#), or [DayView](#) views. For interval appointments, the color appears in the time slot area to the left of the appointment.



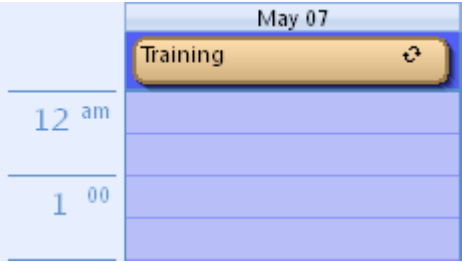
For an all day event, the appointment appears in the all-day area, and the color appears in the time slot area for all appointments for that day.



 **Note:** The Office 2007 Black, Office 2007 Blue, and Office 2007 Silver visual styles do not show the color for the busy status in the appointment or event.

By default, the [Office 2007 Blue](#), [Office 2007 Black](#), and [Office 2007 Silver](#) visual styles have the [ShowDayStatus](#) property set to **False**, which does not display the availability in the time slot area to the left of the appointment when in the [WeekView](#), [WeekView](#), or [DayView](#) views. To enable [ShowDayStatus](#) property for those visual styles, check the **Show day status** check box on the **Day View** node in the **Visual Style** dialog box in the [C1Schedule Smart Designer](#).





Instead of displaying the availability to the left of the appointment, the Office 2007 visual styles set the day or time slot background coloring according to the availability status of appointments.



By default, Office 2007 visual styles have the [HighlightDayStatus](#) property set to **True**. To disable the HighlightDayStatus property, uncheck the **Highlight day status** check box on the **Common** node in the **Visual Style** dialog box in the [C1Schedule Smart Designer](#).

Predefined Availabilities

The predefined availabilities include the following:

Availability	Color	Index
Busy		0
Free		1
Out of Office		2
Tentative		3

Assigning Predefined Availabilities to an Appointment

Availabilities can be assigned in code or at run time through the **Appointment** dialog box. The following code, added the **Form_Load** event, assign a **Tentative** availability to an appointment:

To write code in Visual Basic

```
Visual Basic

' Add a new appointment.
Dim app As C1.C1Schedule.Appointment
app = Me.C1Schedule1.DataStorage.AppointmentStorage.Appointments.Add()

' Set some details for the appointment.
app.Subject = "Meeting"
app.Location = "Large Conference Room"
app.Duration = TimeSpan.FromMinutes(45)
app.Start = New Date(2007, 2, 28, 13, 30, 0)

' Assign a predefined availability to the appointment.
app.BusyStatus =
Me.C1Schedule1.DataStorage.StatusStorage.Statuses(C1.C1Schedule.StatusTypeEnum.Tentative)
' OR app.BusyStatus = Me.C1Schedule1.DataStorage.StatusStorage.Statuses.Item(3)

' Set the view to Work Week to view the availability.
Me.C1Schedule1.ViewType = C1.Win.C1Schedule.ScheduleViewEnum.WorkWeekView
```

To write code in C#

```
C#

// Add a new appointment.
C1.C1Schedule.Appointment app;
```

```
app = this.clSchedule1.DataStorage.AppointmentStorage.Appointments.Add();

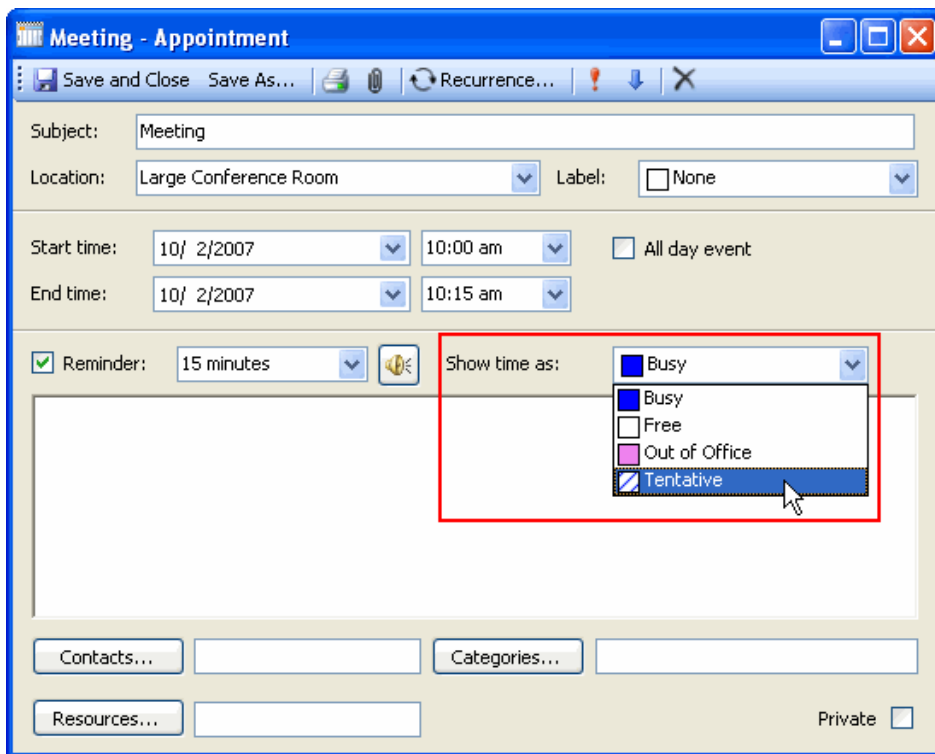
// Set some details for the appointment.
app.Subject = "Meeting";
app.Location = "Large Conference Room";
app.Duration = TimeSpan.FromMinutes(45);
app.Start = new DateTime(2007, 2, 28, 13, 30, 0);

// Assign a predefined availability to the appointment.
app.BusyStatus =
this.clSchedule1.DataStorage.StatusStorage.Statuses[C1.ClSchedule.StatusTypeEnum.Tentative];
// OR app.BusyStatus = this.clSchedule1.DataStorage.StatusStorage.Statuses[3];

// Set the view to Work Week to view the availability.
this.clSchedule1.ViewType = C1.Win.ClSchedule.ScheduleViewEnum.WorkWeekView;
```

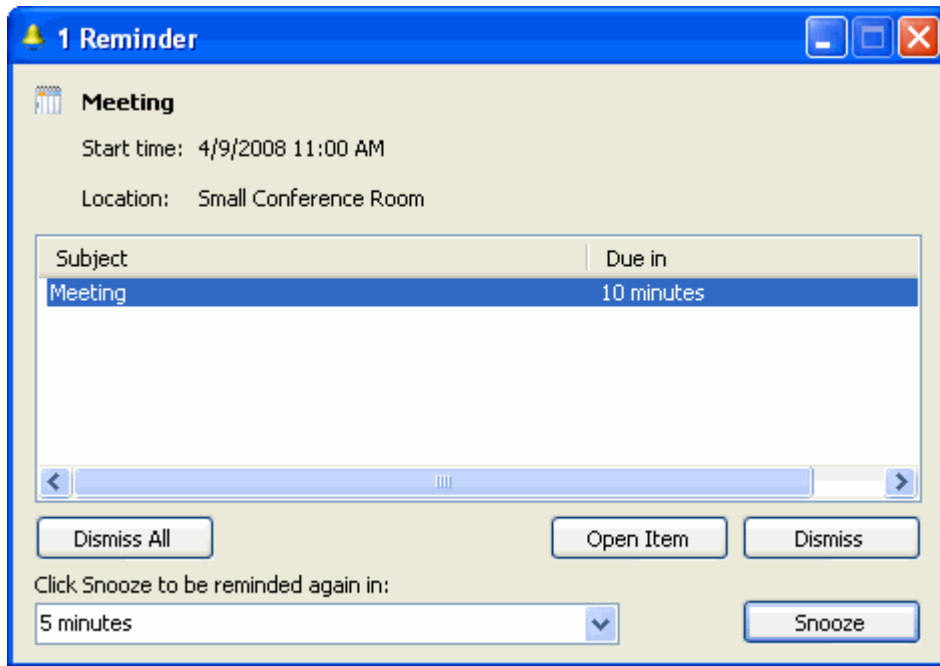
Note that either the [StatusTypeEnum](#) enumeration or the Index can be used to set the availability.

At run time, the availability can be set in the **Appointment** dialog box by setting the **Show time as** drop-down to one of the available options.



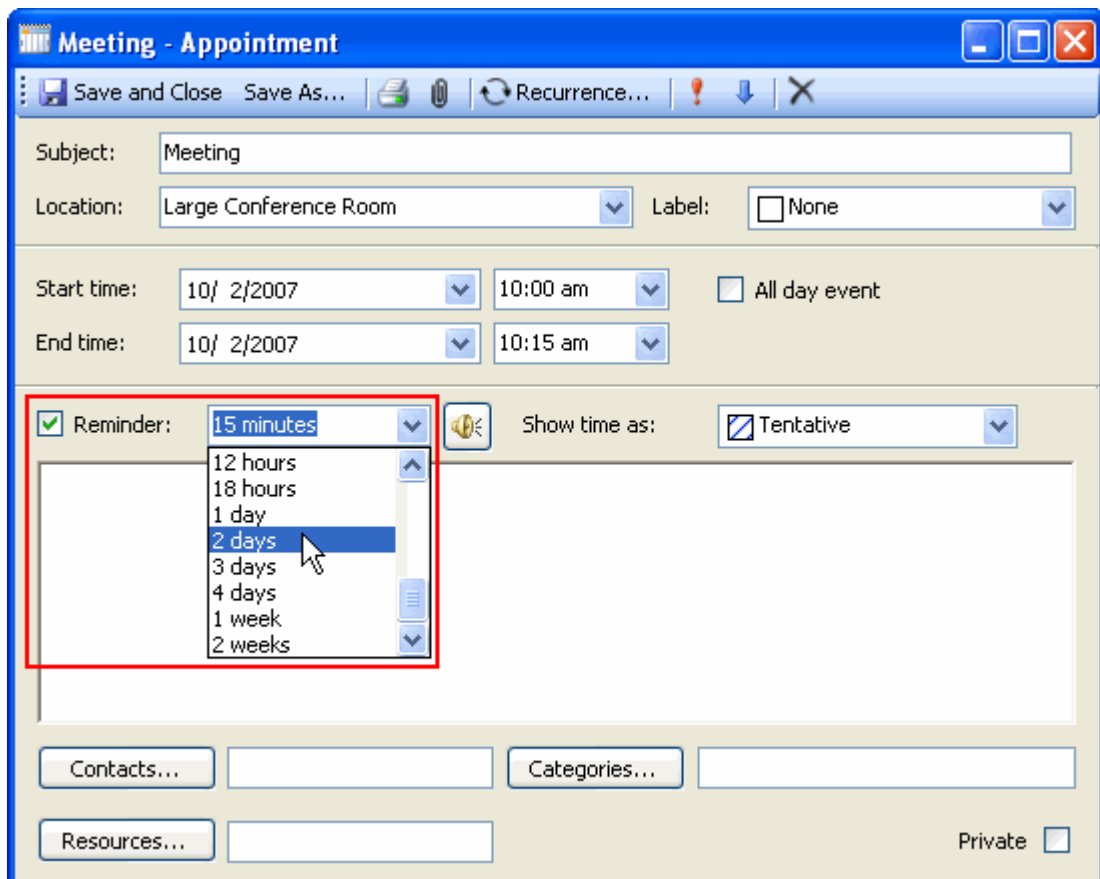
Reminders

Appointment reminders display a **Reminder** dialog box at a specified time before the appointment occurs. In the **Reminder** dialog box, you have the option of dismissing one or more appointments (if multiple appointments are due), opening the item, or resetting the reminder to appear again in a set amount of time.



Note: Pressing ENTER in **Reminder** dialog box when focus is on the snooze options combo box or on the reminders list, sets the selected reminder to snooze for the selected time period before reminding again.

Reminders can be set when creating an appointment by checking the **Reminder** check box, setting the amount of time before the appointment that you would like the reminder to appear, and optionally choosing a sound file to play when the reminder alert appears.



Reminder Properties

The following reminder properties can be set through code:

Property	Description
Reminder	Gets the Reminder object which is associated with the appointment. To associate an appointment with the reminder, set its ReminderSet property to true. Once this property is set to true, a new Reminder object is created and assigned to the Reminder property.
ReminderOverrideDefault	Gets or sets the Boolean value indicating if the associated Reminder object has non-default settings. True if the reminder overrides the default reminder behavior for the appointment. You must set the ReminderOverrideDefault property to validate the ReminderPlaySound and the ReminderSoundFile properties.
ReminderPlaySound	Gets or sets the Boolean value indicating if the reminder should play a sound when it occurs for this appointment or task. The ReminderPlaySound property must be set in order to validate the ReminderSoundFile property. This property is only valid if the ReminderOverrideDefault property is set to true.
ReminderSet	Gets or sets the Boolean value indicating whether a reminder is associated with the appointment. Once this property is set to true, a new Reminder object is created and assigned to the Reminder property.
ReminderSoundFile	Gets or sets the String value indicating the path and file name of the sound file to play when the reminder occurs for the appointment. This property is only valid if the ReminderOverrideDefault and ReminderPlaySound properties are set to true.
ReminderTimeBeforeStart	Gets or sets the TimeSpan value indicating the interval of time the reminder should occur prior to the start of the appointment.

The following reminder properties can be set in the **Appointment** dialog box at run time:

Property	Description
ReminderSet	Gets or sets the Boolean value indicating whether a reminder is associated with the appointment. Once this property is set to true, a new Reminder object is created and assigned to the Reminder property.
ReminderSoundFile	Gets or sets the String value indicating the path and file name of the sound file to play when the reminder occurs for the appointment. This property is only valid if the ReminderOverrideDefault and ReminderPlaySound properties are set to true.

ReminderTimeBeforeStart

Gets or sets the TimeSpan value indicating the interval of time the reminder should occur prior to the start of the appointment.

Contacts

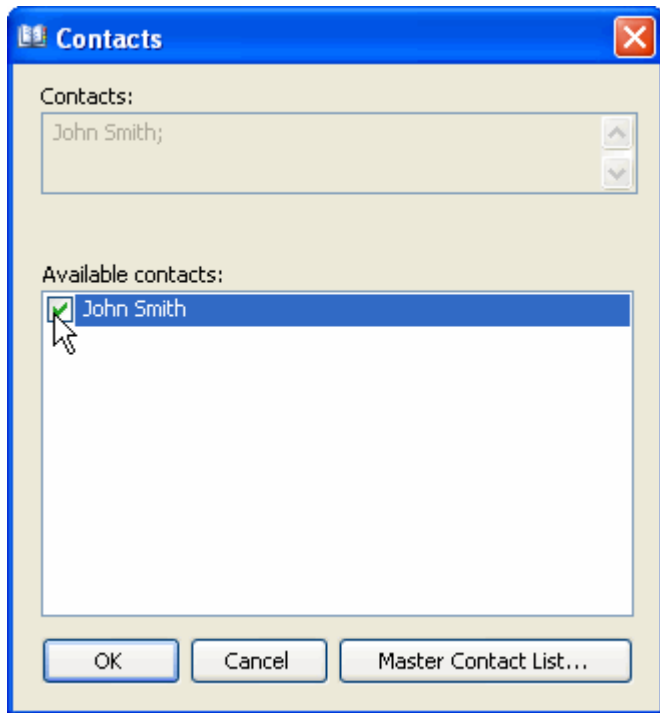
A contact contains personal information for a person. Contacts, which are stored in the [ContactCollection](#) class, are optional and an appointment can have one or more contacts assigned to it.

Assigning Contacts to an Appointment

At run time, assigning a contact or contacts to an appointment can be done through the **Contacts** dialog box. Clicking the **Contacts** button in the **Appointment** dialog box opens the **Contacts** dialog box.

The screenshot shows the 'Untitled - Appointment' dialog box. The 'Contacts...' button is highlighted with a red rectangle, and a mouse cursor is pointing at it. The text 'John Smith;' is visible in the text area next to the 'Contacts...' button.

In the **Contacts** dialog box, check the contacts you would like to add to the appointment.



Note: By default, the **Contacts** dialog box does not contain any contacts. For information on how to add contacts to the **Contacts** dialog box, see [Adding Contacts to the Master Contact List](#).

Adding Contacts to the Master Contact List

C1Schedule supports contacts created in code or at run time through the **Master Contact List** dialog box. The following code, added to the **Form_Load** event, creates a contact for **John Smith**:

To write code in Visual Basic

Visual Basic

```
' Create the contact.
Dim control As C1.C1Schedule.ContactCollection
control = Me.C1Schedule1.DataStorage.ContactStorage.Contacts
Dim contact As New C1.C1Schedule.Contact
contact.Text = "John Smith"

' Insert the contact into the Master Contact List.
control.Insert(0, contact)
```

To write code in C#

C#

```
// Create the contact.
C1.C1Schedule.ContactCollection control;
control = this.c1Schedule1.DataStorage.ContactStorage.Contacts;
C1.C1Schedule.Contact contact = new C1.C1Schedule.Contact();
contact.Text = "John Smith";
```



```
// Insert the contact into the Master Contact List.  
contcol.Insert(0, contact);
```

At run time, creating a contact can be added through the **Master Contact List** dialog box. Clicking the **Master Contact List** button in the **Contacts** dialog box opens the **Master Contact List** dialog box where you add or delete contacts.



After you create a contact or contacts, add the contact to the appointment in the **Contacts** dialog box. For more details on how to add a contact to an appointment, see [Assigning Contacts to an Appointment](#).

Categories

A category is a keyword or a phrased used to help you organize your appointments. There are 20 predefined categories available in **C1Schedule** to assign to appointments. Categories, which are stored in the [CategoryCollection](#) class, are optional and an appointment can one or more categories assigned to it.

Predefined Categories

The predefined categories include the following:

Assigning Predefined Categories to an Appointment

Categories can be assigned in code or at run time through the **Categories** dialog box. The following code, added to the **Form_Load** event, assign the **Strategies** category to an appointment:

To write code in Visual Basic

Visual Basic

```
' Add an appointment  
Dim app As C1.C1Schedule.Appointment  
app = Me.C1Schedule1.DataStorage.AppointmentStorage.Appointments.Add()  
  
' Set some details for the appointment.
```

```
app.Subject = "Meeting"
app.Location = "Large Conference Room"
app.Duration = TimeSpan.FromMinutes(45)
app.Start = New Date(2007, 2, 28, 13, 30, 0)

' Assign a category to the appointment.
Dim category As Cl.ClSchedule.Category
category = Me.ClSchedule1.DataStorage.CategoryStorage.Categories.Item(15)
app.Categories.Add(category)
```

To write code in C#

```
C#

// Add a new appointment.
Cl.ClSchedule.Appointment app;
app = this.clSchedule1.DataStorage.AppointmentStorage.Appointments.Add();

// Set some details for the appointment.
app.Subject = "Meeting";
app.Location = "Large Conference Room";
app.Duration = TimeSpan.FromMinutes(45);
app.Start = new DateTime(2007, 2, 28, 13, 30, 0);

// Assign a category to the appointment.
Cl.ClSchedule.Category category;
category = this.clSchedule1.DataStorage.CategoryStorage.Categories[15];
app.Categories.Add(category);
```

At run time, assigning a category or categories to an appointment can be done through the **Categories** dialog box. Clicking the **Categories** button in the **Appointment** dialog box opens the **Categories** dialog box.

Untitled - Appointment

Save and Close Save As... Recurrence... ?

Subject:

Location: Label: ☐ None

Start time: 10/ 2/2007 10:00 am ☐ All day event

End time: 10/ 2/2007 10:15 am

☒ Reminder: 15 minutes Show time as: ☒ Busy

Contacts... Categories... Strategies;

Resources... Private ☐

In the **Categories** dialog box, you can check the categories you would like to add to the appointment.

Categories

Item(s) belong to these categories:

Strategies;

Available categories:

- ☐ Miscellaneous
- ☐ Personal
- ☐ Phone Calls
- ☐ Status
- ☒ Strategies
- ☐ Suppliers
- ☐ Time & Expenses
- ☐ VIP
- ☐ Waiting

OK Cancel Master Category List...

Adding Custom Categories to the Master Category List

C1Schedule also supports custom categories created in code or at run time through the **Master Category List** dialog box. The following code, added to the **Form_Load** event, creates a custom **Meetings** category:

To write code in Visual Basic

Visual Basic

```
' Create the Meetings category.
Dim catcol As C1.C1Schedule.CategoryCollection
catcol = Me.C1Schedule1.DataStorage.CategoryStorage.Categories
Dim meetings As New C1.C1Schedule.Category
meetings.Text = "Meetings"

' Insert the Meetings category to the Master Category List.
catcol.Insert(0, meetings)
```

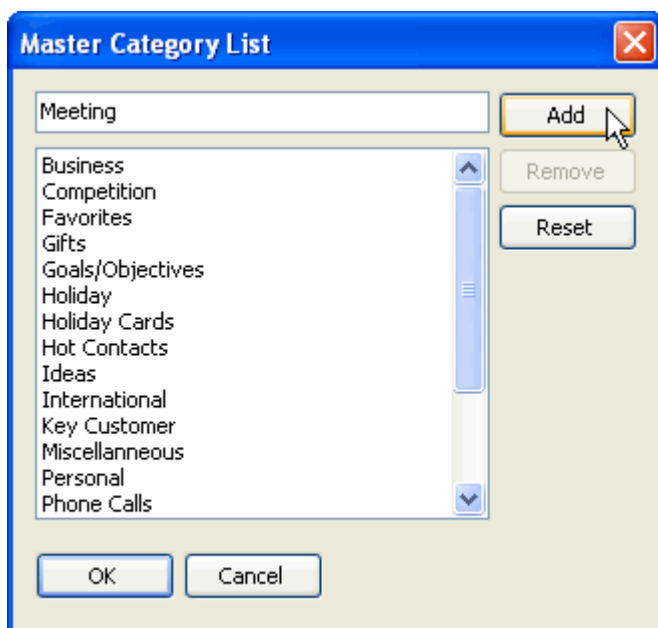
To write code in C#

C#

```
// Create the Meetings category.
C1.C1Schedule.CategoryCollection catcol;
catcol = this.c1Schedule1.DataStorage.CategoryStorage.Categories;
C1.C1Schedule.Category meetings = new C1.C1Schedule.Category();
meetings.Text = "Meetings";

// Insert the Meetings category into the Master Category List.
catcol.Insert(0, meetings);
```

At run time, creating a custom category or categories can be done through the **Master Category List** dialog box. Clicking the **Master Category List** button in the **Categories** dialog box opens the **Master Category List** dialog box where you add, remove, or reset categories.



After you create a custom category or categories, add the categories to the appointment in the **Categories** dialog box the same way you would add a predefined category. For more details on how to add a category to an appointment, see [Assigning Predefined Categories to an Appointment](#).

Resources

A resource is a keyword or a phrase that defines a source of support for a particular task. Resources, which are stored in the [ResourceCollection](#) class, are optional and an appointment can have one or more resources assigned to it.

For example, resources for a meeting could include a digital projector, dry erase board, whiteboard, laser pointer, overhead projector, chalkboard, easel board, television, vcr/dvd player, and video teleconference system.

Assigning Resources to an Appointment

Assigning a resource can be done through code. The following code, added to the **Form_Load** event, creates a **Digital Projector** resource and assigns it to the appointment:

To write code in Visual Basic

Visual Basic

```
' Add an appointment
Dim app As Cl.C1Schedule.Appointment
app = Me.C1Schedule1.DataStorage.AppointmentStorage.Appointments.Add()

' Set some details for the appointment.
app.Subject = "Meeting"
app.Location = "Large Conference Room"
app.Duration = TimeSpan.FromMinutes(45)
app.Start = New Date(2007, 2, 28, 13, 30, 0)

' Create the Digital Projector resource.
Dim rescol As Cl.C1Schedule.ResourceCollection
rescol = Me.C1Schedule1.DataStorage.ResourceStorage.Resources
Dim roomres As New Cl.C1Schedule.Resource
roomres.Text = "Digital Projector"

' Insert the resource into the Master Resource List.
rescol.Insert(0, roomres)

' Assign the resource to the appointment.
app.Resources.Add(roomres)
```

To write code in C#

C#

```
// Add a new appointment.
Cl.C1Schedule.Appointment app;
app = this.c1Schedule1.DataStorage.AppointmentStorage.Appointments.Add();

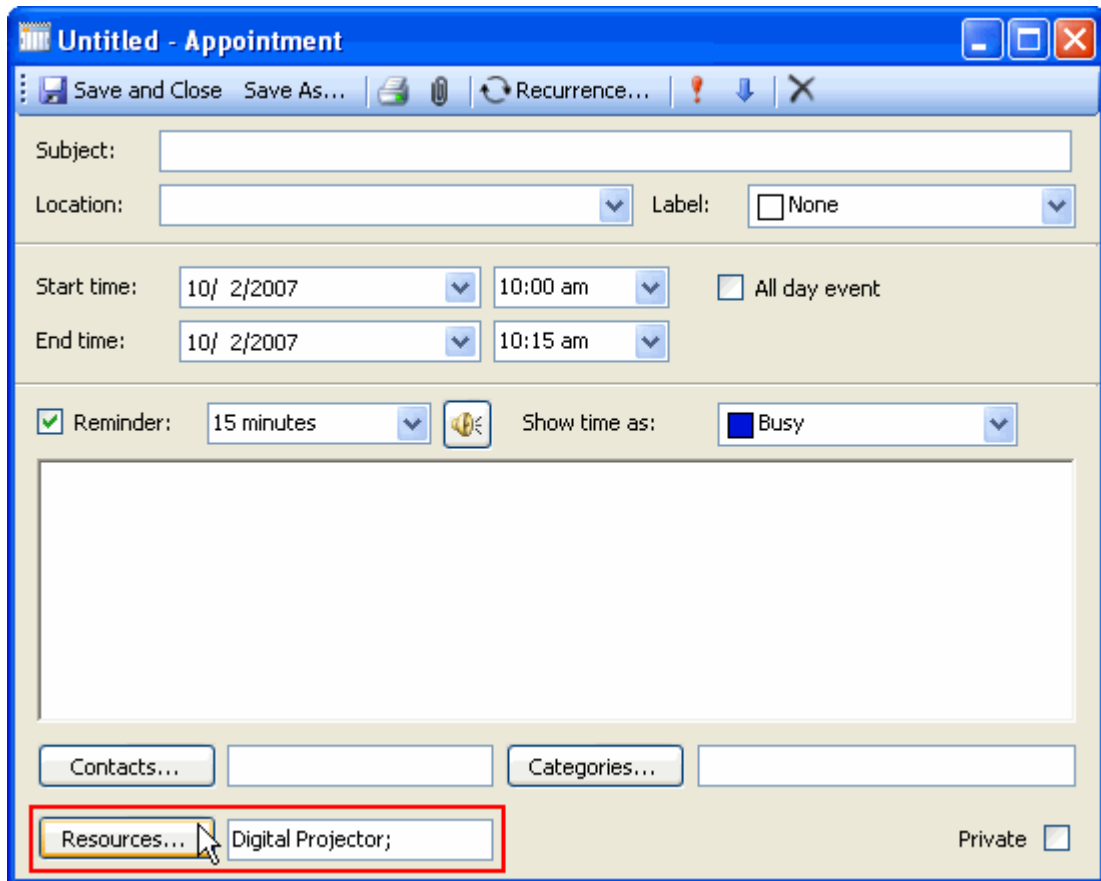
// Set some details for the appointment.
app.Subject = "Meeting";
app.Location = "Large Conference Room";
app.Duration = TimeSpan.FromMinutes(45);
app.Start = new DateTime(2007, 2, 28, 13, 30, 0);
```

```
// Create the Digital Projector resource.
C1.C1Schedule.ResourceCollection rescol;
rescol = this.c1Schedule1.DataStorage.ResourceStorage.Resources;
C1.C1Schedule.Resource roomres = new C1.C1Schedule.Resource();
roomres.Text = "Digital Projector";

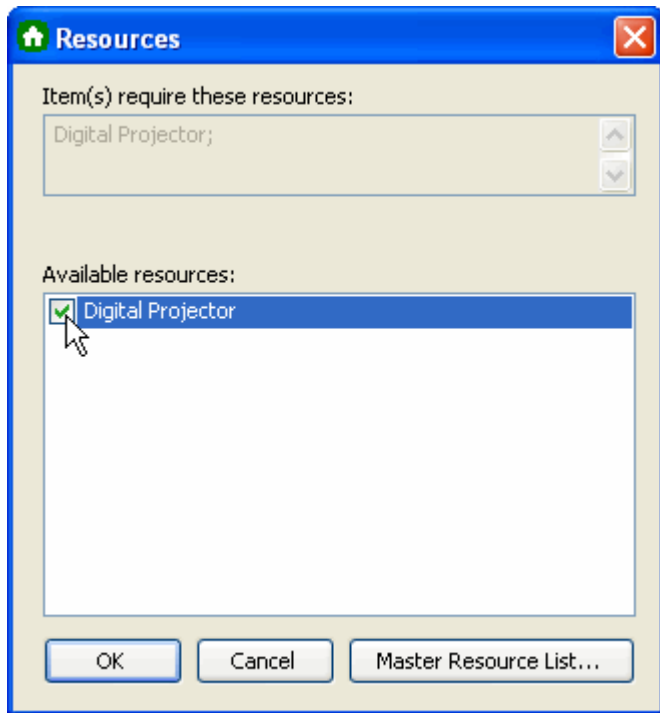
// Insert the resource into the Master Resource List.
rescol.Insert(0, roomres);

// Assign the resource to the appointment.
app.Resources.Add(roomres);
```

At run time, assigning a resource or resources to an appointment can be done through the **Resources** dialog box. Clicking the **Resources** button in the **Appointment** dialog box opens the **Resources** dialog box.



In the **Resources** dialog box, you can check the categories you would like to add to the appointment.



Note: By default, the **Resources** dialog box does not contain any resources. For information on how to add resources to the **Resources** dialog box, see [Adding Resources to the Master Resource List](#).

Adding Resources to the Master Resource List

Scheduler for WinForms supports resources created in code or at run time through the **Master Resource List** dialog box. The following code, added to the **Form_Load** event, creates a **Digital Projector** resource:

To write code in Visual Basic

Visual Basic

```
' Create the Digital Projector resource.
Dim rescol As C1.C1Schedule.ResourceCollection
rescol = Me.C1Schedule1.DataStorage.ResourceStorage.Resources
Dim roomres As New C1.C1Schedule.Resource
roomres.Text = "Digital Projector"

' Insert the resource into the Master Resource List.
rescol.Insert(0, roomres)
```

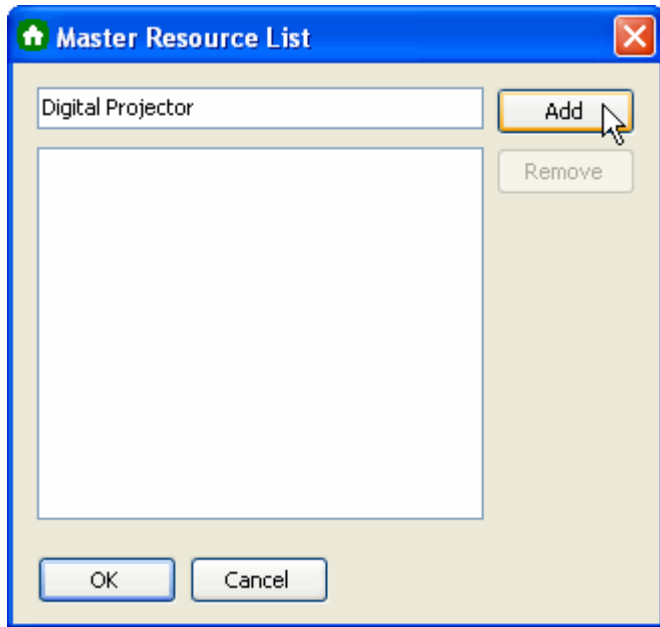
To write code in C#

C#

```
// Create the Digital Projector resource.
C1.C1Schedule.ResourceCollection rescol;
rescol = this.c1Schedule1.DataStorage.ResourceStorage.Resources;
C1.C1Schedule.Resource roomres = new C1.C1Schedule.Resource();
roomres.Text = "Digital Projector";
```

```
// Insert the resource into the Master Resource List.
rescol.Insert(0, roomres);
```

At run time, creating a resource or resources can be done through the **Master Resource List** dialog box. Clicking the **Master Resource List** button in the **Resources** dialog box opens the **Master Resource List** dialog box where you add or remove resources.



After you create a resource or resources, add the resource to the appointment in the **Resources** dialog box. For more details how to add a resource to an appointment, see [Assigning Resources to an Appointment](#).

Appointment Actions

The **C1Schedule** control allows you to create and set appointment actions using the **Action** property. The Action property determines an action to take when the current system time reaches the start of the owning **Appointment** object. By default, control tries to start the system process executing an action. To change this behavior you can set the **Cancel** property to **True** in the event handler for the **AppointmentCustomAction** event.

Appointment actions let you set behaviors such as executing a local application with specified parameters or navigating to a specific website at an appointment's start time. For example, add the following code to the **AppointmentAdded** event to navigate to ComponentOne's Web site at a new appointment's start time:

To write code in Visual Basic

Visual Basic

```
Private Sub C1Schedule1_AppointmentAdded(ByVal sender As Object, ByVal e As
AppointmentEventArgs) Handles C1Schedule1.AppointmentAdded
    ' Create a new action.
    Dim action As New Action()
    action.Command = "http://www.componentone.com"
    ' Set appointment action.
    e.Appointment.Action = action
End Sub
```

To write code in C#

C#

```
private void c1Schedule1_AppointmentAdded(object sender, AppointmentEventArgs e)
{
    // Create a new action.
    Action action = new Action();
    action.Command = "http://www.componentone.com";
    // Set appointment action.
    e.Appointment.Action = action;
}
```

Data Views

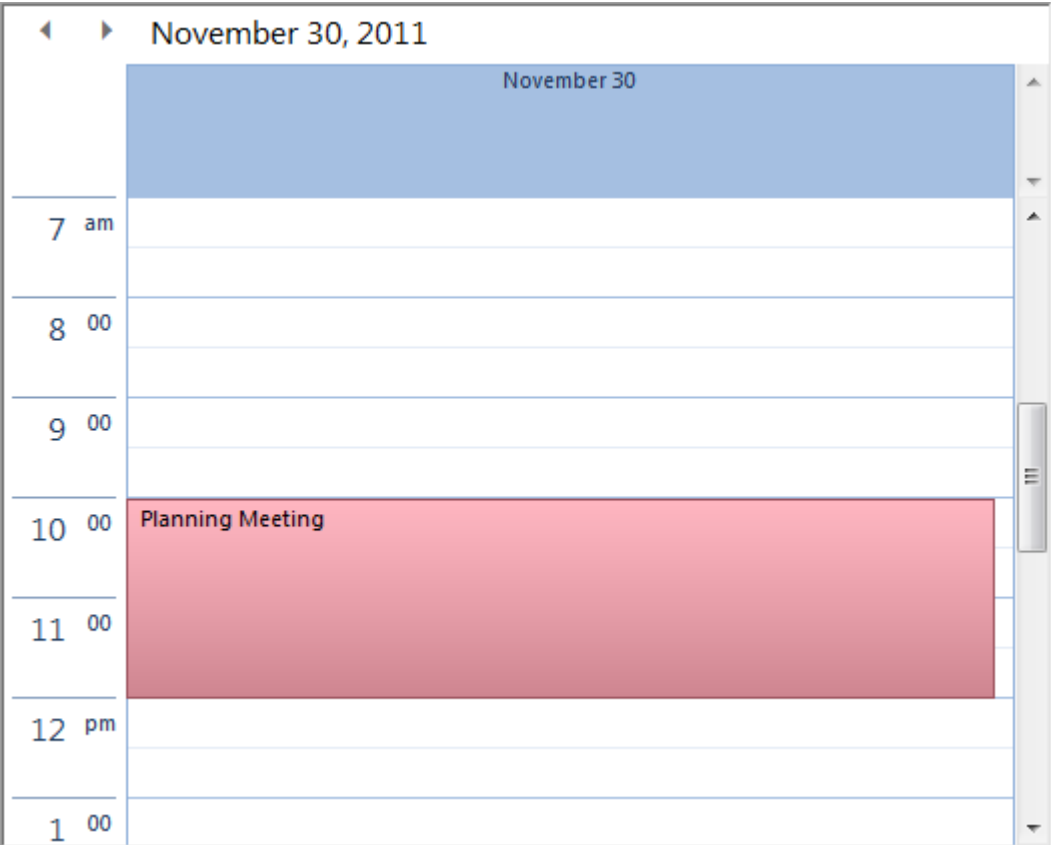
The [C1Schedule](#) control supports different calendar views. There are five predefined data views:

View	Description
DayView	Displays a detailed view showing appointments for a particular day.
TimeLineView	Displays appointments in a horizontal time line.
MonthView	Displays appointments for one or more months.
WeekView	Displays appointments for specified work days.
WorkWeekView	Displays appointments for any given weekly period. The default is Monday through Friday.

Data views can be set in the smart designers, smart tags, or in code. For details on how to change the default data view, see [Changing the Default Data View](#).

Day View

The following image displays the Day view:



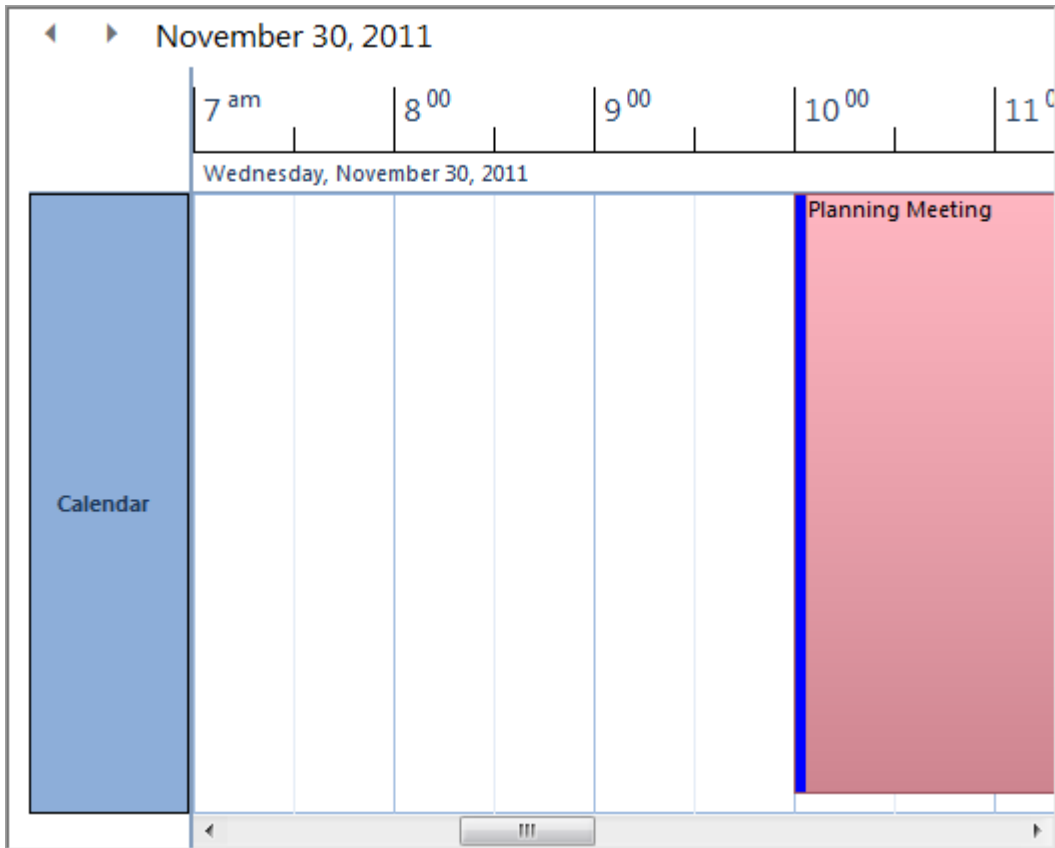
Month View

The following image displays the Month view:



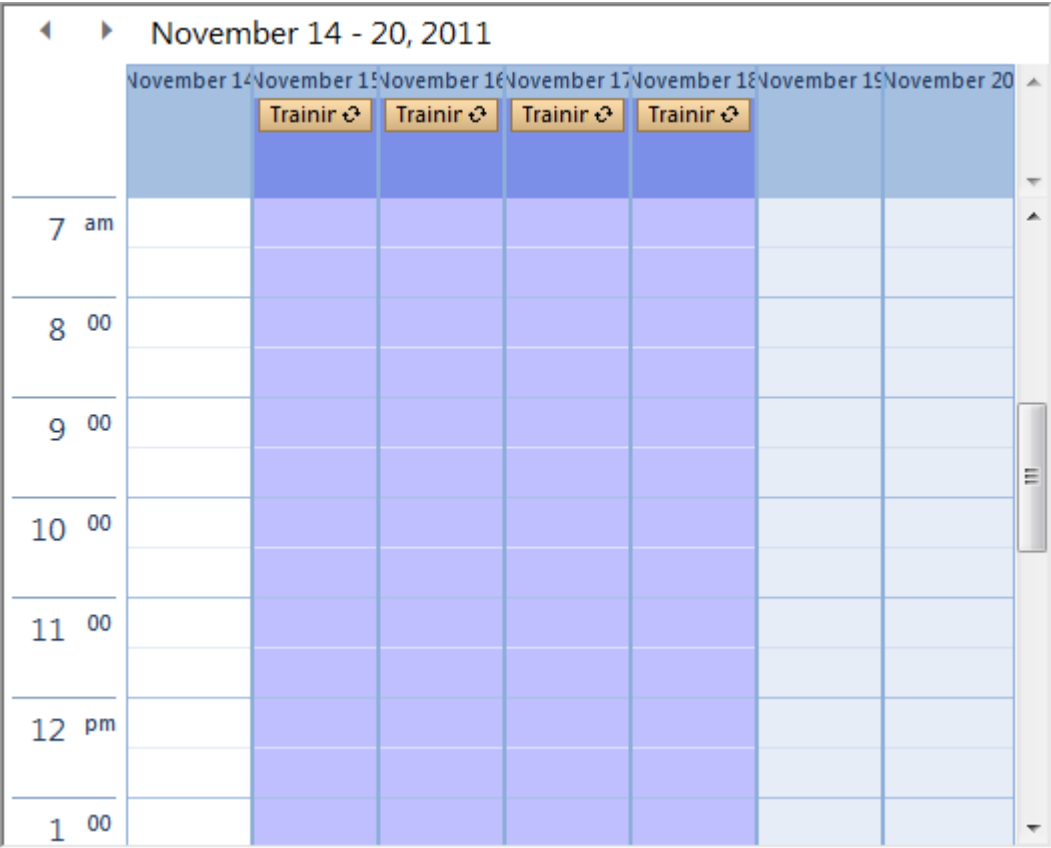
TimeLine View

The following image displays the TimeLine view:



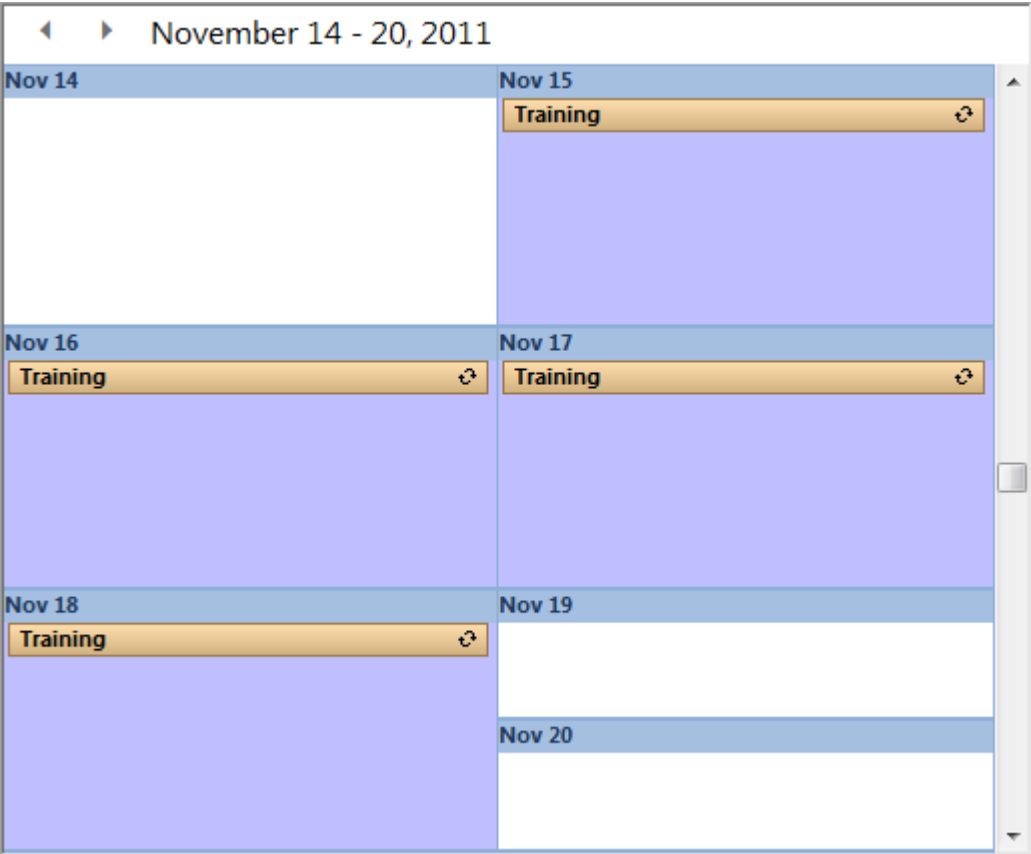
Week View (Office 2007 Style)

The following image displays the Office 2007 style week view. This is the default for the Office 2007 Visual Styles, which have the [WeekViewStyle](#) property is set to [Office2007](#) and the [ViewType](#) property set to [WeekView](#):



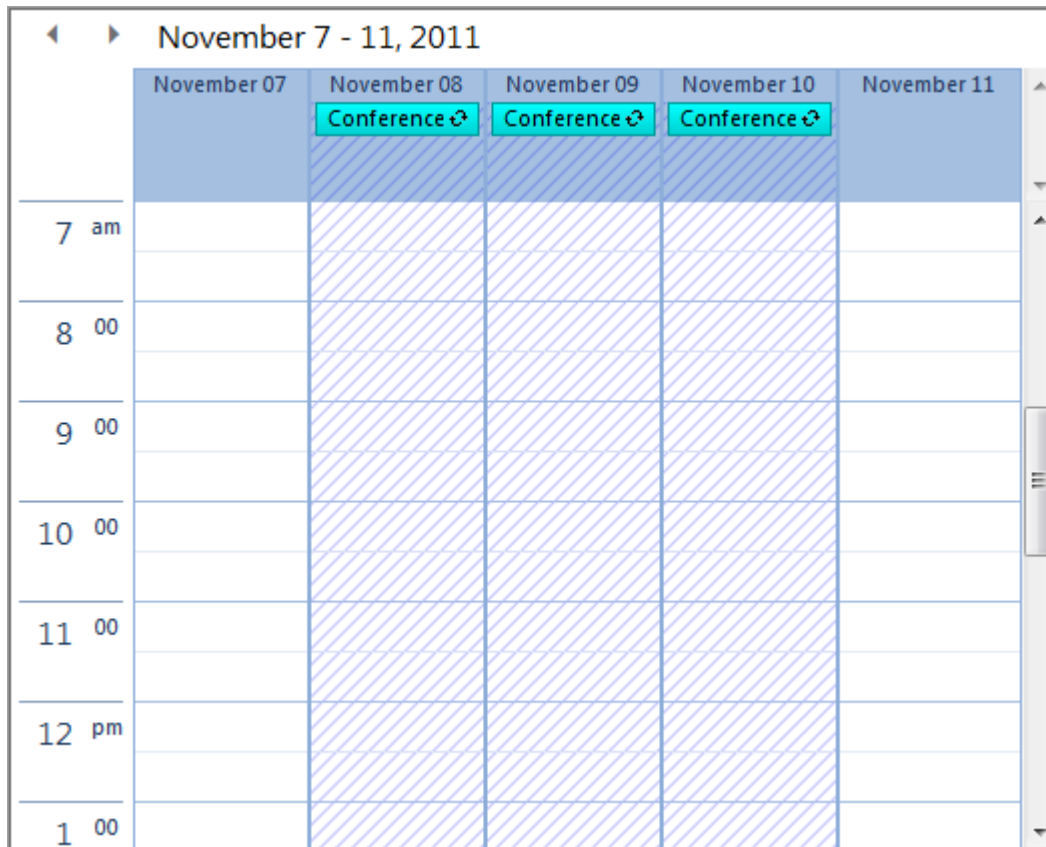
Week View (Office 2003 Style)

The following image displays the Office 2003 style week view. This is the default for the Office 2003 Visual Styles, which have the `WeekViewStyle` property is set to `Office2003` and the `ViewType` property set to `WeekView`:



Work Week View

The following image displays the Work Week view:



Changing the Default Data View

To change the default data view at design time, set the [ViewType](#) property either in the Smart Designer, tasks menu, or in code.

In the Smart Designer

1. On the **C1Schedule Smart Designer**, click the **Visual Style** button. For more information on accessing the **C1Schedule Smart Designer**, see [C1Schedule Smart Designer](#).
2. In the **Visual Style** dialog box, click the **Month View** button.
3. Close the dialog box.

In the Tasks Menu

1. Open the **C1Schedule Tasks** menu. For information accessing the **C1Schedule Tasks** menu, see [C1Schedule Tasks Menu](#).
2. Click the drop-down arrow in the **Default View** drop-down, and select **MonthView**.
3. Close the **C1Schedule Tasks** menu.

In Code

Add the following code to the **Form_Load** event to set the default ViewType property to [MonthView](#):

To write code in Visual Basic

```
Visual Basic
```

```
Me.C1Schedule1.ViewType = C1.Win.C1Schedule.ScheduleViewEnum.MonthView
```

To write code in C#

```
C#
this.c1Schedule1.ViewType = C1.Win.C1Schedule.ScheduleViewEnum.MonthView;
```

Data Mappings

The easiest way to set mappings is to use the **Data Source settings** dialog box in the **C1Schedule Smart Designer**. For more information on accessing the **C1Schedule Smart Designer**, see [C1Schedule Smart Designer](#).

From this dialog box, you can set **DataSource** and **DataMember** properties for all **Scheduler for WinForms** storages. After setting the data source and data member, set the required and optional fields for the storage mapping.

Note: Required field names will appear in green once they have been successfully mapped. Once all required fields have been successfully mapped the **BoundMode** text will also appear in green.

All objects in the **Scheduler for WinForms**' object model are identified by either **Id** or **Index** property. To allow data to be correctly restored from your data sources you should specify mapping either for the [IdMapping](#) or [IndexMapping](#) properties, which makes sense if you bind to your data storages of Resources, Labels, Contacts or

Categories. [C1Schedule](#) saves only Ids or Indexes of these objects in the AppointmentStorage. If there is no mapping for IdMapping or IndexMapping, C1Schedule will not be able to identify the appointment's properties correctly in storages initialized by your data.

To map a data table to the C1Schedule control, first set either the **Id mapping** drop-down list to a Globally Unique Identifier (GUID) field or the **Index mapping** drop-down list to an Integer field. For more information on creating a database to use with the C1Schedule control, see [Creating a Bindable Microsoft Access Database](#).

Appointment Storage Properties Tab

The following fields are available for mapping to the [AppointmentStorage](#):

Mapping	Required/Optional	Data Type
Id mapping	Optional	GUID
Index mapping	Optional	Integer
Properties	Optional	Text
Body	Required	Text
End	Required	Date/Time
Location	Required	Text
Start	Required	Date/Time
Subject	Required	Text
Owner ID	Optional	GUID
Owner Index	Optional	Integer

Category Storage Properties Tab

The following fields are available for mapping to the [CategoryStorage](#):

Mapping	Required/Optional	Data Type
Id mapping	Optional	GUID
Index mapping	Optional	Integer
Caption Mapping	Optional	Text
Text Mapping	Required	Text

Contact Storage Properties Tab

The following fields are available for mapping to the [ContactStorage](#):

Mapping	Required/Optional	Data Type
Id mapping	Optional	GUID
Index mapping	Optional	Integer
Caption Mapping	Optional	Text
Color Mapping	Optional	Text

Mapping	Required/Optional	Data Type
Text Mapping	Required	Text

Label Storage Properties Tab

The following fields are available for mapping to the [LabelStorage](#):

Mapping	Required/Optional	Data Type
Id mapping	Optional	GUID
Index mapping	Optional	Integer
Caption Mapping	Optional	Text
Color Mapping	Optional	Text
Text Mapping	Required	Text

Resource Storage Properties Tab

The following fields are available for mapping to the [ResourceStorage](#):

Mapping	Required/Optional	Data Type
Id mapping	Optional	GUID
Index mapping	Optional	Integer
Caption Mapping	Optional	Text
Color Mapping	Optional	Text
Text Mapping	Required	Text

Status Storage Properties Tab

The following fields are available for mapping to the [StatusStorage](#):

Mapping	Required/Optional	Data Type
Id mapping	Optional	GUID
Index mapping	Optional	Integer
Caption Mapping	Optional	Text
Color Mapping	Optional	Text
Text Mapping	Required	Text

Localization

All end user visible strings in **Scheduler for WinForms** can now be localized (translated) by the developer. **Scheduler for WinForms** localization is based on the same approach as the standard localization of .NET Windows forms employed by the **Localizable** property. As with Windows Forms, you can create a set of resource files for the **Scheduler for WinForms** assembly. You can create separate resource files, with the extension .resx, for each required culture. When the application runs you can switch between those resources and between languages. All parts of your

application using components from a **Scheduler for WinForms** DLL must use the same localization resource.

Localization file conventions

It is recommended that the following conventions are followed when .resx files are created:

- All .resx files should be placed in the **C1LocalizedResources** subfolder of your project.
- Files should be named as follows:

XXX.YYY.resx, where:

- XXX is the name of the Component one assembly.
- YYY is the culture code of the resource. If your translation is only for the invariant culture, the .resx file does not need to contain a culture suffix.

For example:

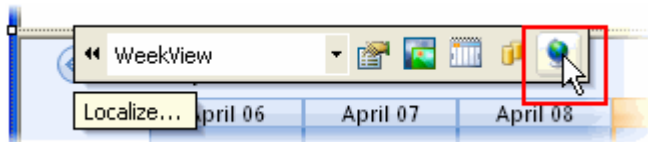
- C1.Win.C1Schedule.2.de.resx - German (de) resource for C1.Win.C1Schedule.2 assembly.
- C1.Win.C1Schedule.2.resx - Invariant culture resource for C1.Win.C1Schedule.2 assembly.

Note that if you create a culture in the [Localize dialog box](#), these conventions will be followed in the file that is created for you.

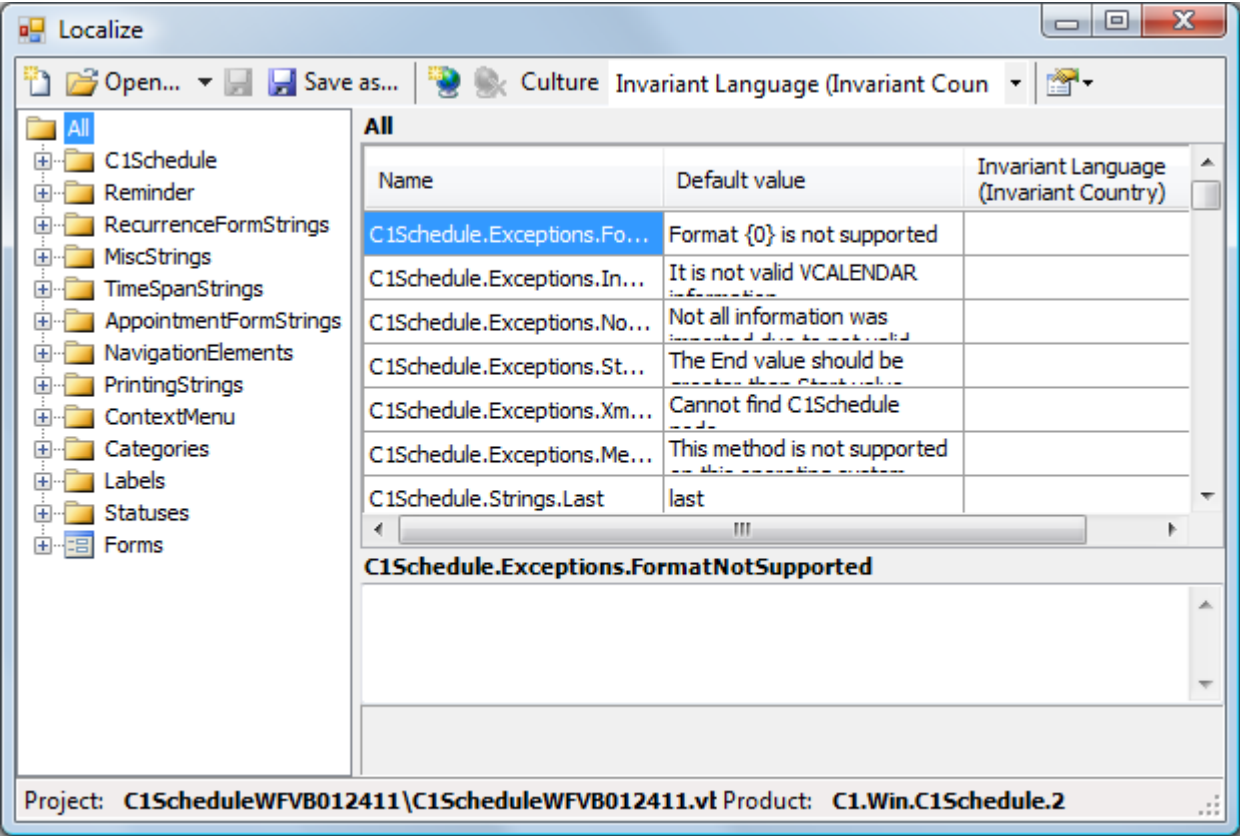
Localize Dialog Box

To localize **C1Schedule** components in your application, select **Localize** in the [C1Schedule](#) or [C1Calendar](#) tasks or context menus, or select **Localize** in the C1Schedule or C1Calendar Smart Designer.

For example, in the [C1Schedule Smart Designer](#):



Clicking **Localize** opens the **Localize** dialog box:



The **Localize** dialog box allows you to localize the **Scheduler for WinForms** assembly (C1.Win.C1Schedule.2.dll) and save the localized resource in any project of your solution.

On the left of the **Localize** dialog box, there is a tree listing the localizable strings' IDs, and on the right are the strings themselves. The structure of the tree reflects the hierarchy of sub-classes in the Strings class. The right panel can show either all strings or just the strings belonging to the selected tree node.

The strings' list contains the following columns:








Column	Description
Name	The string's name (ID); this column repeats the selected tree node, and can be optionally turned off.
Default value	The default (English) value of the string.
Value	The string value for the currently selected culture (the column header displays that culture).

Below the list is the currently selected string's value, along with an optional description.

The status bar displays the project which will contain the localized resources and the name of the ComponentOne assembly which is currently being localized.

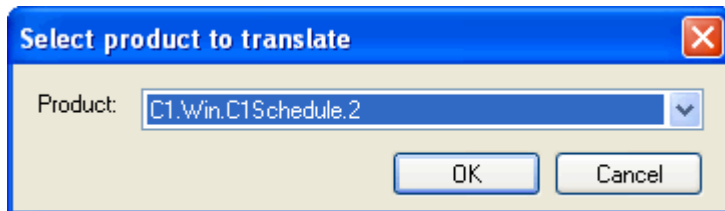
Localization Toolbar


The **Localize** dialog box contains the following toolbar menu buttons:

Button	Description
	Create new translation begins a new localization for a ComponentOne assembly.
 Open... ▾	Open opens an existing translation for a particular assembly.
	Save saves the current translation.
 Save as...	Save as saves the current translation and allows you to select the project in which to save the translation.
	Add culture adds a new culture.
	Delete culture removes a culture from the translations.
Culture Invariant Language (Invar ▾	Select culture selects the culture to display and edit.
	Options customizes the appearance and behavior of the localization window.

Create New Translation

Clicking the **Create new translation** button begins a new localization for a ComponentOne assembly. A dialog box opens for you to select the ComponentOne assembly to localize.

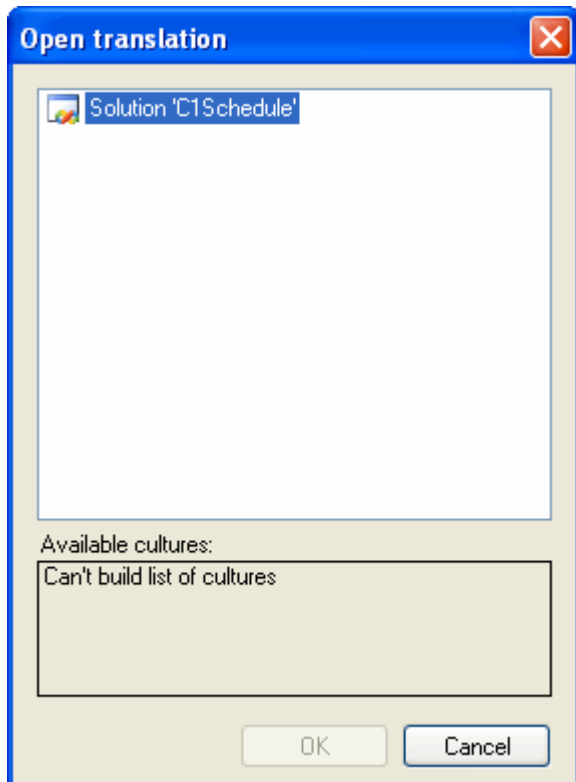


 **Note:** The assembly must be referenced in the currently open solution.

Open

Clicking the **Open** button opens an existing translation for a particular assembly. All translations that you create are stored as .resx files and are automatically added to the project that you select while saving the translation. Clicking this item shows a dialog box where a previously saved translation can be selected.

When you create a new solution it does not contain any translations, so initially that window would look like this:




After you have created and saved a translation, the **Available cultures** panel shows the list of cultures for which translations were created for the selected assembly.

Save

Clicking the **Save** button saves the current translation.

The translation is saved in the project shown in the status bar. When the translation is saved, a folder with the name **C1LocalizedResources** is created in the selected project (if it does not already exist), and the .resx files with translations are saved in that folder and added to the project. For each culture, a separate .resx file is created. These files are visible in the Solution Explorer window.

 **Note:** If your translation is only for the invariant culture, the .resx file does not contain a culture suffix.

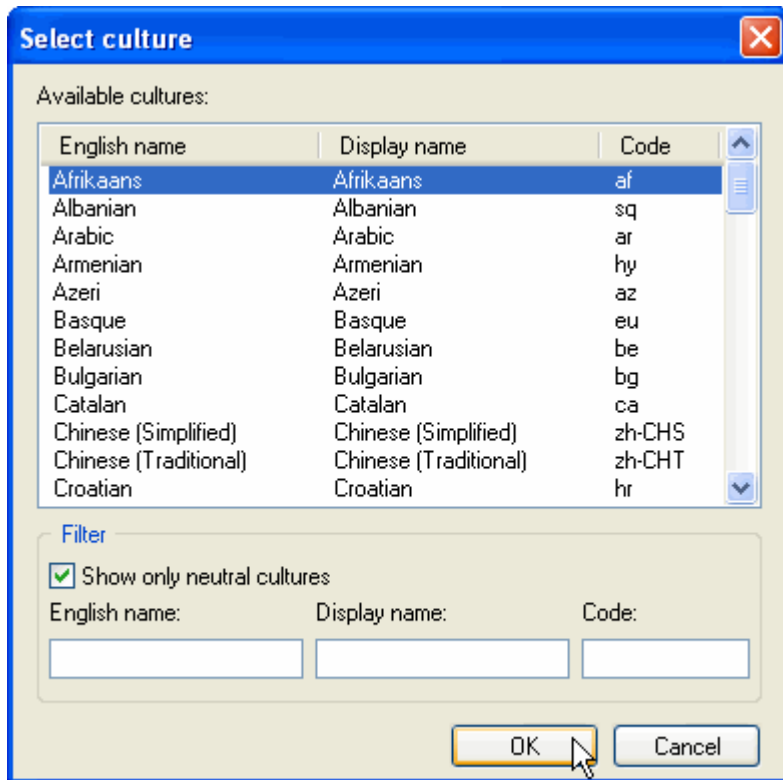
Save As

Clicking the **Save as** button saves the current translation and allows you to select the project in which to save the translation.

Add Culture

Clicking the **Add culture** button adds a new culture.

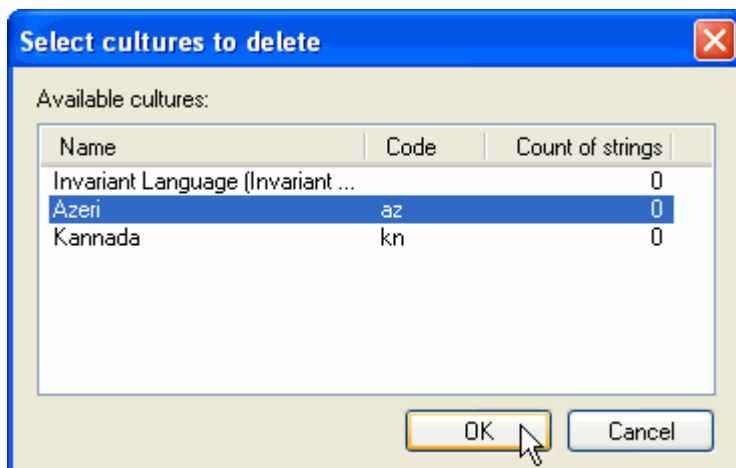
You can make translations for several cultures, and dynamically switch between them at run time. For each culture, a separate .resx file is created in the **C1LocalizedResources** folder. Clicking the **Add culture** button opens the **Select culture** dialog box that provides a list of available cultures:



Initially the list contains neutral cultures only. To show all cultures, uncheck the **Show only neutral cultures** checkbox. You can use the **English name**, **Display name**, and **Code** boxes to filter the list of shown cultures. After you have selected a culture, press the **OK** button to add it to the translations. The newly added culture will appear in the **Culture** drop-down in the toolbar and will become current in the window.

Delete Culture

Clicking the **Delete culture** button removes a culture from the translations. The **Select cultures to delete** dialog box provides the list of cultures existing in the translations:



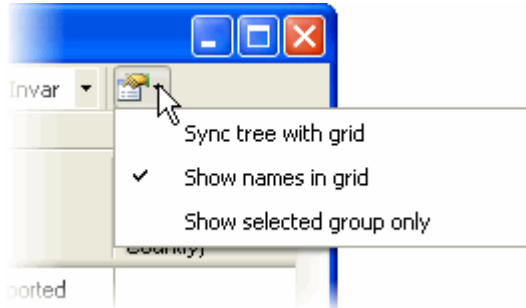
Selecting a culture and clicking **OK** removes it from the translations.

Select Culture

The **Culture** drop-down allows you to select the culture to display and edit.

Options

Clicking the **Options** button allows you to customize the appearance and behavior of the localization window.



The available localization options include:

Option	Description
Sync tree with grid	When this item is checked, selecting a string in the right panel list also selects that string in the tree on the left. By default this item is unchecked.
Show names in grid	When this item is checked, the <i>Name</i> column is shown in the right-hand panel, otherwise that column is hidden. By default this item is checked.
Show selected group only	When this item is checked, the list of strings on the right contains only the strings from the group currently selected in the tree on the left. By default this item is unchecked.

Setting the Current Culture

The **C1Schedule** control will use localization files automatically according to the culture selected in the application as long as you haven't moved files to another location or excluded files from the project. By default, the current culture is designated as **System.Threading.Thread.CurrentThread.CurrentCulture**. If you want to use a culture other than the current culture, you can set the desired culture in your application using the following code:

To write code in Visual Basic

Visual Basic

```
Public Sub New()
    ' Set desired culture, for example here the French (France) locale.
    System.Threading.Thread.CurrentThread.CurrentCulture = New
    System.Globalization.CultureInfo("fr-FR")

    ' This call is required by the Windows Form Designer.
    InitializeComponent()

    ' Add any initialization after the InitializeComponent() call.
End Sub
```

To write code in C#

C#

```
public Form1()
{
    // Set desired culture, for example here the French (France) locale.
    System.Threading.Thread.CurrentThread.CurrentCulture = new
System.Globalization.CultureInfo("fr-FR");

    // This call is required by the Windows Form Designer.
    InitializeComponent();

    // Add any initialization after the InitializeComponent() call.
}
```

Switching Resources at Run Time

Once you have created your localized resources, you can also easily switch between resources at run time by using the following code:

To write code in Visual Basic

Visual Basic

```
' This will switch to German locale.
System.Threading.Thread.CurrentThread.CurrentUICulture = New CultureInfo("de")
```

To write code in C#

C#

```
// This will switch to German locale.
System.Threading.Thread.CurrentThread.CurrentUICulture = new CultureInfo("de");
```

Grouping

Scheduler for WinForms supports visually grouping appointments. The [C1Schedule](#) control supports grouping by resources, contacts, categories and by the Owner property value. All C1Schedule views and visual styles support grouping and contain UI elements for navigation between groups.

The following classes in **Schedule for WinForms** support grouping:

- [ScheduleGroupItem](#): Holds all the data required for displaying individual UI part for the single resource, category or contact when the GroupBy property is set, or default UI otherwise.
- [ScheduleGroupItemCollection](#): Represents a collection of the ScheduleGroupItem objects.
- The following properties were added in the C1Schedule class to support grouping:
- [GroupBy](#): Gets or sets the string value determining the type of grouping. This property can be equal to one for the next supported keywords:
 - An empty string - no grouping.
 - "Owner" - grouping by [Owner](#) property value.
 - "Category" - grouping by [Categories](#) property value.
 - "Contact" - grouping by [Links](#) property value.
 - "Resource" - grouping by [Resources](#) property value.

The default value is an empty string.

- **ShowGroupNavigation**: Gets or sets the Boolean value determining whether the C1Schedule control should display group navigation buttons. The default value is **True**. This property only makes sense when the GroupBy property is set.
- **ShowEmptyGroupItem**: Gets or sets the Boolean value determining whether the C1Schedule control should display an empty group item. This property only makes sense when the GroupBy property is set. If this property is set to **True**, the C1Schedule control will display a group item without assigned owner (resource, contact or category). This group item will contain all appointments, which don't fall into other group items. The default value is **False**.
- **EmptyGroupName**: Gets or sets a String value used as default value for the **Name** property. The default value is "Calendar".
- **GroupItems**: Gets a collection of all available ScheduleGroupItem objects for the currently set type of grouping.
- **VisibleGroupItems**: Gets a collection of currently visible ScheduleGroupItem objects.
- **IsGrouped**: Gets a Boolean value determining whether grouping has been set for the C1Schedule control.
- **SelectedGroupItem**: Gets the selected ScheduleGroupItem object or returns null if the selection is empty.
- **GroupPageSize**: Gets or sets the Integer value determining the maximum number of the ScheduleGroupItem objects displayed in UI at the same time. Increasing this value might affect performance. The default value is 2.
- The following method in the C1Schedule class supports grouping:
- **NavigateToScheduleGroup**: Navigates the C1Schedule control back or forth to another ScheduleGroupItem object by the specified increment.

Showing Working Time From One Day to Another

If **CalendarInfo.EndDayTime** property value is less than **CalendarInfo.StartDayTime** property value, working time will be represented as 2 time intervals - from day start to **CalendarInfo.EndDayTime** and from **CalendarInfo.StartDayTime** to day end.

The **C1Schedule** control honors this setting when **ShowWorkTimeOnly** property is **True**. Note, when the **ShowWorkTimeOnly** property is true, C1Schedule always shows the full hour. As a result, some free time might be included into the view.

Customizing Scheduler for WinForms' Appearance

The following topics explain how to configure elements of **Scheduler for WinForms'** display, such as Visual Styles, color schemes, headers, borders, and date formats:

Visual Styles

Scheduler for WinForms has thirteen built-in Visual Styles, as well as the ability to create custom Visual Styles. You can also save and load previously saved Visual Styles.

C1Calendar Visual Style Settings

Appearance settings for the [C1Calendar](#) control, including the Visual Style, can be set through the **Visual Style** dialog box of the [C1Calendar Smart Designer](#). In the **Visual Style** dialog box, you can set properties for the entire C1Calendar control, headers, days, and selected days.

C1Schedule Visual Style Settings

Appearance settings for the [C1Schedule](#) control, including the Visual Style, can be set through the **Visual Style** dialog box of the [C1Schedule Smart Designer](#). In the **Visual Style** dialog box, you can set properties for the entire C1Schedule control, title, appointments, time ruler, day view, week view, month view, and month view headers.

Visual Styles

The [C1Schedule](#) and the [C1Calendar](#) controls support Visual Styles. In addition to Visual Styles, Scheduler for WinForms also supports **C1ThemeController** so you can choose from many predefined themes using [Themes for WinForms](#).

There are several predefined Visual Styles:

Visual Style	Description
Aero	Uses a light yellow palette.
Office 2007 Black	Mimics the Microsoft Office 2007 Black theme.
Office 2007 Blue	Mimics the Microsoft Office 2007 Blue theme.
Office 2007 Silver	Mimics the Microsoft Office 2007 Silver theme.
Office 2010 Black	Mimics the Microsoft Office 2010 Black theme.
Office 2010 Blue	Mimics the Microsoft Office 2010 Blue theme.
Office 2010 Silver	Mimics the Microsoft Office 2010 Silver theme.
Royale	Mimics the Microsoft Windows Media Center Royale theme.
System	Uses the current desktop settings.
Windows XP Blue	Mimics the Microsoft Windows XP Blue theme.
Windows XP Olive	Mimics the Microsoft Windows XP Olive Green theme.
Windows XP Silver	Mimics the Microsoft Window XP Silver theme.

Visual Style	Description
Yahoo	Uses a grey palette.

Visual Styles can be set in the smart designers, smart tags, or in code. For details on how to set a predefined Visual Styles, see [Setting a Predefined Visual Style](#).

You cannot edit predefined Visual Styles, but you can create new Visual Styles. At design-time, Visual Styles can be saved to an XML file or previously saved Visual Styles can be loaded.

Note: Clicking the **Save** button will save all of the custom Visual Styles added to the Visual Style dialog box to an XML file. Clicking the **Load** button will load all of the custom Visual Styles in the XML file.

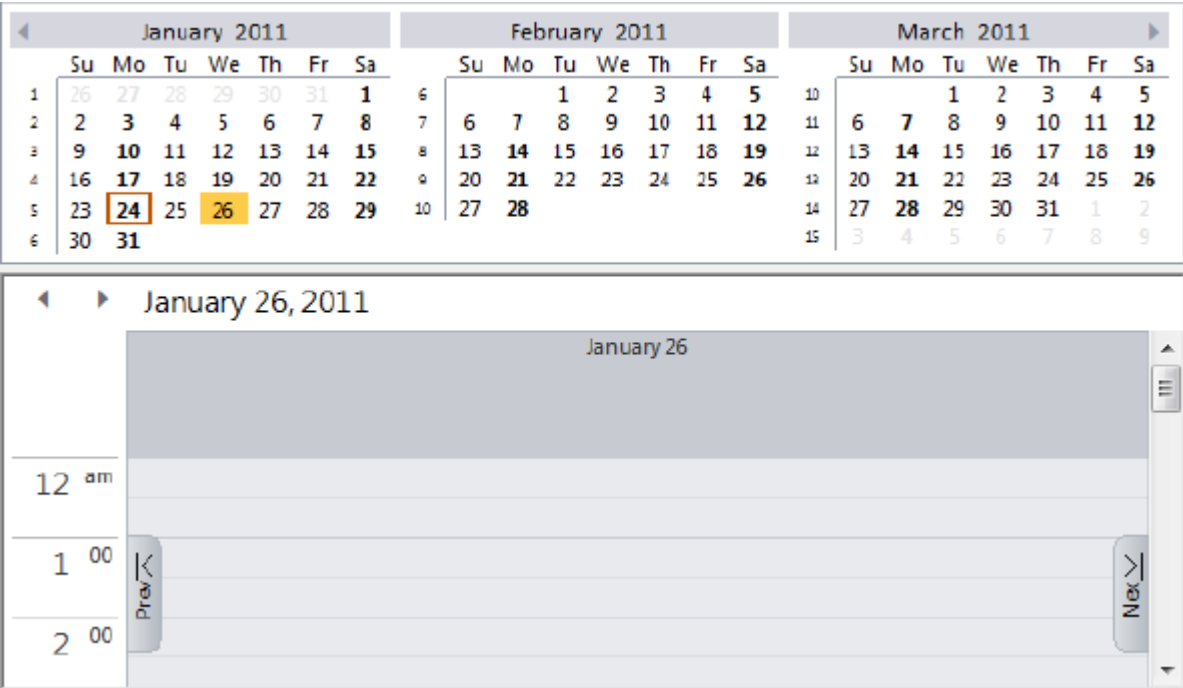
Aero Visual Style

The following image displays the Aero Visual Style:



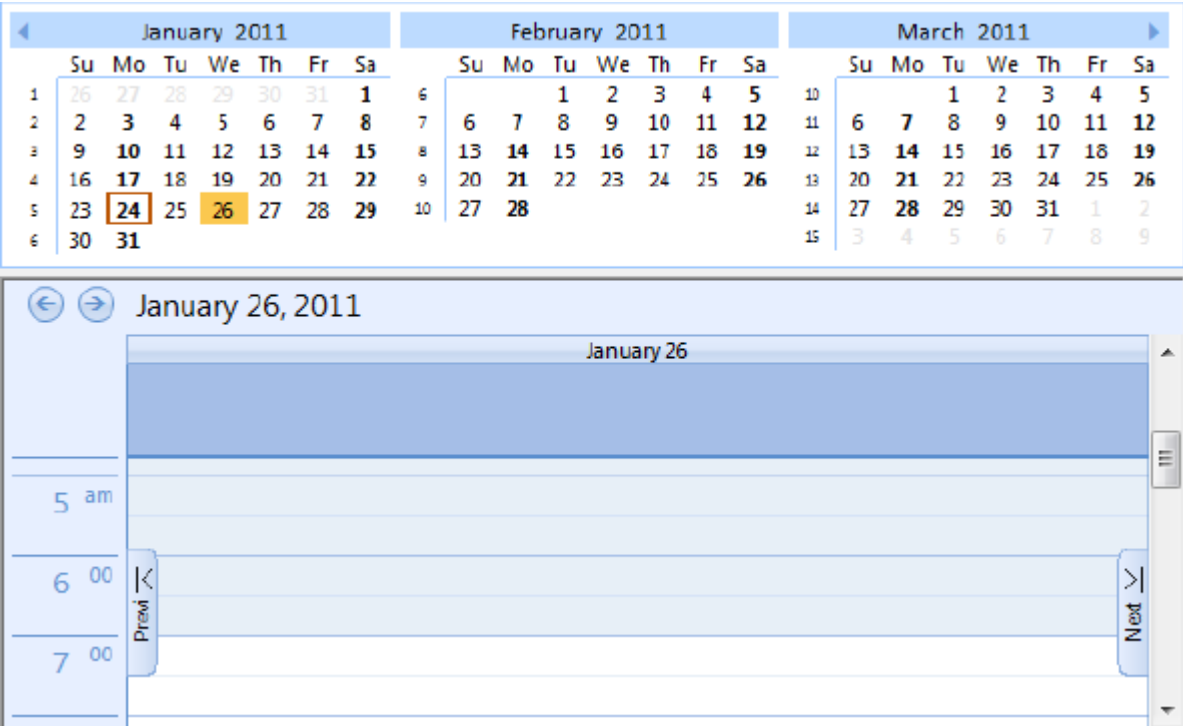
Office 2007 Black Visual Style

The following image displays the Office 2007 Black Visual Style:



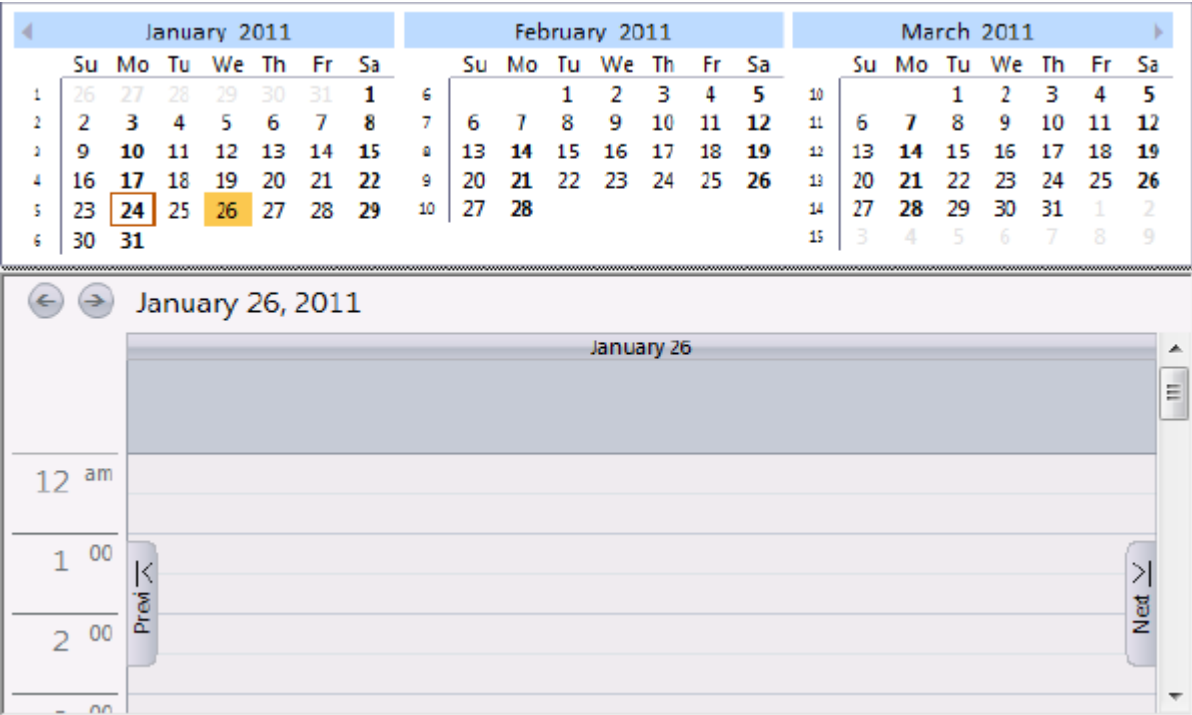
Office 2007 Blue Visual Style

The following image displays the Office 2007 Blue Visual Style:



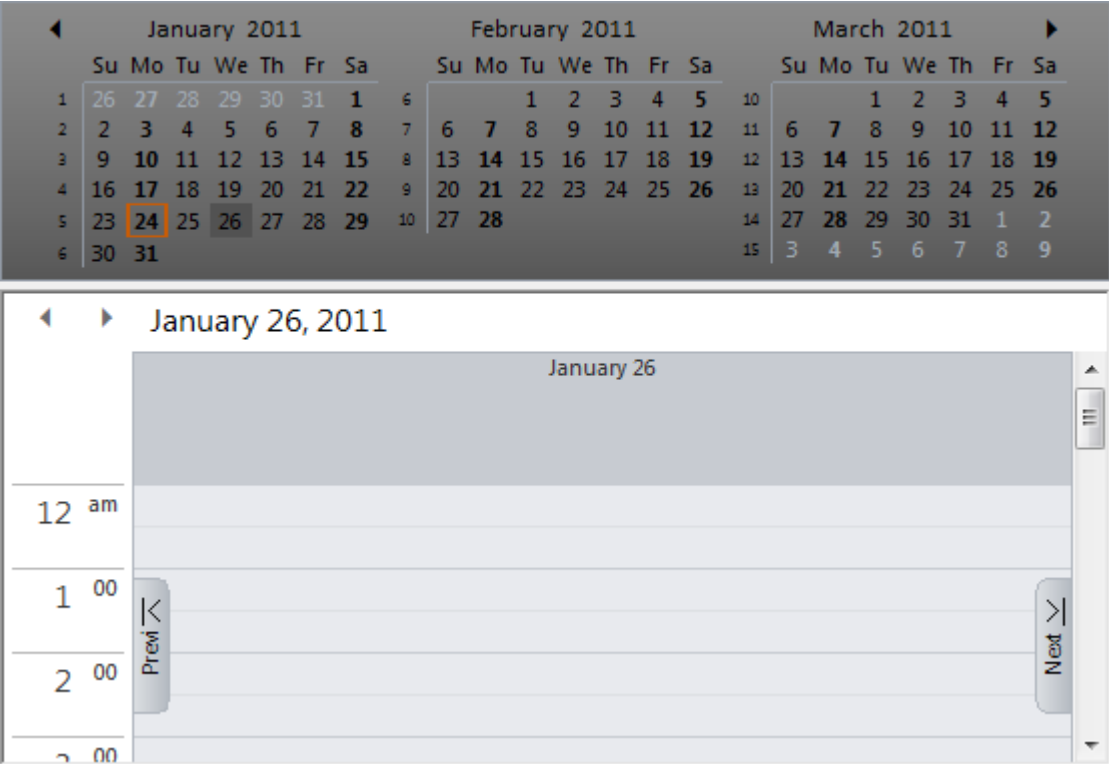
Office 2007 Silver Visual Style

The following image displays the Office 2007 Silver Visual Style:



Office 2010 Black Visual Style

The following image displays the Office 2010 Black Visual Style:



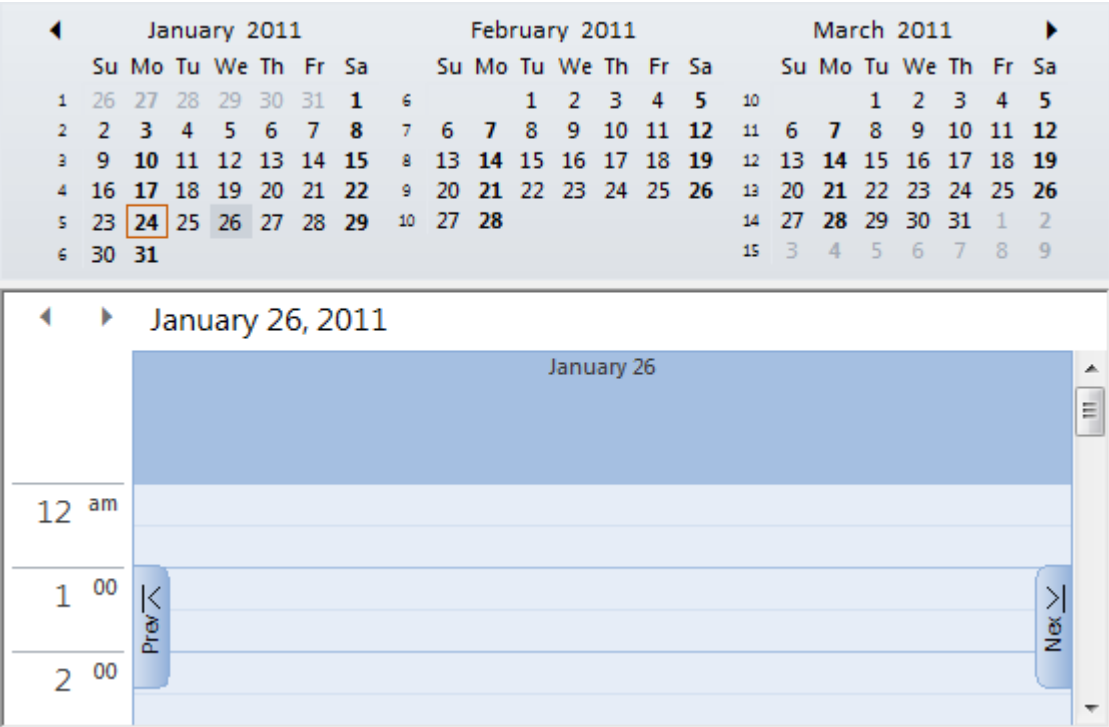
Office 2010 Blue Visual Style

The following image displays the Office 2010 Blue Visual Style:



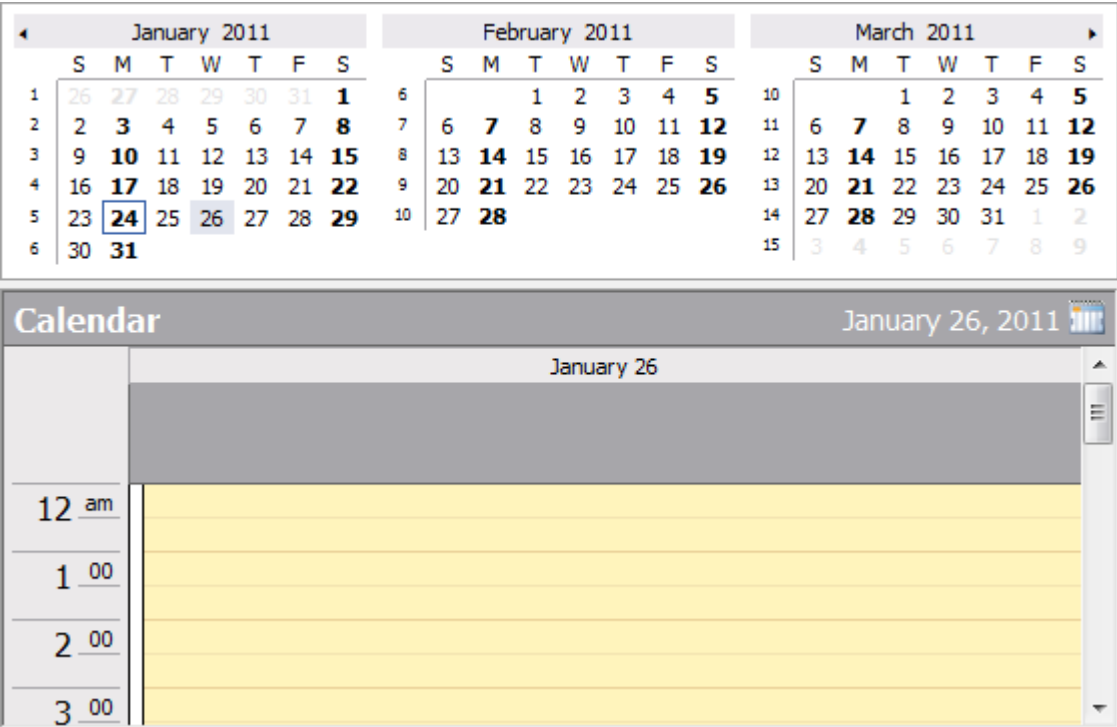
Office 2010 Silver Visual Style

The following image displays the Office 2010 Silver Visual Style:



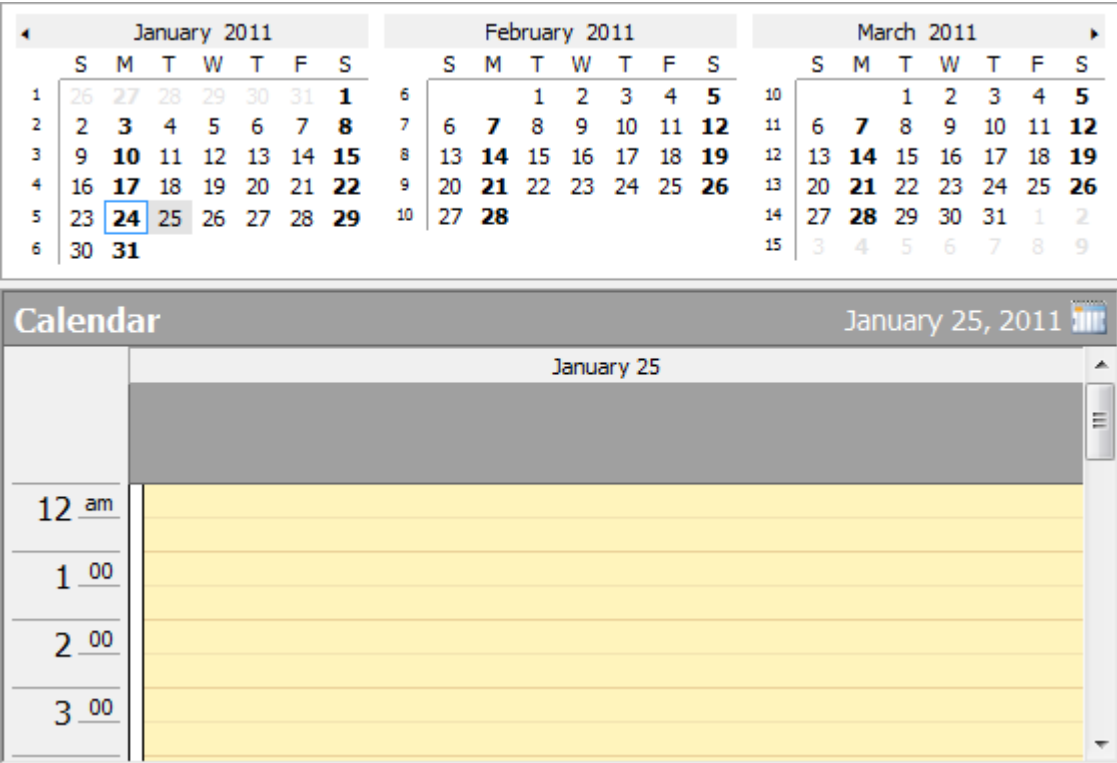
Royale Visual Style

The following image displays the Royale Media Center Visual Style:



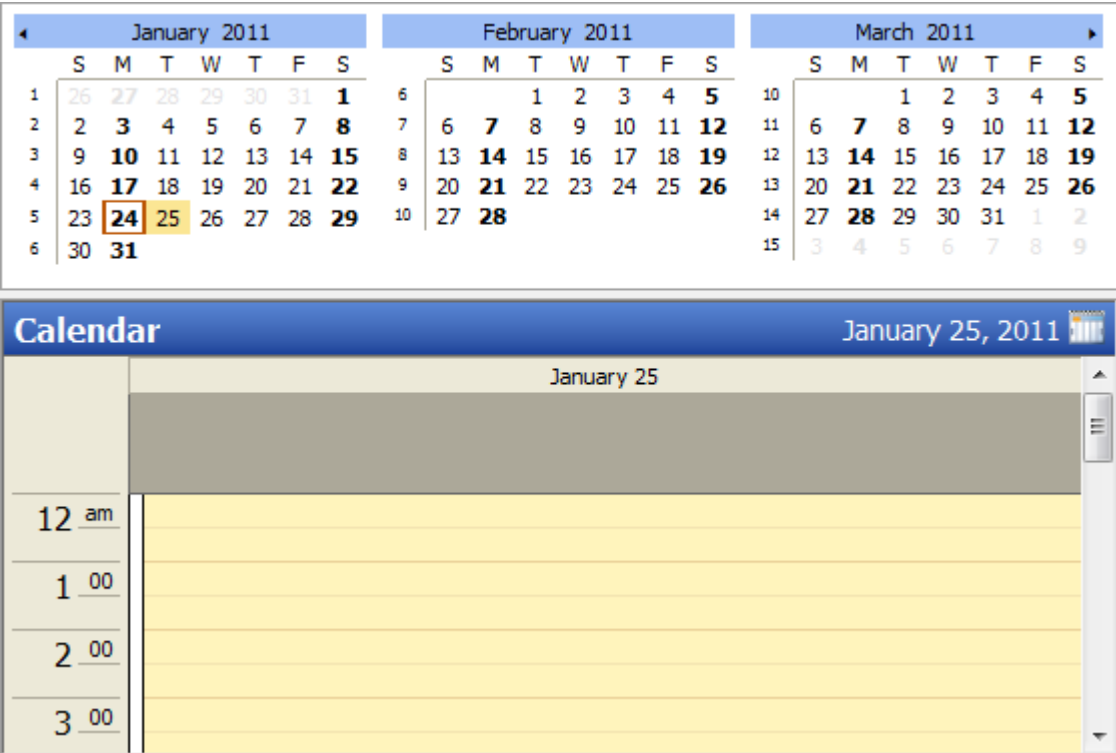
System Visual Style

The following image displays the current desktop settings:



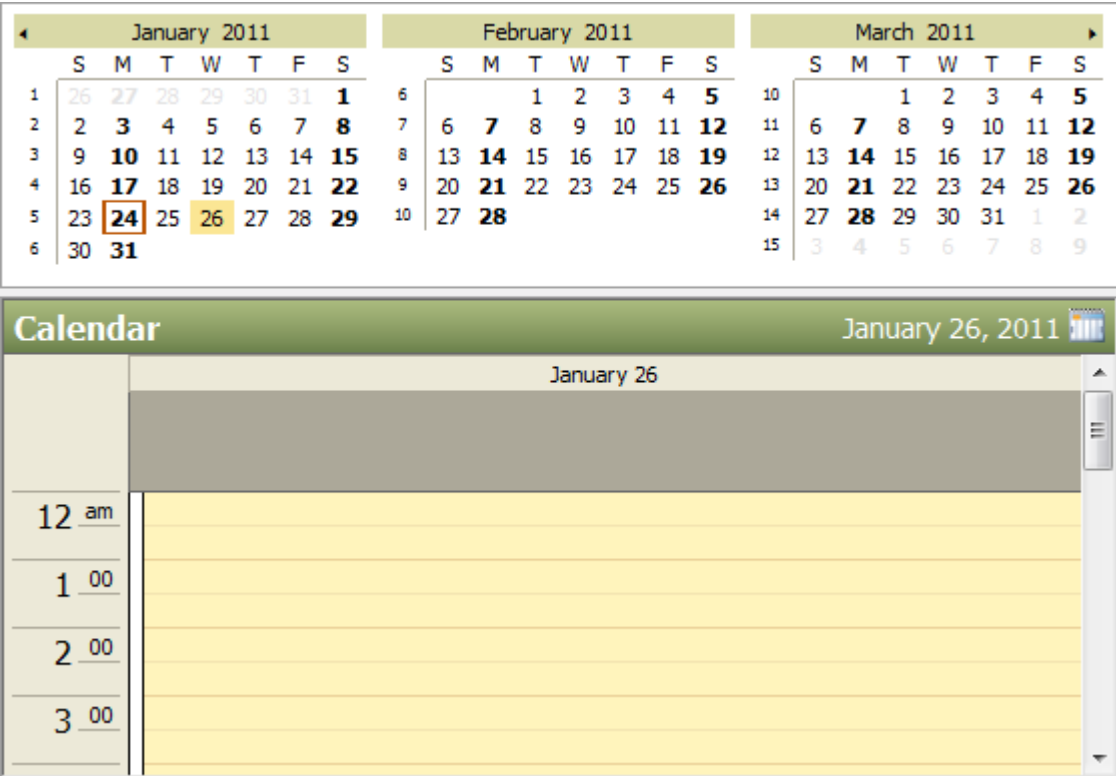
Windows XP Blue Visual Style

The following image displays the Windows XP Blue Visual Style:



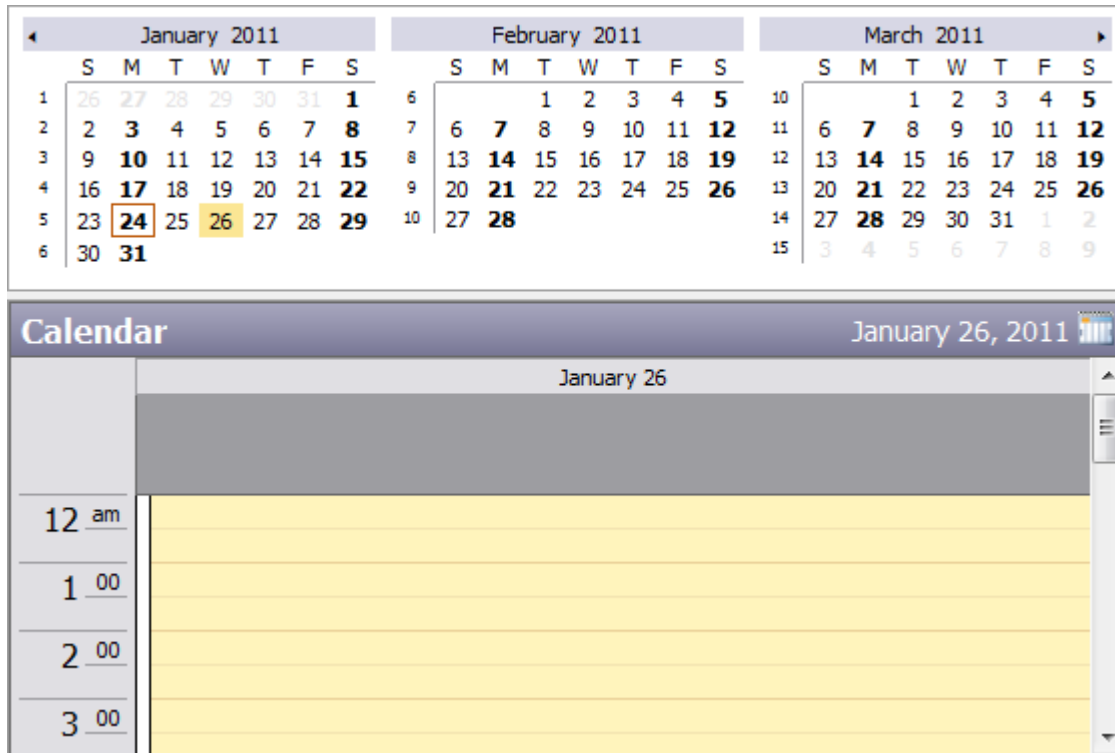
Windows XP Olive Visual Style

The following image displays the Windows XP Olive Visual Style:



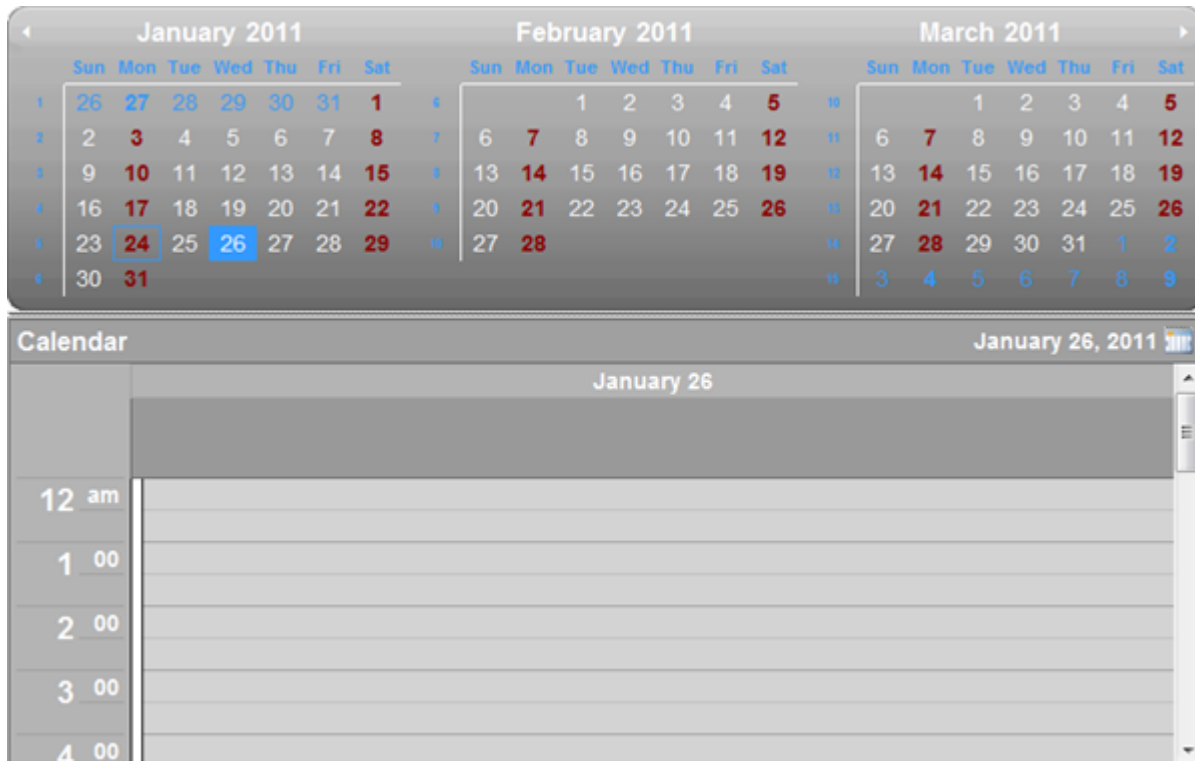
Windows XP Silver Visual Style

The following image displays the Windows XP Silver Visual Style:



Yahoo Visual Style

The following image displays the Yahoo Visual Style:



Please note that Visual Styles - Aero, Royal, System, Yahoo, and Windows XP are marked with obsolete attribute. You are advised to use other modern visual styles available. You can also use themes such as Office 2010, Office 2013, etc. included into C1Themes assembly, see [Themes for WinForms](#) for more information.

Setting a Predefined Visual Style

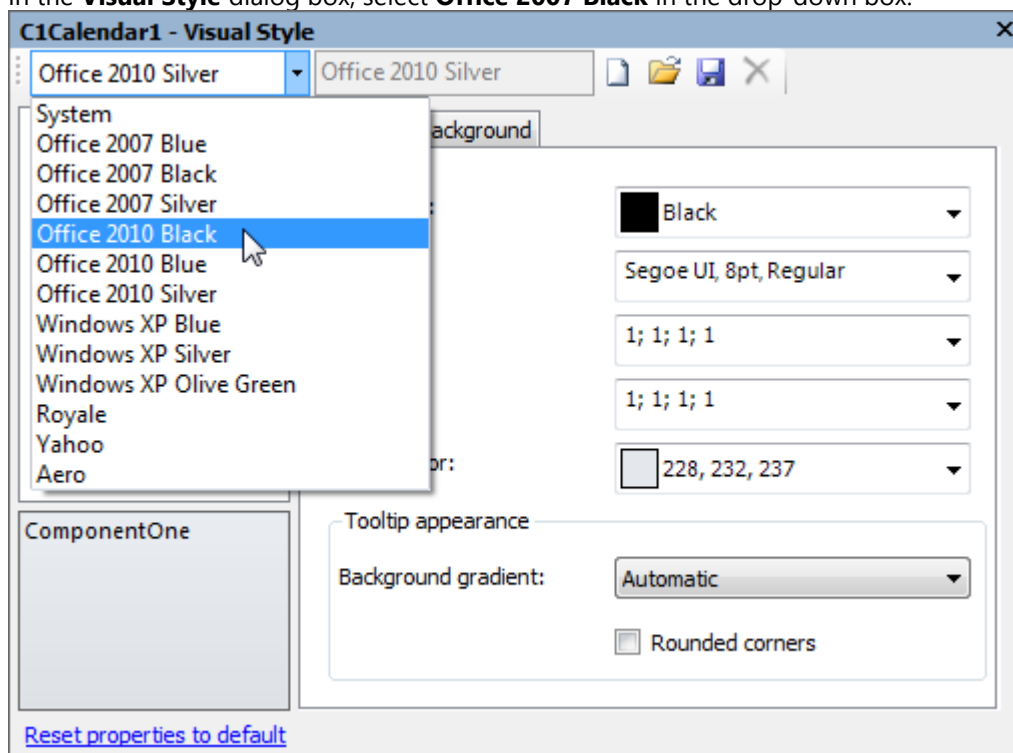
You can define a Visual Style by setting the **VisualStyle** property for both the [C1Calendar](#) and [C1Schedule](#) controls either in the smart designers, tasks menu, or in code. For descriptions of the available Visual Styles, see [Visual Styles](#).

Please note that Visual Styles - Aero, Royal, System, Yahoo, and Windows XP are marked with obsolete attribute. You are advised to use other modern visual styles available. You can also use themes such as Office 2010, Office 2013, etc. included into C1Themes assembly, see [Themes for WinForms](#) for more information.

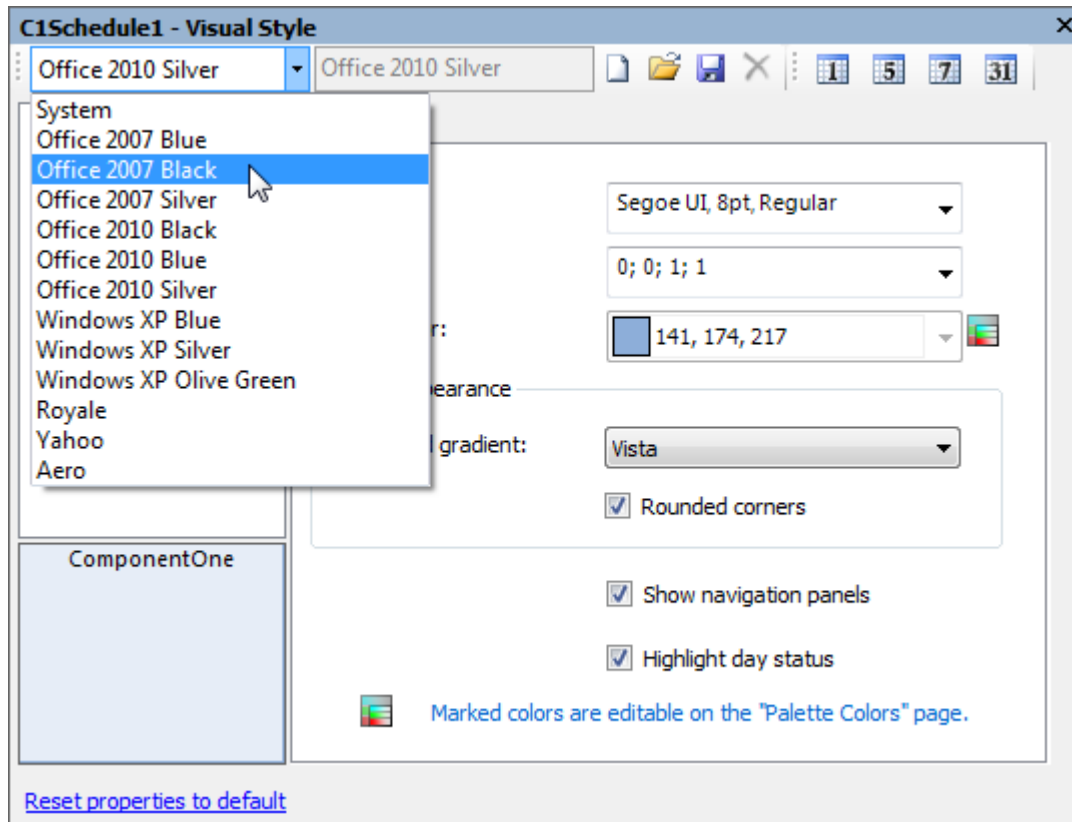
In the Smart Designer

To set the **VisualStyle** property using the smart designers, complete the following steps:

1. On the **C1Calendar Smart Designer**, click the **Visual Style** button. For more information on accessing the **C1Calendar Smart Designer**, see [C1Calendar Smart Designer](#).
2. In the **Visual Style** dialog box, select **Office 2007 Black** in the drop-down box.



3. Close the dialog box.
4. On the **C1Schedule Smart Designer**, click the **Visual Style** button. For more information on accessing the **C1Schedule Smart Designer**, see [C1Schedule Smart Designer](#).
5. In the **Visual Style** dialog box, select **Office 2007 Black** in the drop-down box.

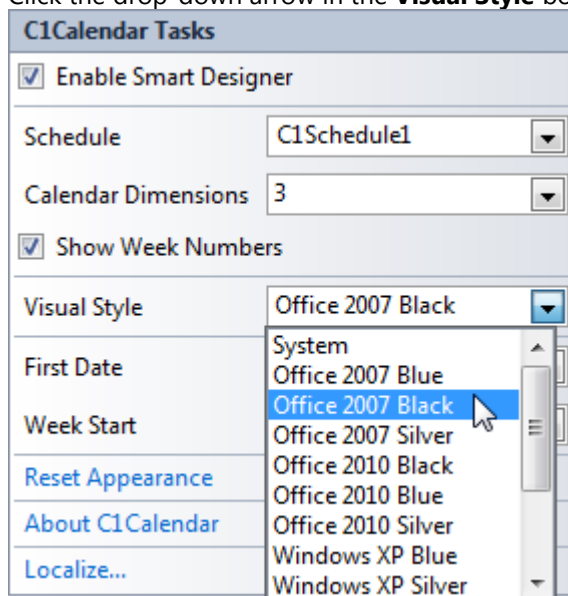


6. Close the dialog box.

In the Tasks menu

To set the **VisualStyle** property using the Tasks menu, complete the following steps:

1. Open the **C1Calendar Tasks** menu. For more information on accessing the **C1Calendar Tasks** menu, see [C1Calendar Tasks Menu](#).
2. Click the drop-down arrow in the **Visual Style** box, and select **Office 2007 Black**.



3. Close the **C1Calendar Tasks** menu.
4. Open the **C1Schedule Tasks** menu. For information on accessing the **C1Schedule Tasks** menu, see [C1Schedule Tasks Menu](#).

5. Click the drop-down arrow in the **Visual Style** box, and select **Office 2007 Black**.
6. Close the **C1Schedule Tasks** menu.

In Code

Add the following code to the **Form_Load** event to set the visual styles for the C1Calendar and C1Schedule controls to **Office2007Black**:

To write code in Visual Basic

Visual Basic

```
Me.C1Calendar1.VisualStudio = C1.Win.C1Schedule.UI.VisualStudio.Office2007Black
Me.C1Schedule1.VisualStudio = C1.Win.C1Schedule.UI.VisualStudio.Office2007Black
```

To write code in C#

C#

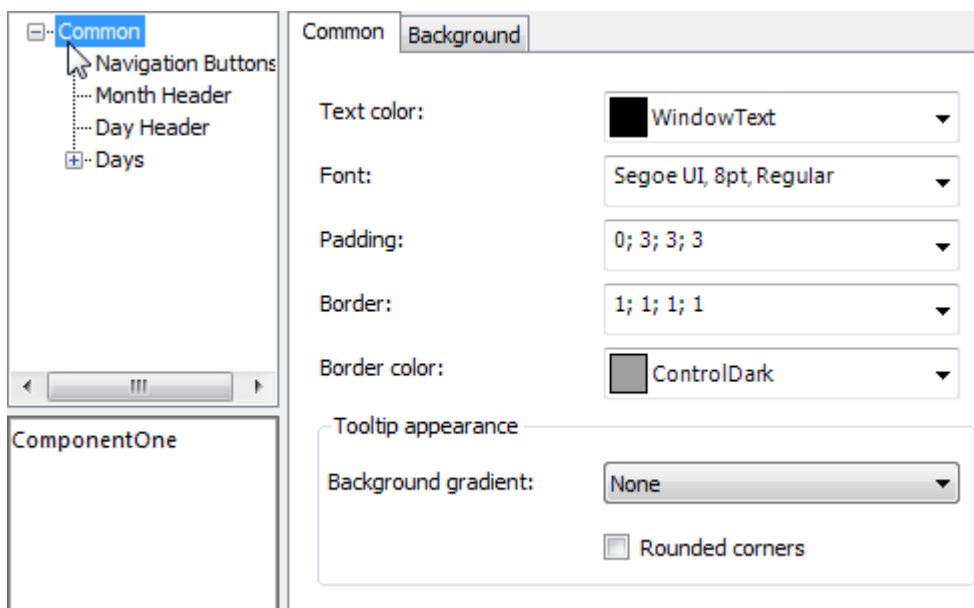
```
this.c1Calendar1.VisualStudio = C1.Win.C1Schedule.UI.VisualStudio.Office2007Black;
this.c1Schedule1.VisualStudio = C1.Win.C1Schedule.UI.VisualStudio.Office2007Black;
```

C1Calendar Visual Style Settings

Appearance settings for the **C1Calendar** control can be set through the **Visual Style** dialog box of the C1Calendar Smart Designer. For more information on accessing the **Visual Style** dialog box, see **C1Calendar Smart Designer**.

Common Node

Through the **Common** node of the **Visual Style** dialog box, you can set properties common to all of the **C1Calendar** control.

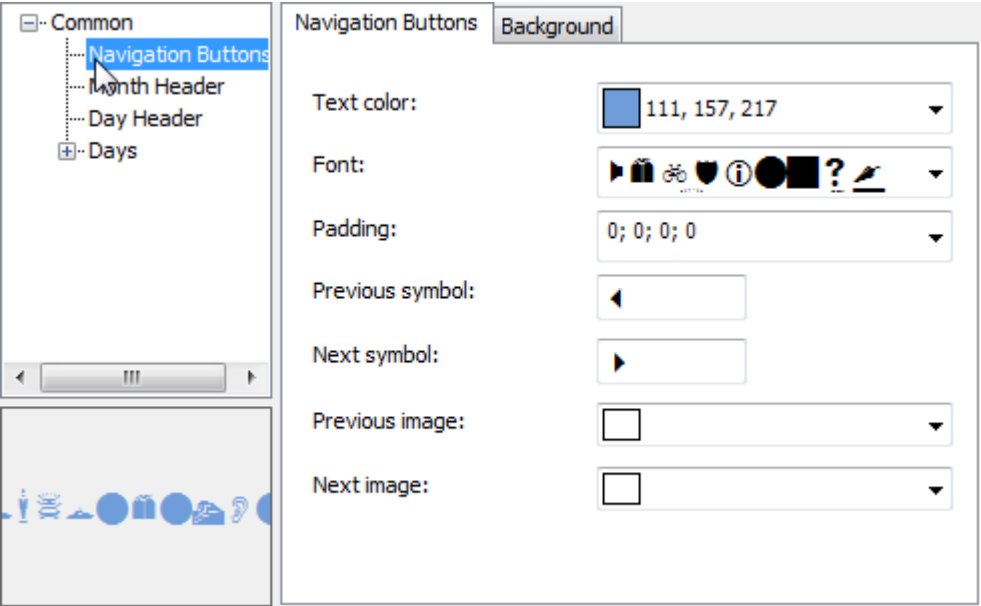


The following properties are available through the **Common** node:

Tab	Property	Description
Common	Text color	Sets the color for all text in the C1Calendar control.
	Font	Sets the font for all text in the C1Calendar control.
	Padding	Sets the padding around the calendars in the C1Calendar control.
	Border	Sets the border around the calendars in the C1Calendar control.
	Border color	Sets the color for the border around the calendars in the C1Calendar control.
	Background Gradient	Sets the ToolTip's background gradient color. Options include None , Automatic , Silver , Blue , Gold , Olive , and Vista .
	Rounded corners	Gets or sets a Boolean value determining whether to show ToolTips with rounded corners via the ToolTipRoundedCorners property.
Background	Background color	Sets the background color for the C1Calendar control.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Navigation Buttons Node

Through the **Navigation Buttons** node of the **Visual Style** dialog box, you can set properties for the navigation buttons.

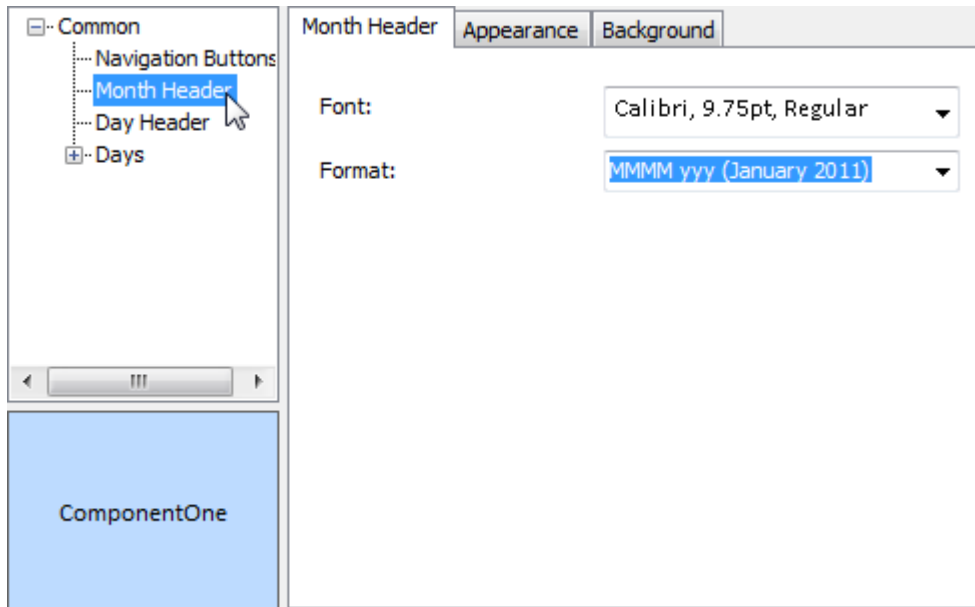


The following properties are available through the **Navigation Buttons** node:

Tab	Property	Description
Navigation Buttons	Text color	Sets the text color for the navigation buttons.
	Font	Sets the font for the Previous and Next buttons.
	Padding	Sets the padding around the Previous and Next buttons.
	Previous symbol	Sets the symbol to appear for the Previous button, based on the Font for the navigation buttons.
	Next symbol	Sets the symbol to appear for the Next button, based on the Font for the navigation buttons.
	Previous image	Sets the image to appear for the Previous button.
	Next image	Sets the image to appear for the Next button.
Background	Background color	Sets the background color for the Previous and Next buttons.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Month Header Node

Through the **Month Header** node of the **Visual Style** dialog box, you can set properties for the month header.



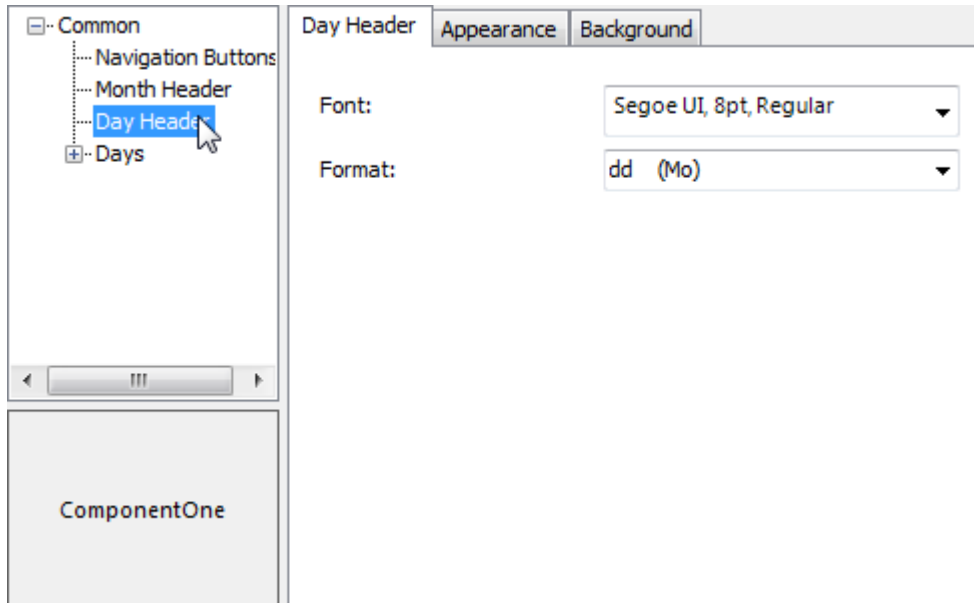
The following properties are available through the **Month Header** node:

Tab	Property	Description
Month header	Font	Sets the font for the month header.
	Format	Sets the format of the month header text. Available formats include: <ul style="list-style-type: none"> • MMMM yyyy - Displays the full name of the month and the year. For example, February 2007. • MMM yyyy - Displays the abbreviated name of the month and the year. For example, Feb 2007. • yyy MMM - Displays the year and the abbreviated name of the month. For example, 2007 Feb. • yyy MMMM - Displays the year and the full name of the month. For example, 2007 February.
Appearance	Text color	Sets the color for the text in the month header.
	Padding	Sets the padding around the month header text.
	Border	Sets the border width for the month header.
	Border color	Sets the border color for the month header.
	Alignment	Sets the alignment for the month header text.
Background	Background color	Sets the background color for the month header.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.

Tab	Property	Description
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Day Header Node

Through the **Day Header** node of the **Visual Style** dialog box, you can set properties for the month header.



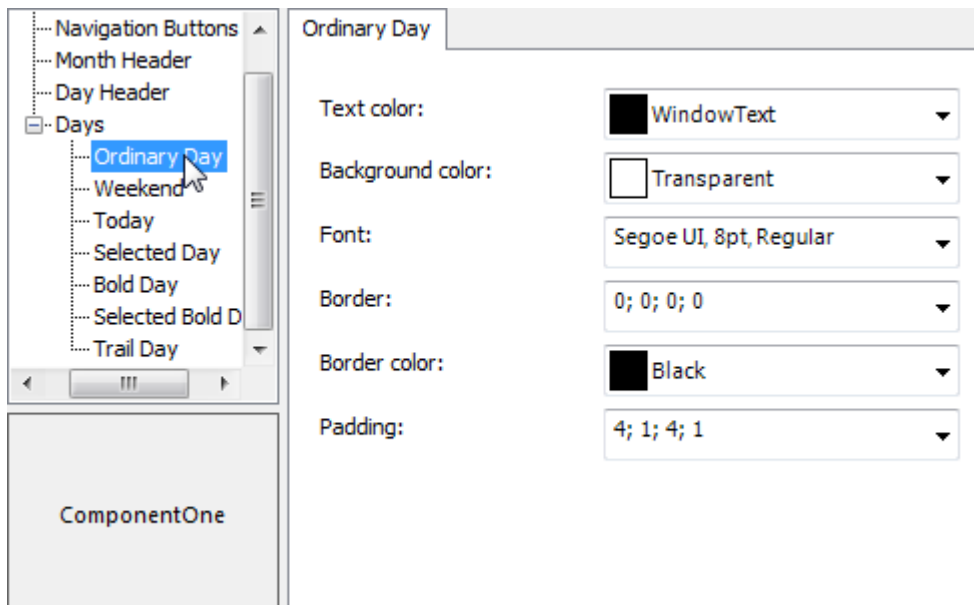
The following properties are available through the **Day Header** node:

Tab	Property	Description
Day header	Font	Sets the font for the day header.
	Format	Sets the format of the day header text. Available formats include: * The following formats are available in the Japanese version: <ul style="list-style-type: none"> • ddd - Displays the 3-letter abbreviated name for each day. For example, Mon. • dddd - Displays the full name of the day. For example, Monday. • d - Displays the 1-letter abbreviated name for each day. For example, M. • dd - Displays the 2-letter abbreviated name for each day. For example, Mo. • ddd - Displays the 3-letter abbreviated name for each day. For example, Mon.
Appearance	Text color	Sets the color for the text in the day header.
	Padding	Sets the padding around the day header text.
	Border	Sets the border width for the day header.

Tab	Property	Description
Background	Border color	Sets the border color for the day header.
	Alignment	Sets the alignment for the day header text.
	Background	Sets the background color for the day header.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Days Node

Through the **Days** node of the **Visual Style** dialog box, you can set properties for ordinary days, today, selected days, bold days, selected bold days, and trailing days.



The following properties are available through the **Days** node:

Tab	Property	Description
Ordinary Day	Text color	Sets the text color for the dates in the calendar.
	Background color	Sets the background color for dates in the calendar.
	Font	Sets the font for the dates in the calendar.
	Border	Sets the border with for the dates in the calendar.
	Border color	Sets the border color for the dates in the calendar.

Tab	Property	Description
	Padding	Sets the padding around the dates in the calendar.
Weekend	Text color	Sets the text color for the dates in the calendar.
	Background color	Sets the background color for dates in the calendar.
	Font	Sets the font for the dates in the calendar.
	Border	Sets the border width for the dates in the calendar.
	Border color	Sets the border color for the dates in the calendar.
	Padding	Sets the padding around the dates in the calendar.
Today	Text color	Sets the text color for today's date.
	Background color	Sets the background color for today's date.
	Font	Sets the font for today's date.
	Border	Sets the border width for today's date.
	Border color	Sets the border color for today's date.
	Padding	Sets the padding around today's date.
Selected Day	Text color	Sets the text color for the selected date in the calendar.
	Background	Sets the background color for the selected date in the calendar.
	Font	Sets the font for the selected date in the calendar.
	Border	Sets the border width for the selected date in the calendar.
	Border color	Sets the border color for the selected date in the calendar.
	Padding	Sets the padding around the selected date in the calendar.
Selected Bold Day	Text color	Sets the text color for the selected date in the calendar when the date is bold.
	Background	Sets the background color for the selected date in the calendar when the date is bold.
	Font	Sets the font for the selected date in the calendar when the date is bold.
	Border	Sets the border width for the selected date in the calendar when the date is bold.
	Border color	Sets the border color for the selected date in the calendar when the date is bold.
	Padding	Sets the padding around the selected date in the calendar when the date is bold.
Bold Day	Text color	Sets the text color for days that are bold in the calendar.
	Background	Sets the background color for days that are bold in the

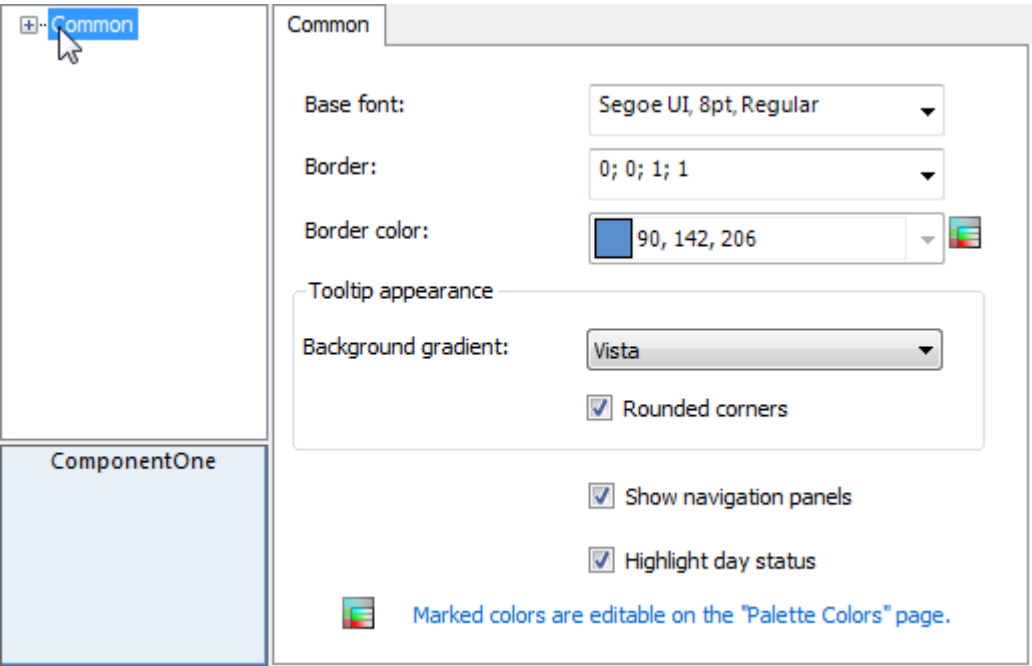
Tab	Property	Description
		calendar.
	Font	Sets the font for days that are bold in the calendar.
	Border	Sets the border width for days that are bold in the calendar.
	Border color	Sets the border color for days that are bold in the calendar.
	Padding	Sets the padding around days that are bold in the calendar.
Trail Day	Text color	Sets the text color for trailing days in the calendar.
	Background	Sets the background color for trailing days in the calendar.
	Font	Sets the font for trailing days in the calendar.
	Border	Sets the border width for trailing days in the calendar.
	Border color	Sets the border color for trailing days in the calendar.
	Padding	Sets the padding around trailing days in the calendar.

C1Schedule Visual Style Settings

Appearance settings for the [C1Schedule](#) control can be set through the **Visual Style** dialog box of the C1Schedule Smart Designer. For more information on accessing the **Visual Style** dialog box, see [C1Schedule Smart Designer](#).

Common Node

Through the **Common** node of the **Visual Style** dialog box, you can set the font style for the entire [C1Schedule](#) control.

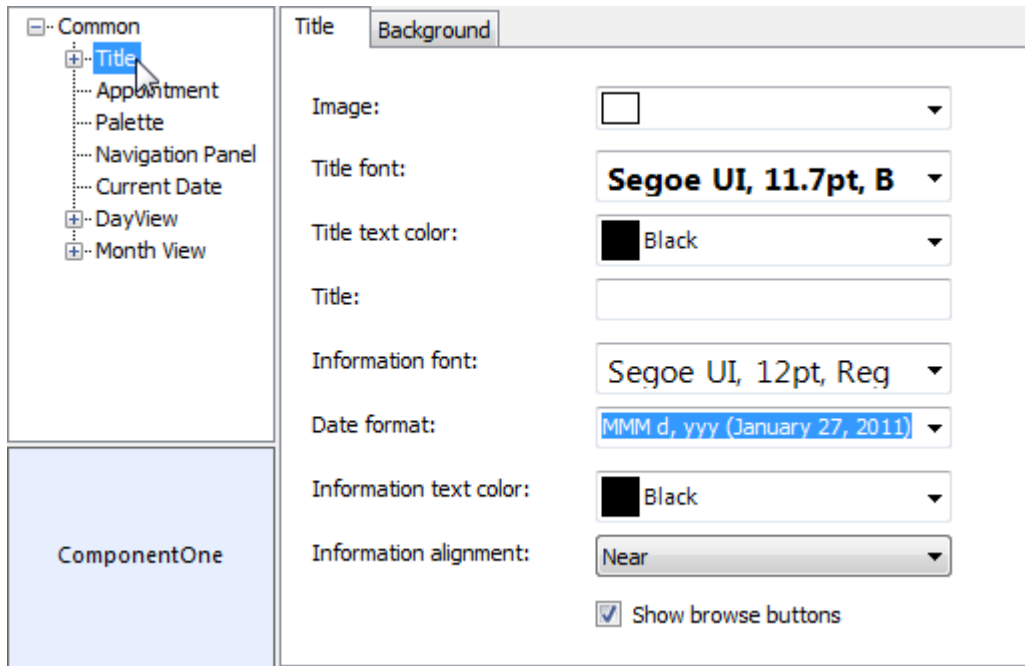


The following properties are available through the **Common** node:

Tab	Property	Description
Common	Base font	Sets the font for the entire C1Schedule control.
	Border	Sets the border width.
	Border color	Sets the border color.
	Background gradient	Sets the background gradient style for the ToolTip.
	Rounded corners	Displays the ToolTip with rounded corners when checked.
	Show navigation panels	Displays the navigation panels when checked.
	Highlight day status	Highlights availability status when checked.

Title Node

Through the **Title** node of the **Visual Style** dialog box, you can set the font and text properties for the title.



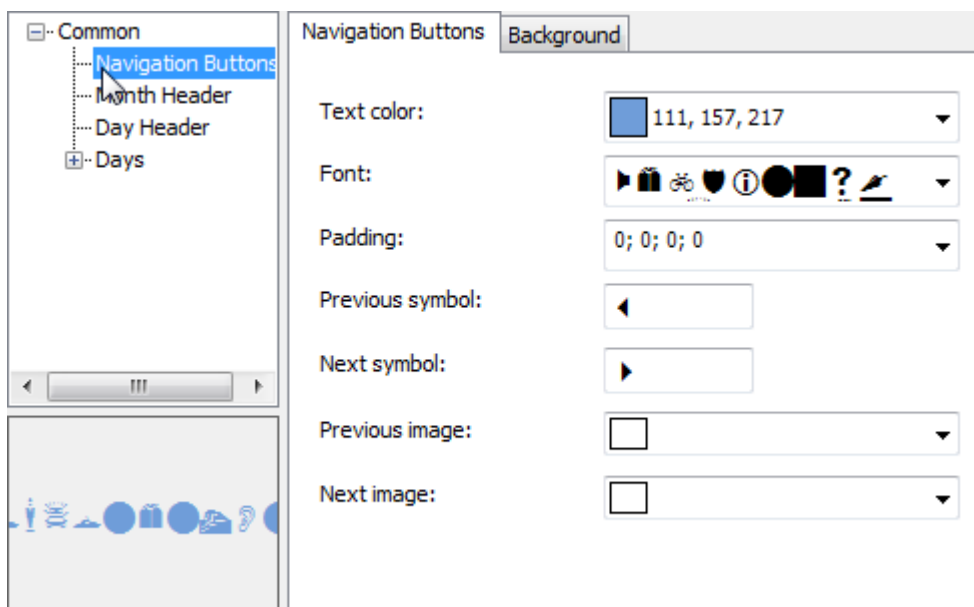
The following properties are available through the **Title** node:

Group	Property	Description
Title	Image	Sets the image for the title.
	Title font	Sets the font for the title.
	Title text color	Sets the text color for title bar.
	Title	Sets the text that will appear in the title bar.
	Information font	Sets the font for the date in the title bar.
	Date format	Sets the format of the date in the title bar. Available formats include: <ul style="list-style-type: none"> • d - Displays the numeric date value. For example, 1/9/2007. • MMMM d, yyyy - Displays the full name of the month, the numeric date value, and the year. For example, January 9, 2007. • yyyy, MMMM d - Displays the year, the full name of the month and the numeric date value. For example, 2007, January 9. • yyyy, d MMMM - Displays the year, the numeric date value and the full name of the month. For example, 2007, 9 January. • d MMMM, yyyy - Displays the date value, the full name of the month and the year. For example, 9 January, 2007.
	Information text color	Sets the text color for the date in the title bar.
	Information alignment	Sets the alignment for the date in the title bar.
	Show browse buttons	Displays the browse buttons when checked.

Group	Property	Description
Background	Background color	Sets the background color for the title.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Navigation Buttons Node

Through the **Navigation Buttons** node of the **Visual Style** dialog box, you can set properties for the navigation buttons.



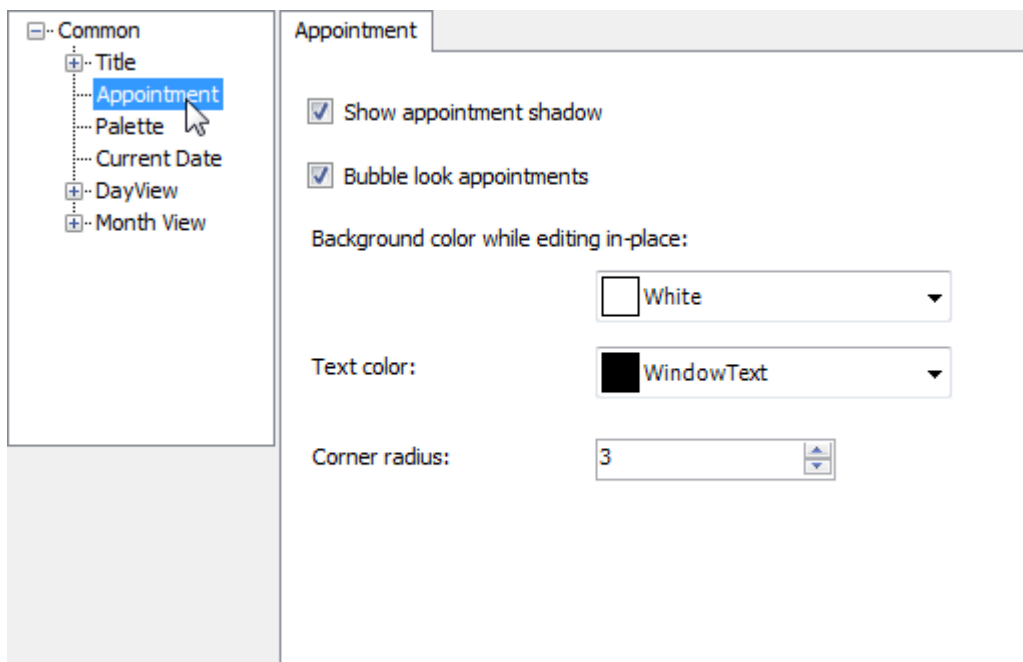
The following properties are available through the **Navigation Buttons** node:

Tab	Property	Description
Navigation Buttons	Text color	Sets the symbol color for the Previous and Next buttons.
	Font	Sets the font for the Previous and Next buttons.
	Padding	Sets the padding around the Previous and Next buttons.
	Previous symbol	Sets the symbol to appear for the Previous button, based on the Font for the navigation buttons.
	Next symbol	Sets the symbol to appear for the Next button, based on the Font for the navigation buttons.

Tab	Property	Description
	Previous image	Sets the image to appear for the Previous button.
	Next image	Sets the image to appear for the Next button.
	Border	Sets the border width for the navigation buttons.
	Border color	Sets the border color for the navigation buttons.
Background	Background color	Sets the background color for the Previous and Next buttons.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Appointment Node

Through the **Appointment** node of the **Visual Style** dialog box, you can set the appointment properties.



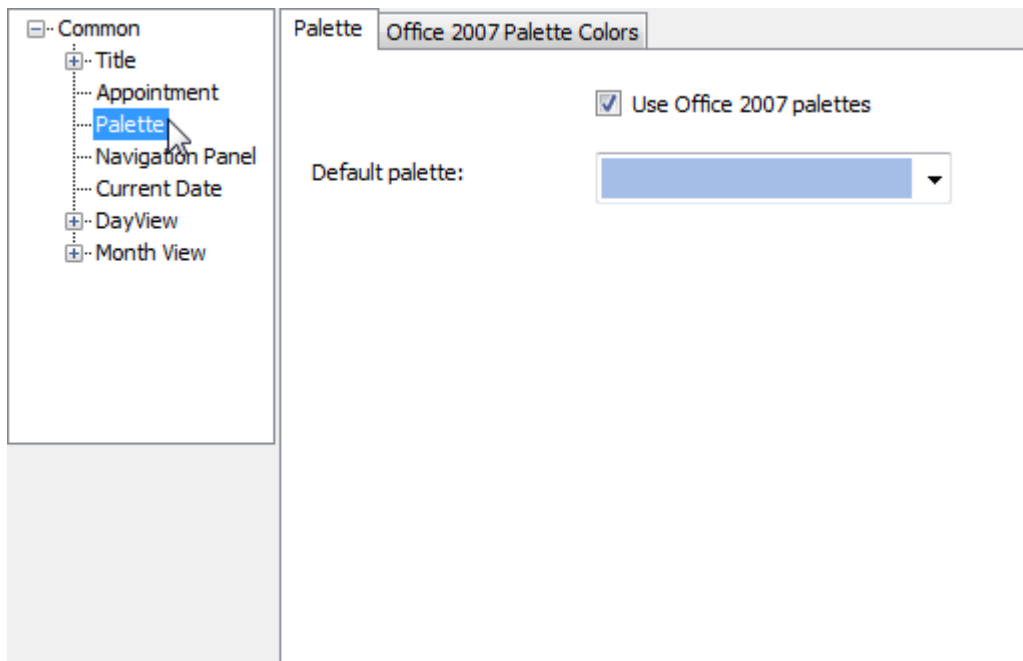
The following properties are available through the **Appointment** node:

Tab	Property	Description
Appointment	Show appointment shadow	Displays the appointment shadow when checked.
	Bubble look	Displays a bubble-look for appointments when checked.

Tab	Property	Description
	appointments	
	Background color while editing in-place	Sets the background color for appointments at in-place editing.
	Text color	Sets the text color for appointments.
	Corner radius	Sets the corner radius for bubble-look appointments.

Palette Node

Through the **Palette** node of the **Visual Style** dialog box, you can set the palette properties.



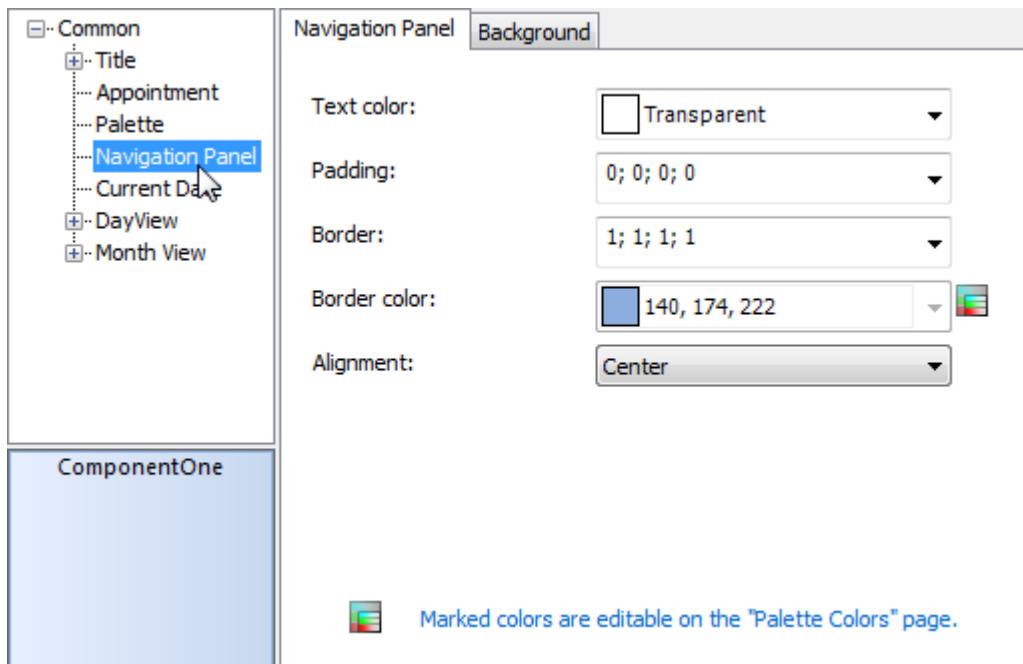
The following properties are available through the **Palette** node:

Tab	Property	Description
Palette	Use Office 2007 palettes	Uses the Office 2007 palettes when checked.
	Default palette	Sets the palette colors using a built-in color palette.
Office 2007 Palette Colors	Alternate month day, all-day area, work and free hours dark border	The border color for alternate month day, all-day area, work hours and free hours dark borders.
	Free hour, selected day, work hour light border	The border color for free hour, selected day, and work hour light borders.
	Free hour light border	The border color for free hour horizontal light borders.
	Day header border	The border color for day header borders.
	Day header background	The background color for day headers.
	Day header gradient color	The gradient color for day headers.

Tab	Property	Description
	Border	The border color.
	Selected slot, selected all-day area	The background color for selected slot and selected all-day area.

Navigation Panel Node

Through the **Navigation Panel** node of the **Visual Style** dialog box, you can set properties for the navigation panel.



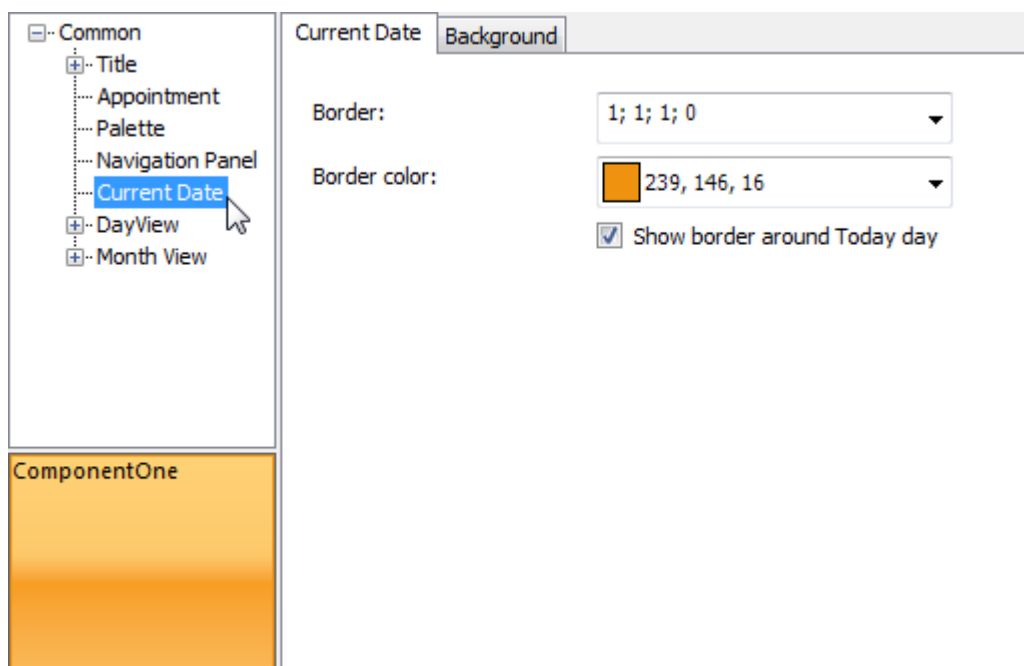
The following properties are available through the **Navigation Panel** node:

Tab	Property	Description
Navigation Panel	Text color	Sets the text color for text in the navigation panel.
	Padding	Sets the amount of padding used in the navigation panel.
	Border	Sets the border width for each side of the navigation panel.
	Border color	Sets the border color for navigation panel.
	Alignment	Sets the alignment option (NotSet, Near, Far, Center, or Spread) for the navigation panel.
Background	Background color	Sets the background color for current date.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the

Tab	Property	Description
		background gradient.
	Image	Sets the background image.

Current Date Node

Through the **Current Date** node of the **Visual Style** dialog box, you can set the current date properties.

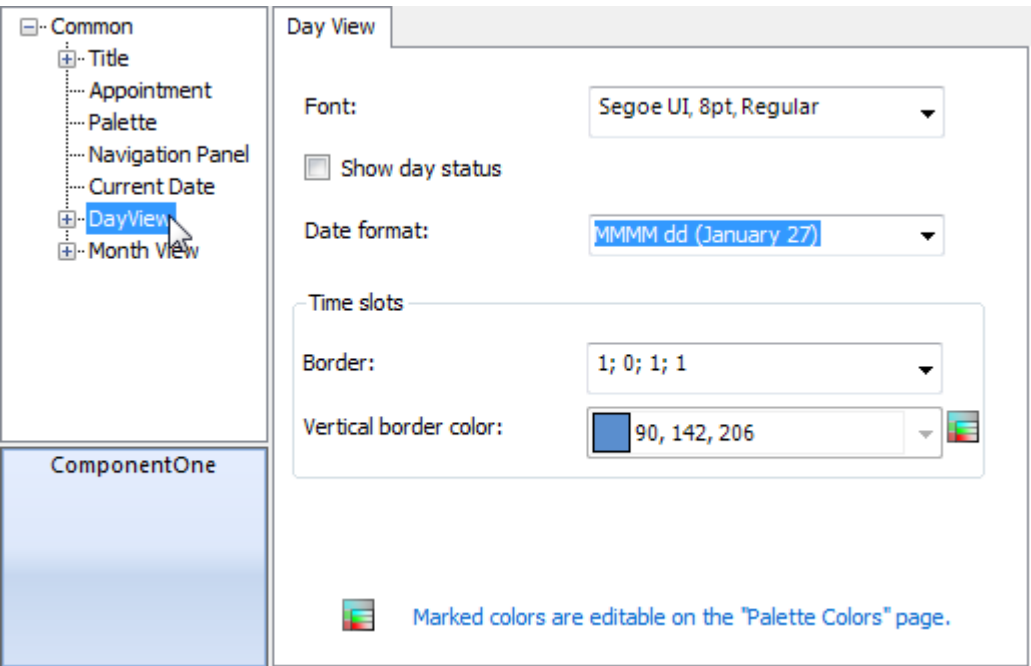


The following properties are available through the **Current Date** node:

Tab	Property	Description
Current Date	Border	Sets the border for the current date.
	Border color	Sets the border color for current date.
	Show border around Today day	Displays an orange border around the today date (as in Office 2007) when checked.
Background	Background color	Sets the background color for current date.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Day View Node

Through the **Day View** node of the **Visual Style** dialog box, you can set the day view properties.



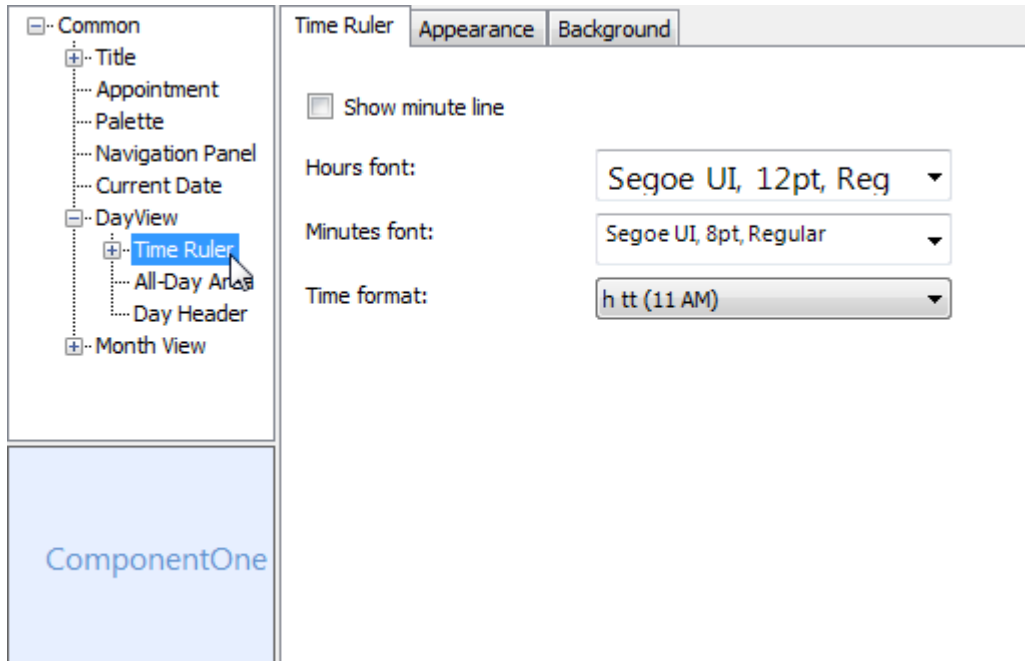
The following properties are available through the **Day View** node:

Group	Property	Description
Day View	Font	Sets the font for the day view.
	Show day status	Displays the status bar when checked.
	Date format	<p>Sets the format for the date. Available formats include:* The following formats are available in the Japanese version:</p> <ul style="list-style-type: none">• d - Displays the numeric date value. For example, 2007/02/01.• MMMM d dddd - Displays the month, measured as a number between 1 and 12, the current day of the month, measured as a number between 1 and 31, inclusive, and the full name of the day. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 2 1 Thursday.• MMMM d - Displays the month, measured as a number between 1 and 12, and the current day of the month, measured ad a number between 1 and 31, inclusive. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 2 1.• MMMM dd - Displays the month, measured as a number between 1 and 12, and the current day of the month, measured ad a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For

Group	Property	Description
		<p>example, 2 01.</p> <ul style="list-style-type: none"> • d ddd ♦ Displays the current day of the month, measured as a number between 1 and 31, inclusive, and the full name of the day. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 1 Thursday. • d - Displays the numeric date value. For example, 2/1/2007. • dddd, MMMM dd - Displays the full name of the day and month, and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, Thursday, February 01. • MMMM d - Displays the full name of the month and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, February 1. • MMMM dd - Displays the full name of the month and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, February 01. • dd MMMM - Displays the current day of the month, measured as a number between 1 and 31, inclusive, and the full name of the month. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, 01 February. • d MMMM - Displays the current day of the month, measured as a number between 1 and 31m inclusive, and the full name of the month. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 1 February. • ddd dd - Displays the abbreviated day name and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, Thu 01. • dd ddd - Displays the current day of the month, measured as a number between 1 and 31, inclusive, and Displays the abbreviated day name. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, 01 Thu.
	Border	Sets the border width for the day view.
	Vertical border color	Sets the vertical border color for the day view.

Time Ruler Node

Through the **Time Ruler** node of the **Visual Style** dialog box, you can set the font and border properties for the time ruler.



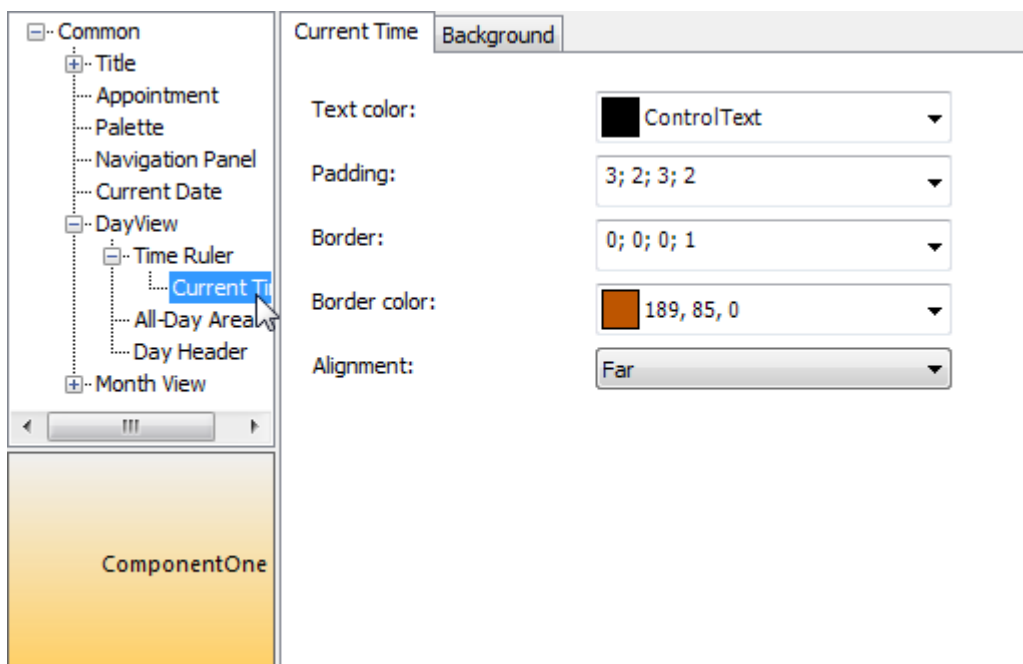
The following properties are available through the **Time Ruler** node:

Tab	Property	Description
Time Ruler	Show minute line	Displays the minute line when checked.
	Hours font	Sets the font for the hours.
	Minutes font	Sets the font for the minutes.
	Time format	Sets the format for the time. Available formats include: <ul style="list-style-type: none"> • T - Displays the hour, minute, and second range, as well as the A.M./P.M. designator. For example, 8:00:00 AM. • t - Displays the hour and minute range. For example, 8:00. • h mm - Displays the hour in the range 0-23 and the minute in the range 0-59. If the minute is a single digit (0-9), it is formatted with a preceding 0 (01-09). For example, 8 00. • h tt - Displays the hour in the range 1-12 and the A.M./P.M. designator. For example, 8 AM.
Appearance	Text color	Sets the text color for the time ruler.
	Padding	Sets the padding around the time text.
	Border	Sets the border width for the time ruler.
	Border color	Sets the border color for the time ruler.

Tab	Property	Description
	Alignment	Sets the alignment for the time ruler text.
Background	Background color	Sets the background color for the time ruler.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Current Time Node

Through the **Current Time** node of the **Visual Style** dialog box, you can set the font and border properties for the current time.



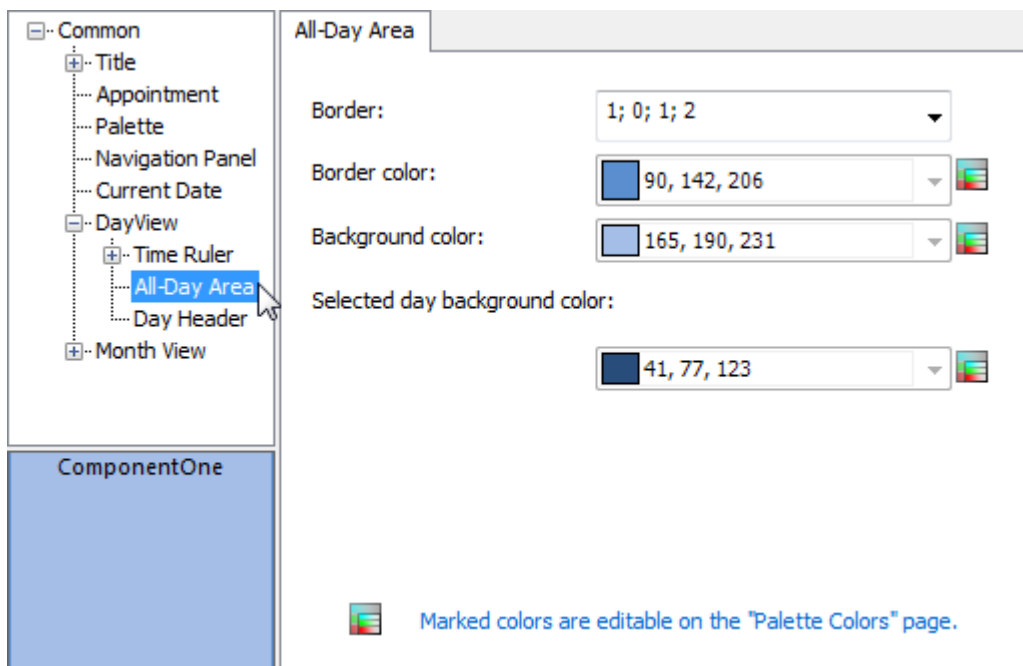
The following properties are available through the **Current Time** node:

Tab	Property	Description
Current Time	Text color	Sets the text color for the current time.
	Padding	Sets the padding around the current time.
	Border	Sets the border width for the current time.
	Border color	Sets the border color for the current time.
	Alignment	Sets the alignment for the current time text.

Tab	Property	Description
Background	Background color	Sets the background color for the current time.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

All-Day Area Node

Through the **All-Day Area** node of the **Visual Style** dialog box, you can set properties for the all-day area.

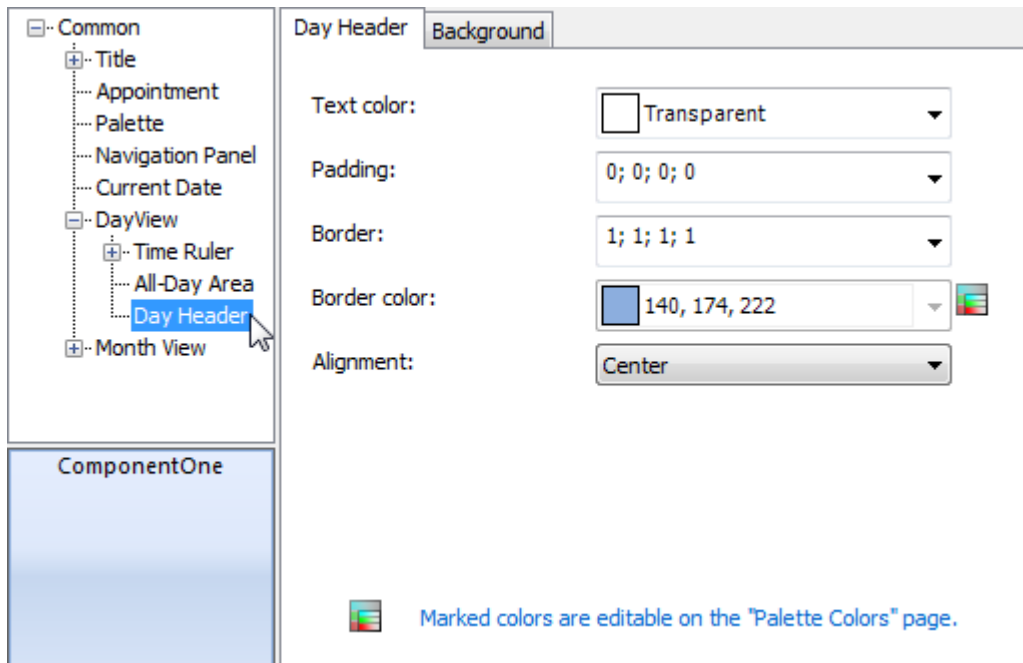


The following properties are available through the **All-Day Area** node:

Tab	Property	Description
All-Day Area	Border	Sets the border width for the all-day area.
	Border color	Sets the border color for the all-day area.
	Background color	Sets the background color for the all-day area.
	Selected day background color	Sets the background color for the selected day in the all-day area.

Day Header Node (Day View)

Through the **Day Header** node of the **Visual Style** dialog box, you can set properties for the day header.

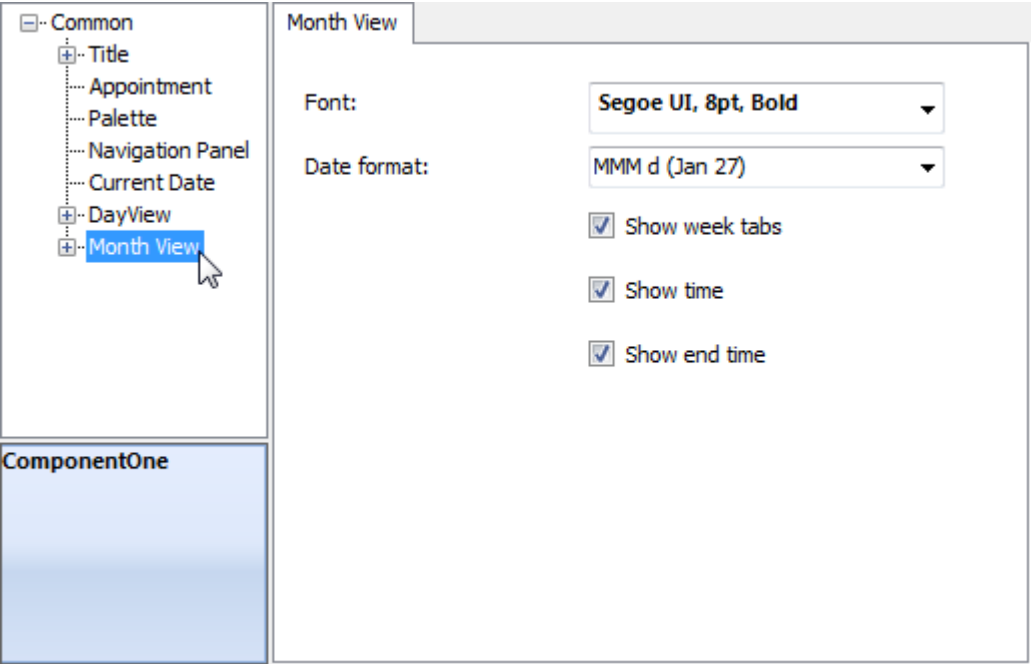


The following properties are available through the **Day Header** node:

Tab	Property	Description
Day Header	Text color	Sets the text color for day header.
	Padding	Sets the padding around the day header.
	Border	Sets the border width for the day header.
	Border color	Sets the border color for the day header.
	Alignment	Sets the alignment for the day header text.
Background	Background color	Sets the background color for the day header.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Month View Node

Through the **Month View** node of the **Visual Style** dialog box, you can set properties for the month view.



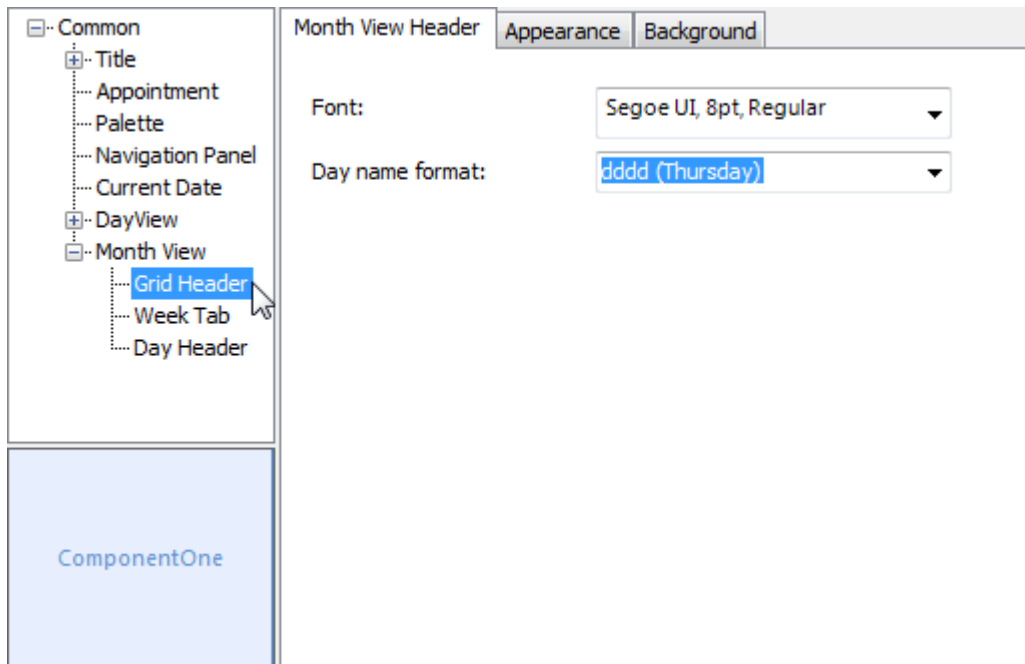
The following properties are available through the **Month View** node:

Tab	Property	Description
Month View	Font	Sets the font for the month view.
	Date format	<p>Sets the date format for the day header. Available formats include:</p> <p>* The following formats are available in the Japanese version:</p> <ul style="list-style-type: none">• d - Displays the numeric date value. For example, 2007/02/01.• MMMM d dddd - Displays the month, measured as a number between 1 and 12, the current day of the month, measured as a number between 1 and 31, inclusive, and the full name of the day. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 2 1 Thursday.• MMMM d - Displays the month, measured as a number between 1 and 12, and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 2 1.• MMMM dd - Displays the month, measured as a number between 1 and 12, and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, 2 01.• d ddd - Displays the current day of the month, measured as a number between 1 and 31, inclusive, and the full name of the day. If the day is a single digit only (1-9), then it is displayed as a

Tab	Property	Description
		<p>single digit. For example, 1 Thursday.</p> <ul style="list-style-type: none"> • d - Displays the numeric date value. For example, 2/1/2007. • dddd, MMMM dd - Displays the full name of the day and month, and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, Thursday, February 1. • MMMM d - Displays the full name of the month and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, February 1. • MMMM dd - Displays the full name of the month and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, February 01. • dd MMMM - Displays the current day of the month, measured as a number between 1 and 31, inclusive, and the full name of the month. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, 01 February. • d MMMM - Displays the current day of the month, measured as a number between 1 and 31m inclusive, and the full name of the month. If the day is a single digit only (1-9), then it is displayed as a single digit. For example, 1 February. • ddd dd - Displays the abbreviated day name and the current day of the month, measured as a number between 1 and 31, inclusive. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, Thu 01. • dd ddd - Displays the current day of the month, measured as a number between 1 and 31, inclusive, and Displays the abbreviated day name. If the day is a single digit only (1-9), it is formatted with a preceding 0 (01-09). For example, 01 Thu.
	Show week tabs	Displays week tabs when checked.
	Show time	Displays the appointment's time when checked.
	Show end time	Displays the appointment's end time when checked.

Grid Header Node

Through the **Grid Header** node of the **Visual Style** dialog box, you can set properties for the grid header.

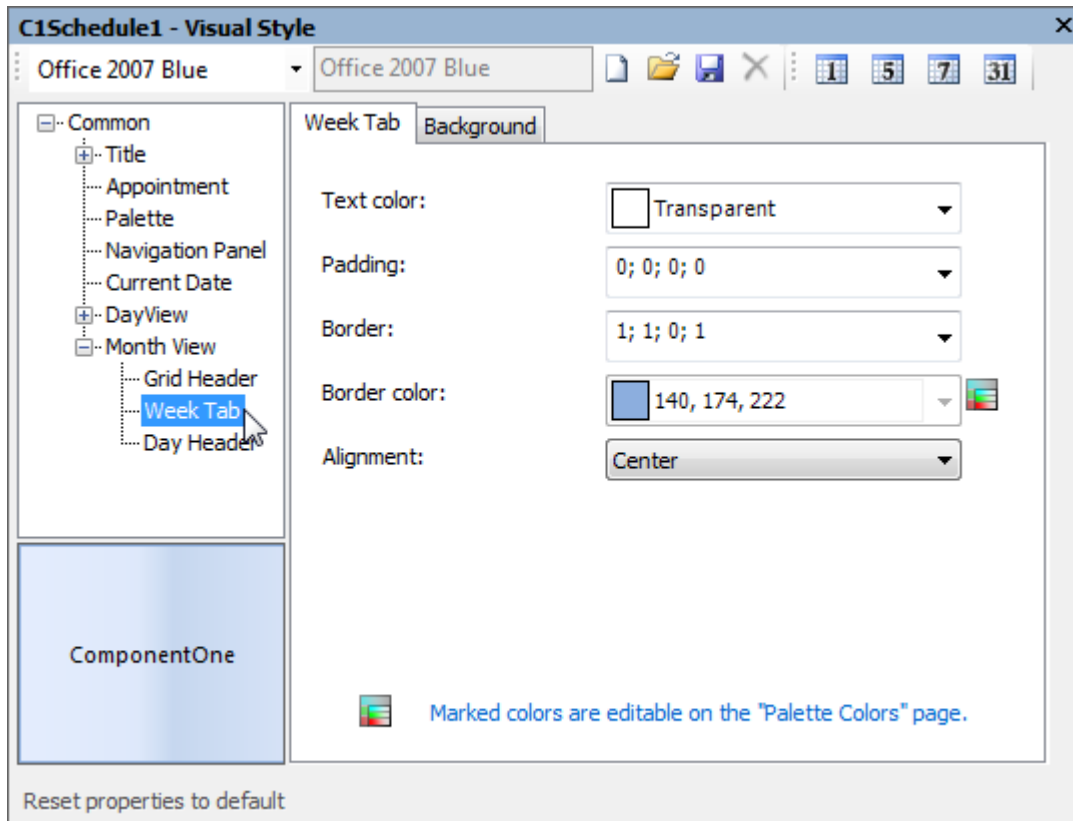


The following properties are available through the **Grid Header** node:

Tab	Property	Description
Month View Header	Font	Sets the font for the month view header.
	Day name format	Sets the date format for the header. Available formats include: dddd - Displays the full name of each day. For example, Thursday . ddd - Displays the 3-letter abbreviated name for each day. For example, Thu .
Appearance	Text color	Sets the text color for the header.
	Padding	Sets the padding around the header.
	Border	Sets the border width for the header.
	Border color	Sets the border color for the header.
	Alignment	Sets the alignment for the header text.
Background	Background color	Sets the background color for the header.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Week Tab Node

Through the **Week Tab** node of the **Visual Style** dialog box, you can set properties for the week tab.

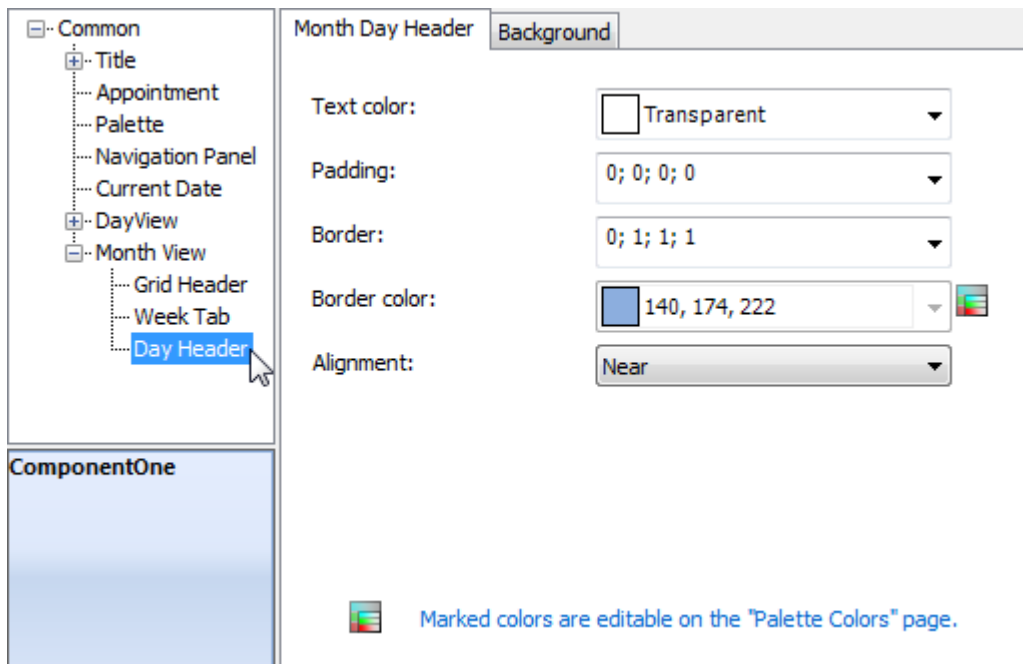


The following properties are available through the **Week Tab** node:

Tab	Property	Description
Week Tab	Text color	Sets the text color for the week tab.
	Padding	Sets the padding around the week tab.
	Border	Sets the border width for the week tab.
	Border color	Sets the border color for the week tab.
	Alignment	Sets the alignment for the week tab text.
Background	Background color	Sets the background color for the week tab.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Day Header Node

Through the **Day Header** node of the **Visual Style** dialog box, you can set properties for the day header.



The following properties are available through the **Day Header** node:

Tab	Property	Description
Month Day Header	Text color	Sets the text color for the month day header.
	Padding	Sets the padding around the month day header.
	Border	Sets the border width for the month day header.
	Border color	Sets the border color for the month day header.
	Alignment	Sets the alignment for the month day header text.
Background	Background color	Sets the background color for the month day header.
	Gradient color	Sets the background gradient color.
	Gradient	Sets the background gradient mode.
	Gamma correction	Applies gamma correction to the background gradient when checked.
	Gradient center	Sets the center of the gradient background.
	Gradient blend	Sets the Drawing.Drawing2D.Blend used to paint the background gradient.
	Image	Sets the background image.

Run-Time Interaction

C1Schedule also provides run-time interaction. The following topics describe how users of your application will interact with **Scheduler for WinForms** at run time:

C1Schedule Context Menu

At run time, the **C1Schedule** context menu allows you to create and edit appointments, navigate the calendar, as well as import or export data. For more information on how to use the context menu for the **C1Schedule** control at run time, see [C1Schedule Run-Time Context Menu](#).

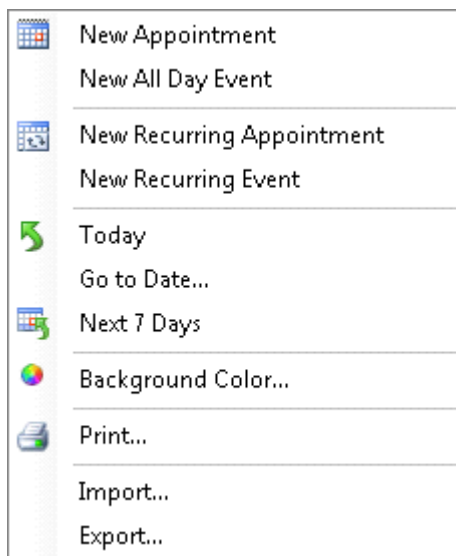
C1Calendar Interactivity

At run time, you can change the default data view by selecting a date or series of dates on the **C1Calendar** control. For more information on **C1Calendar** interactivity, see the [C1Calendar Interactivity](#).

C1Schedule Run-Time Context Menu

Using the run-time context menu, you can add appointments, navigate the calendar, print or preview the schedule, and import or export data.

If you are in the [DayView](#) or [WorkWeekView](#) views, the following menu will appear when you right-click within the **C1Schedule** control:



The following items operate as follows:

- **New Appointment**

Clicking this item opens the **Appointment** dialog box which appears like the following:

Untitled - Appointment

Save and Close Save As... Recurrence... ! ↓ X

Subject:

Location: Label: ☐ None

Start time: 2/25/2013 2:00 pm ☐ All day event

End time: 2/25/2013 2:30 pm

☒ Reminder: 15 minutes Show time as: ☐ Out of office

Contacts... Categories...

Resources... Private ☐

- **New All Day Event**

Clicking this item opens the **Event** dialog box which appears like the following:

- **New Recurring Appointment**

Clicking this item opens the **Appointment** dialog box which appears like the following:

- **New Recurring Event**

Clicking this item opens the **Appointment** dialog box.

- **Today**


Clicking on the Today highlights today's date on the calendar and opens the schedule for today's date appear

- **Go to Date**

Clicking on the Go to Date item opens the Go to Date dialog box where you can select the date in the dropdown listbox and the view from the dropdown listbox.

- **Next 7 Days**

Clicking this item forces C1Schedule to show 7 days starting from the current date. If the currently shown view is not **TimeLine**, the 7 days will be shown in the **DayView** mode.

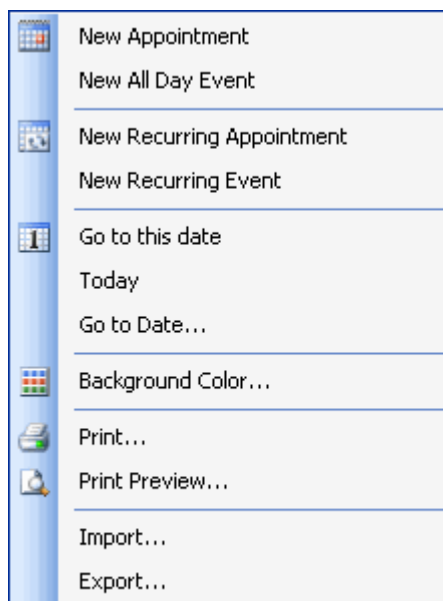
 **Note:** if you use localized resources, make sure that you added translation for this item.

- **Import**

Clicking on the Import item opens the Import dialog box where you can select the .xml file you wish to import.

- **Export**


Clicking on the Export item opens the Export dialog box where you can select the .xml to export. The WeekView and the MonthView views have an added navigation option, **Go to this date**, when you right-click within the [C1Schedule](#) control:



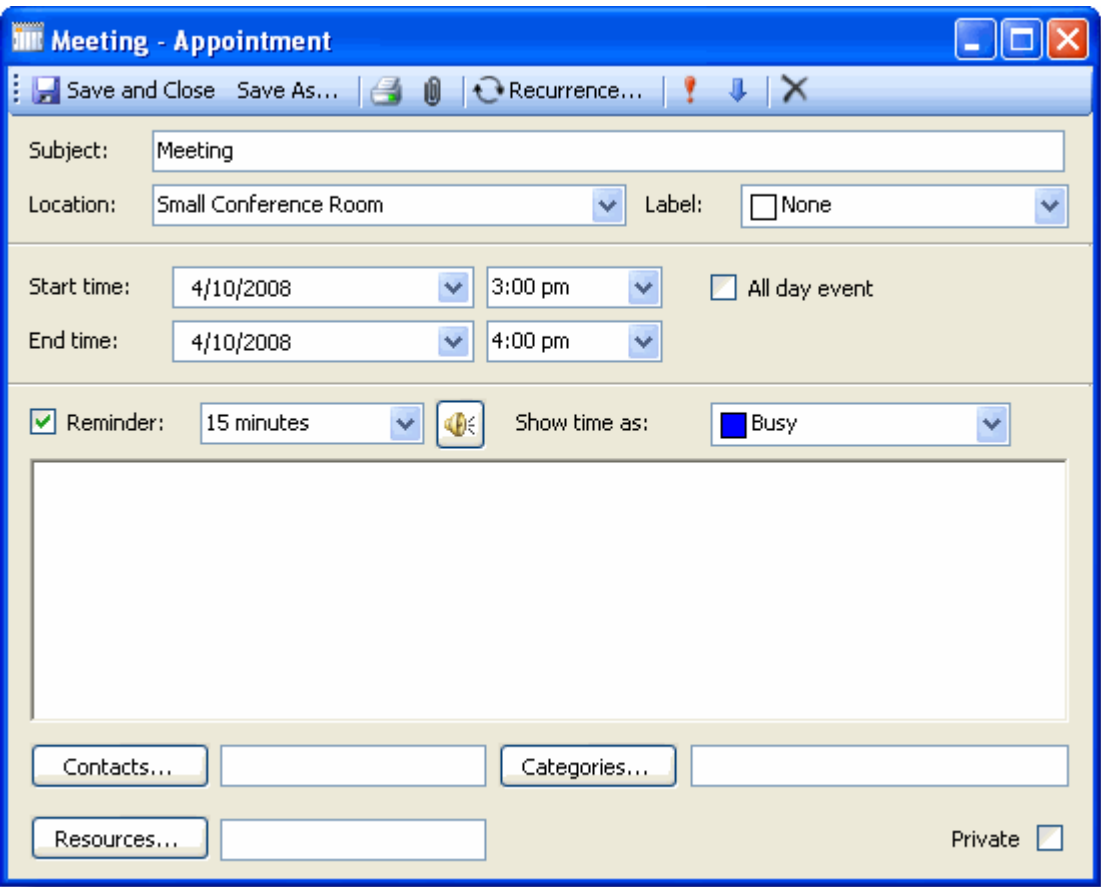
Working with Appointments

Appointments can be created at run time through the [context menu](#) by selecting one of the following options:

- New Appointment
- New All Day Event
- New Recurring Appointment
- New Recurring Event

 **Note:** Appointments can also be created by double-clicking on a day or time, which will open the **Appointment** dialog box for that day or time. For more information about appointments, see [Appointments](#).

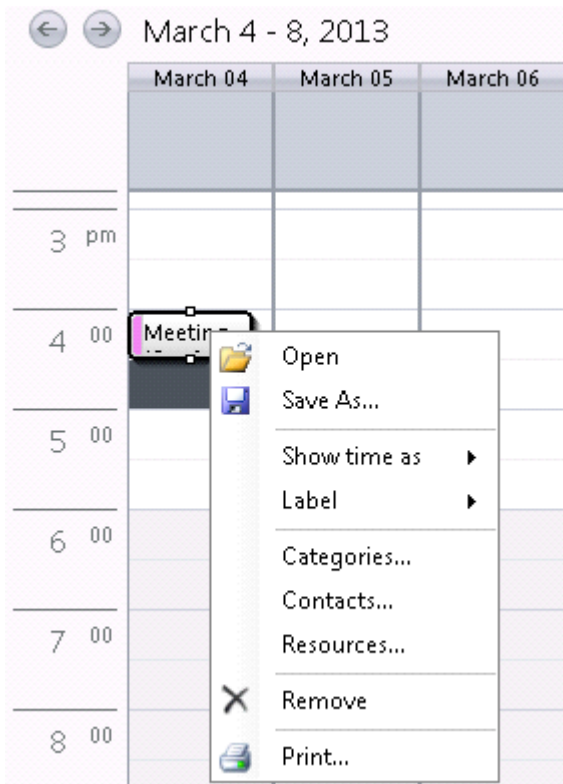
At run time, setting an appointment opens the **Appointment** dialog box.



Pressing the following keys while in the **Appointment** dialog box will result in the following actions:

Key	Action
TAB or ENTER	Moves the cursor from one field to another according to the tab order.
ESC	Closes the Appointment dialog box without saving any changes.

After an appointment has been created, selecting the appointment and right-clicking opens another context menu that allows you to edit, save, or delete the selected appointment.



Appointment Dialog Box Properties

The following properties of the [Appointment](#) class can be set in the **Appointment** dialog box at run time:

Property	Description
AllDayEvent	Gets or sets the Boolean value indicating if the current appointment is an All-day appointment.
Body	Gets or sets the String value representing the body of the Appointment object.
BusyStatus	Gets or sets the Status object indicating the busy status of the user for the appointment.
Duration	Gets or sets the TimeSpan value indicating the duration of the appointment.
End	Gets or sets the DateTime value determining the end date and time of the appointment.
Importance	Gets or sets the ImportanceEnum value indicating the relative importance level for the appointment. Default value is Normal .
Label	Gets or sets the Label object associated with this appointment.
Location	Gets or sets the String value specifying where the appointment is going to be.
Private	Gets or sets the Boolean value determining whether the calendar owner intend to keep the Appointment object

Property	Description
	private.
ReminderSet	Gets or sets the Boolean value indicating whether a reminder is associated with the appointment. Once this property is set to true, a new Reminder object is created and assigned to the Reminder property.
ReminderSoundFile	Gets or sets the String value indicating the path and file name of the sound file to play when the reminder occurs for the appointment. This property is only valid if the ReminderOverrideDefault and ReminderPlaySound properties are set to true.
ReminderTimeBeforeStart	Gets or sets the TimeSpan value indicating the interval of time the reminder should occur prior to the start of the appointment.
Sensitivity	Gets or sets the SensitivityEnum value indicating the sensitivity level of the appointment.
Start	Gets or sets the DateTime value determining the start date and time of the appointment.
Subject	Gets or sets the String value representing the subject of the Appointment object.

Opening an Appointment

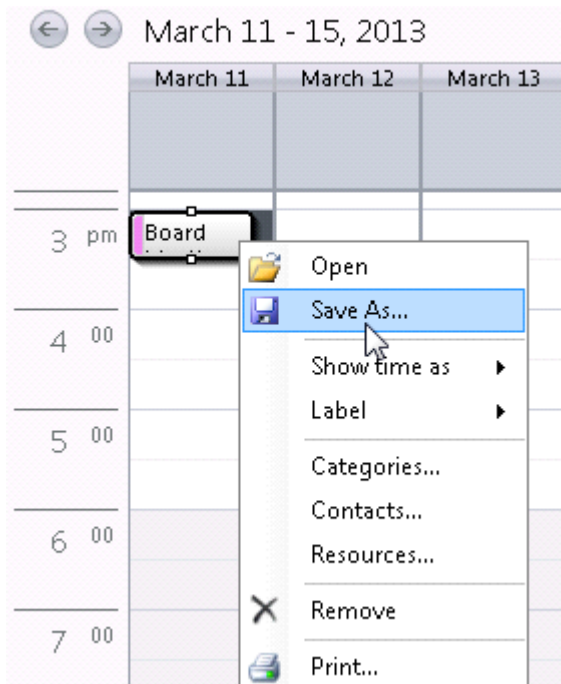
Selected appointments can be opened in the **Appointment** dialog box either through the context menu or by double-clicking on the appointment. If the appointment is recurring, the **Open Recurring Appointment** dialog box will appear with the following options:

- **Open this occurrence:** Opens only this occurrence of the selected appointment.
- **Open the series:** Opens all instances of the selected appointment.

Saving an Appointment

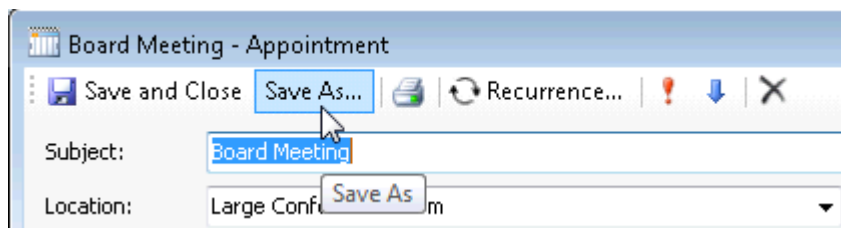
Selected appointments can be saved as binary (*.dat) files, iCal (*.ics) files, or XML files. To save an appointment in any of the file formats, either:

- Select the appointment, right-click, and select **Save As** from the context menu.



OR

- Open the appointment and select **Save As** from the toolbar in the **Appointment** dialog box.



Editing an Appointment

The context menu provides a number of options for editing an appointment without having to open the appointment in the **Appointment** dialog box. In the context menu, you can change the status or label of a selected appointment, or assign categories, contacts, or resources to it.

To change the status or label of a selected appointment, select either **Status** or **Label** from the context menu and an option from either the **Status** or **Label** submenu.

To assign categories, contacts, or resources to an appointment, click **Categories**, **Contacts**, or **Resources** from the context menu. Each option will open either the **Categories**, **Contacts**, or **Resources** dialog box, respectively.

Deleting an Appointment

Selected appointments can be removed from the schedule through the context menu or the **Appointment** dialog box. To delete an appointment through the context menu, select the appointment and select **Remove**. If the appointment is recurring, the **Confirm Removal** dialog box will appear with the following options:

- **Remove this occurrence:** Removes only the selected appointment.
- **Remove the series:** Removes all instances of the selected appointment.

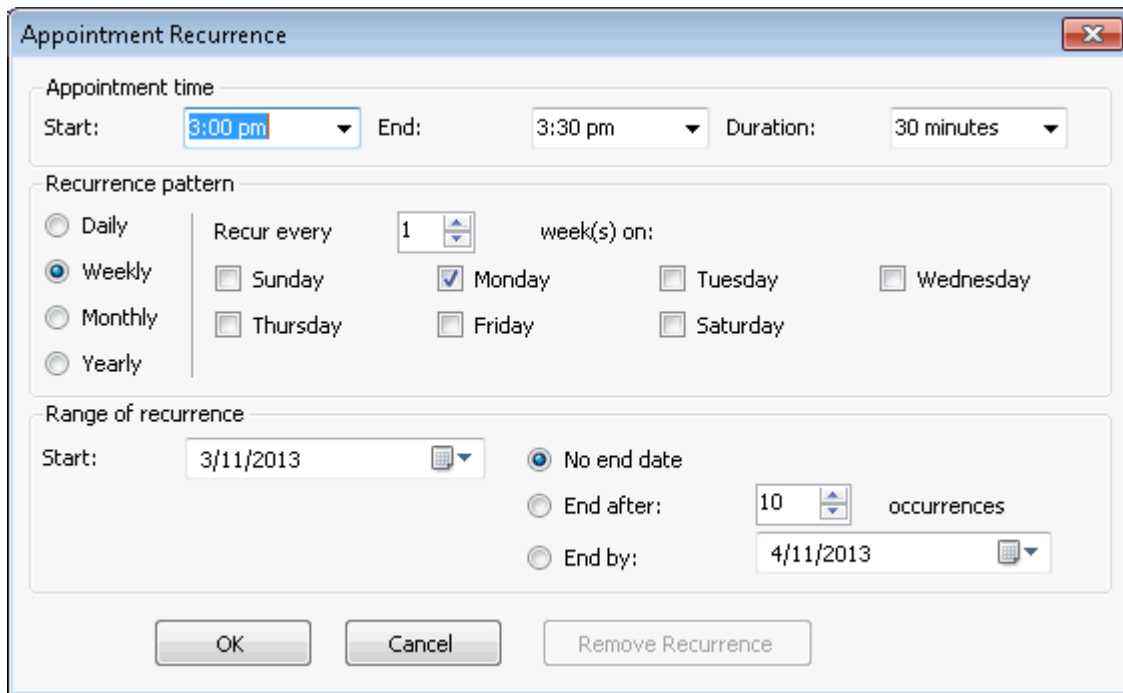
Appointments can also be removed through the **Appointment** dialog box. To remove an appointment through the

Appointment dialog box, click the **Delete** button  in the toolbar.

Recurring Appointments

Appointments can be set to recur at specified intervals. An appointment can recur daily, weekly, monthly, or yearly.

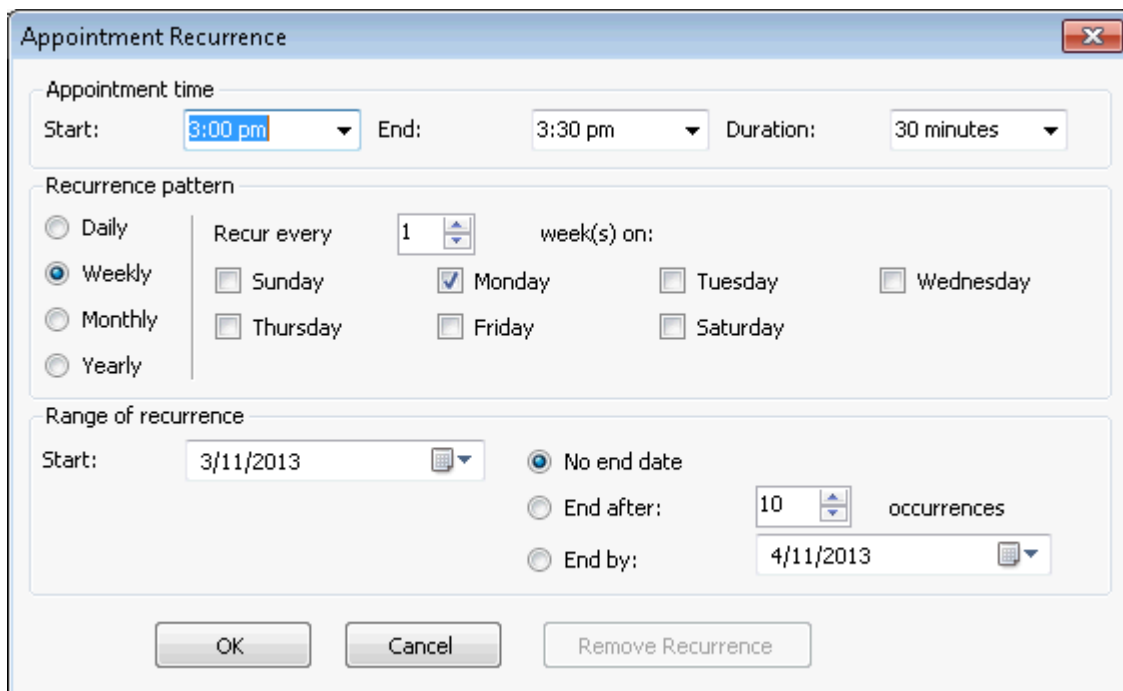
At run time, clicking the **Recurrence** button in the **Appointment** dialog box or selecting **New Recurring Appointment** or **New Recurring Event** from the context menu opens the **Appointment Recurrence** dialog box.




The **Appointment Recurrence** dialog box is shown. It has a title bar with a close button. The dialog is divided into three main sections:

- Appointment time:** Contains three dropdown menus: **Start:** (set to 3:00 pm), **End:** (set to 3:30 pm), and **Duration:** (set to 30 minutes).
- Recurrence pattern:** Contains radio buttons for **Daily**, **Weekly** (selected), **Monthly**, and **Yearly**. To the right of the **Weekly** radio button is a section labeled "Recur every" with a spinner box set to 1, followed by "week(s) on:". Below this are checkboxes for the days of the week: **Sunday**, **Monday** (checked), **Tuesday**, **Wednesday**, **Thursday**, **Friday**, and **Saturday**.
- Range of recurrence:** Contains a **Start:** date field set to 3/11/2013. To its right are three radio buttons: **No end date** (selected), **End after:** (with a spinner box set to 10 and the text "occurrences"), and **End by:** (with a date field set to 4/11/2013).

At the bottom of the dialog are three buttons: **OK**, **Cancel**, and **Remove Recurrence**.



This is an identical copy of the **Appointment Recurrence** dialog box shown above, with the same settings and layout.

 **Note:** The difference between the **New Recurring Appointment** and the **New Recurring Event** options is that for the **New Recurring Event** option, the **Duration** item is set to **1 day** and the **Start** and **End** items are set to

12:00 am (00:00).

Appointment time

The properties in the **Appointment time** group, allow you to set the start time, end time, and duration of the appointment.

Recurrence pattern

The **Recurrence pattern** group settings change depending on whether the appointment recurs **Daily**, **Weekly**, **Monthly**, or **Yearly**.

Setting	Description
Daily	<p>The Daily settings allow you to repeat an appointment every specified number of days or only on working days.</p> <div> <div>Recurrence pattern</div> <div> <input checked="" type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly </div> <div> <input checked="" type="radio"/> Every 1 day(s) <input type="radio"/> Every weekday </div> </div> <p>example, setting the appointment to Every 2 day(s) will make the appointment appear every other day. Setting the appointment to Every weekday will make the appointment appear only Monday through Friday, by default. You can change which days are set as week days in the Calendar Settings dialog in the C1Schedule Smart Designer.</p>
Weekly	<p>The Weekly settings allow you to repeat the appointment on every specified week on specified days.</p> <div> <div>Recurrence pattern</div> <div> <input type="radio"/> Daily <input checked="" type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly </div> <div> Recur every 1 week(s) on: <input type="checkbox"/> Sunday <input checked="" type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input type="checkbox"/> Wednesday <input type="checkbox"/> Thursday <input type="checkbox"/> Friday <input type="checkbox"/> Saturday </div> </div> <p>example, setting the appointment to Recur every 2 week(s) on and selecting Tuesday and Thursday will repeat the appointment every other week on Tuesdays and Thursdays.</p>
Monthly	<p>The Monthly settings allow you to repeat an appointment on every specified date at a specified interval of months or on a specified day and week of the month at a specified interval of months.</p> <div> <div>Recurrence pattern</div> <div> <input type="radio"/> Daily <input type="radio"/> Weekly <input checked="" type="radio"/> Monthly <input type="radio"/> Yearly </div> <div> <input checked="" type="radio"/> Day 11 of every 1 month(s) <input type="radio"/> The first Monday of every 1 month(s) </div> </div> <p>example, setting the appointment to Day 8 of every 2 month(s) will make the appointment appear on the 8th of every other month. Setting the appointment to The 3 Monday of every 2 month(s) will make the appointment appear on the 3rd Monday of every other month.</p>
Yearly	<p>The Yearly setting allows you to repeat an appointment on a specified date or a specified day and</p>

Setting	Description
	<p>week of a specified month.</p> <div> <div> Recurrence pattern </div> <div> For </div> </div> <div> <div> <input type="radio"/> Daily </div> <div> <input checked="" type="radio"/> Every </div> <div> <input type="radio"/> Weekly </div> <div> <input type="radio"/> Monthly </div> <div> <input checked="" type="radio"/> Yearly </div> </div> <div> <div> March </div> <div> 11 </div> </div> <div> <div> The </div> <div> first </div> <div> Monday </div> <div> of </div> <div> March </div> </div>

Range of recurrence

The **Range of recurrence** group allows you to set a time span for the recurrence. The **Start** drop-down represents the date that the recurrence will start from. There are three options to choose from for an end date:

- **No end date** will make the appointment recur indefinitely.
- **End after 0 occurrences** will make the appointment recur a specified number of times. For example, if an appointment repeated every day, setting **End after 25 occurrences** would allow the appointment to repeat every day 25 times.
- **End by** will make the appointment recur until that date.

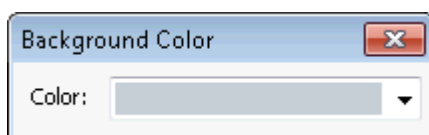
Mouse and Keyboard Navigation

C1Schedule includes several mouse and keyboard options for increased accessibility when navigating appointments. At run time, end users can use the following keyboard and mouse wheel combinations to navigate the schedule:

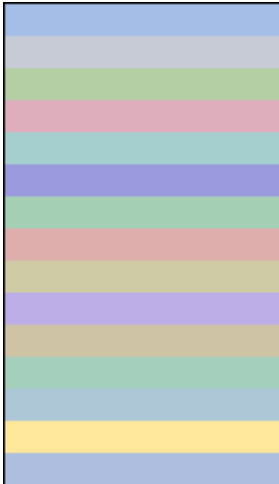
- Clicking on an **Appointment** element selects the appointment.
- Clicking the mouse or pressing the F2 key on the selected Appointment element switches appointment into the inline edit mode.
- Double-clicking on the Appointment element opens the **Appointment** dialog box.
- Pressing the ENTER key in inline edit mode finishes editing.
- Pressing the ESC key in inline edit mode cancels all changes.

Changing the Background Color

Using the **C1Schedule** control built-in color scheme collection, you can change the color palette of the schedule through the context menu at run time. Selecting **Background Color** from the context menu opens the **Background Color** dialog box.



In the Office 2007 themes, the following colors are available in the **Color** drop-down in the **Background Color** dialog box:



The default background color in the Windows XP, Royale, Yahoo, and Aero themes is blue (the first in the above list). In the Windows XP, Royale, Yahoo, and Aero themes the following colors are available in the **Color** drop-down in the **Background Color** dialog box:



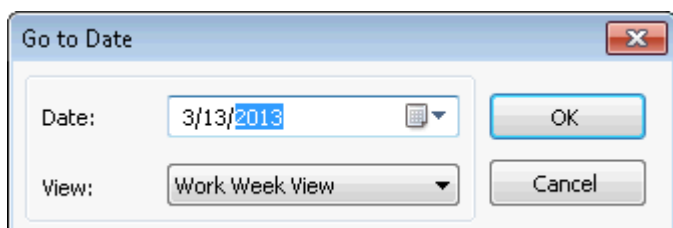
The default background color in the Windows XP, Royale, Yahoo, and Aero themes is white (the second in the above list).

Navigating the Calendar

Using the context menu at run time, you can navigate through the calendar for a specific date, today's date, or a selected date (when in the [WeekView](#) and the [MonthView](#) views).

Go to Date

Clicking **Go to Date** in the context menu opens the **Go to Date** dialog box.



Select a date that you would like to be displayed and the view that you would like to display it in. For example, selecting **3/17/2007** in the **Date** drop-down and **Week View** in the **View** drop-down will display the week containing March 17, 2007.

Today

Clicking **Today** in the context menu display today's date depending on the data view in the [C1Schedule](#) control:

- If the current data view is the [DayView](#) view, clicking **Today** will display today's date in DayView, regardless if today's date is visible within the DayView view.
- If the current data view is the [WorkWeekView](#) view, clicking **Today** will display the work week that includes today's date, regardless if today's date is visible within the WorkWeekView view.
- If the current data view is the [WeekView](#) view, clicking **Today** will display the week that includes today's date, regardless if today's date was visible within the WeekView.
- If the current data view is the [MonthView](#) view, clicking **Today** will display the month view that includes today's date, regardless if today's date was visible within the MonthView.

Go to this date (WeekView and MonthView views only)

Clicking **Go to this date** opens the selected date in DayView view.

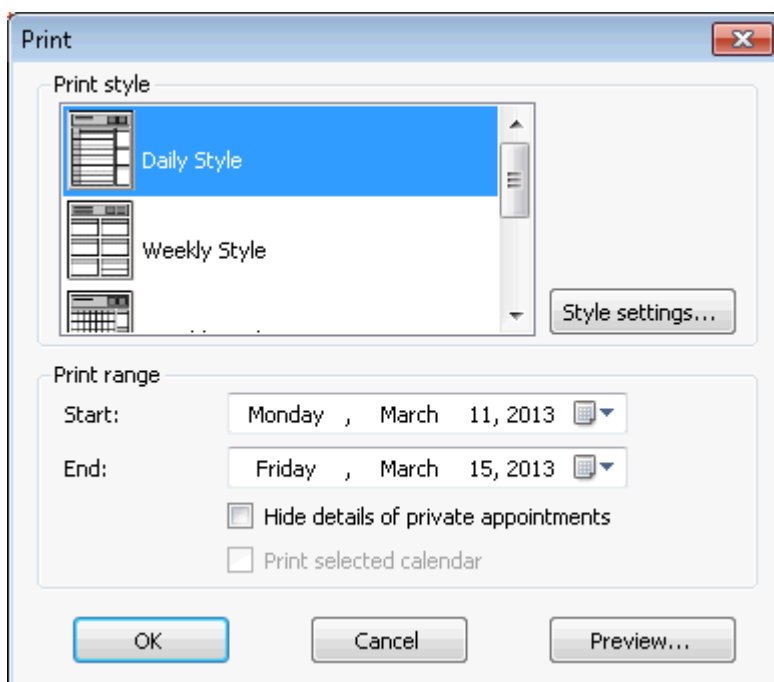
Printing and Previewing Data

Printing support was added to **Scheduler for WinForms** in 2008 v3. At run time users can now print the schedule and view a print preview of schedule appointment data using the run-time context menu or buttons on the toolbar of the Appointment dialog box.

Opening Print Options


- Clicking **Print** will open the **Print** dialog box, which allows you to select the print style and print range before printing.
- Clicking **Print Preview** will open the **Print** dialog box, which allows you to select the print style and print range before previewing your selection.

The **Print** dialog box appears similar to the following:



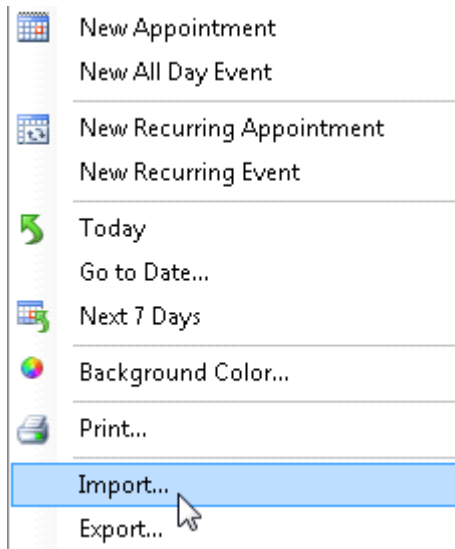
The **Print** dialog box includes options to select a print style (daily, weekly, monthly, details) and a print range. The

print style can be customized by clicking the **Style settings** button and users have the option of hiding the details of private appointments.


 **Note:** If users select a specific appointment to print (right-click the appointment and select **Print** or **Print Preview**) the print range will not be visible and cannot be changed in the **Print** dialog box.

Importing and Exporting Data

At run time you can import and export your appointment data using the context menu:




- Clicking **Export** will open the **Export Data to File** dialog box, which allows you to save your appointment data as a binary (*.dat), an iCal (*.ics), or an XML file.
- Clicking **Import** will open the **Import Data** dialog box which allow you to open previously exported data into the [C1Schedule](#) control.

 **Tip:** While the current version of **Scheduler for WinForms** does not support direct export of data to Microsoft Outlook, you can export appointment data to the iCal (*.ics) format using [C1Schedule](#) and then import data into another scheduling application, such as Outlook.

C1Calendar Interactivity

Like the [C1Schedule](#) control, the [C1Calendar](#) control also has run-time interactivity. At run time, you can change the current month or year, or display a different data view by selecting a date or series of dates on the [C1Calendar](#) control.

 **Note:** Weekends and Holidays appear in bold text in the calendar.

Changing the Month or Year

You can easily navigate through months or years in the calendar at run time. To do so, click once on the month or year shown in the [C1Calendar](#) control's header. A pop-up selector will appear allowing you to choose a different month or year:

February 2013						
Mo	Tu	We	Th	Fr	Sa	Su
5	28	29	30	31	1	2
6	4	5	6	7	8	9
7	11	12	13	14	15	16
8	18	19	20	21	22	23
9	25	26	27	28	29	30
March 2013						
Mo	Tu	We	Th	Fr	Sa	Su
9	4	5	6	7	8	9
10	11	12	13	14	15	16
11	18	19	20	21	22	23
12	25	26	27	28	29	30
13	1	2	3	4	5	6

Changing the View

At run time, you can change the current data view by selecting a date or series of dates on the [C1Calendar](#) control. To change the view select a date ranges by clicking the start date, holding down the SHIFT key, and clicking the last date in the date range. You can set the following views:

- **Day View**

Click on the specified date in the calendar.

March 2013						
Mo	Tu	We	Th	Fr	Sa	Su
9	25	26	27	28	1	2
10	4	5	6	7	8	9
11	11	12	13	14	15	16
12	18	19	20	21	22	23
13	25	26	27	28	29	30

- **Work Week View**

Click on the first day in the work week and the last day of the work week.

March 2013						
Mo	Tu	We	Th	Fr	Sa	Su
9	25	26	27	28	1	2
10	4	5	6	7	8	9
11	11	12	13	14	15	16
12	18	19	20	21	22	23
13	25	26	27	28	29	30

- **Week View**

Click on the first day of the week and the last day of the week.

March 2013						
Mo	Tu	We	Th	Fr	Sa	Su
9	25	26	27	28	1	2
10	4	5	6	7	8	9
11	11	12	13	14	15	16
12	18	19	20	21	22	23
13	25	26	27	28	29	30

- **Month View**

Click on the first day of the week and the last day of the month (at least 4 weeks).

March 2013							
	Mo	Tu	We	Th	Fr	Sa	Su
9	25	26	27	28	1	2	3
10	4	5	6	7	8	9	10
11	11	12	13	14	15	16	17
12	18	19	20	21	22	23	24
13	25	26	27	28	29	30	31

Using Mouse and Keyboard Navigation

[C1Calendar](#) includes several mouse and keyboard navigation options for increased accessibility. At run time, end users can use the following keyboard and mouse wheel combinations to navigate the calendar:

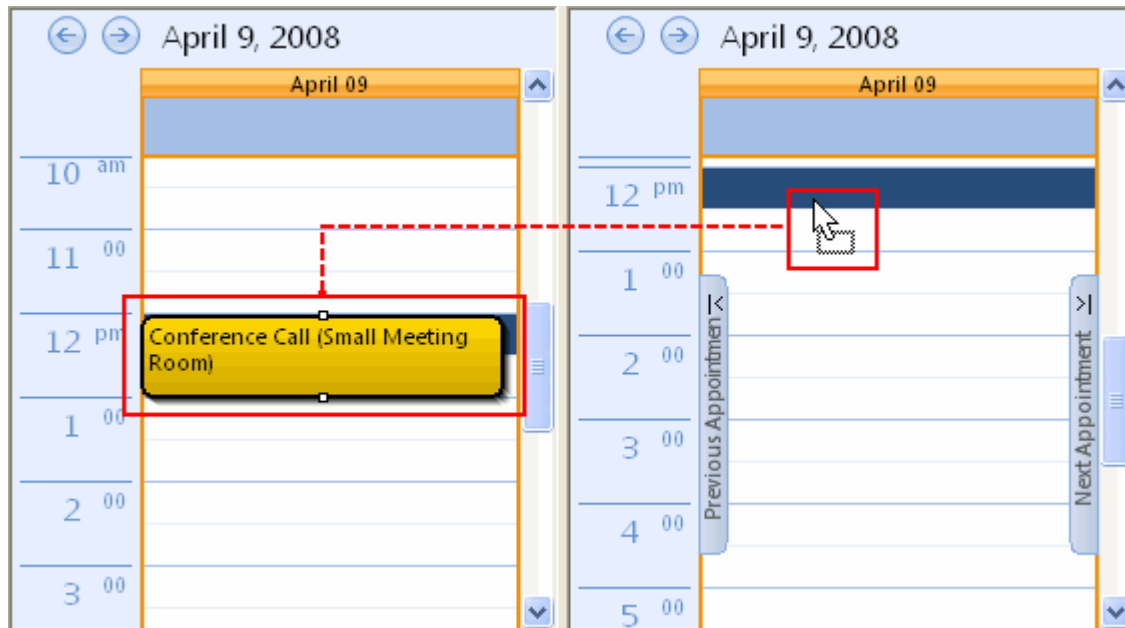
Key/Mouse	Description
PAGE UP	Navigates to the previous month.
PAGE DOWN	Navigates to the next month.
Mouse Wheel	Navigates one month backward or forward.
CTRL + PAGE UP	Navigates to the previous year.
CTRL + PAGE DOWN	Navigates to the next year.
CTRL + Mouse Wheel	Navigates one year backward or forward.
HOME	Navigates to current date.

Drag-and-Drop Support

The [C1Schedule](#) and [C1Calendar](#) controls include drag-and-drop support. At run time, users can interact with **Scheduler for WinForms** with Outlook-style drag-and-drop functionality. Note that you can use the [BeforeAppointmentDrop](#) event to cancel a drag-and-drop operation or customize drag-and-drop behavior.

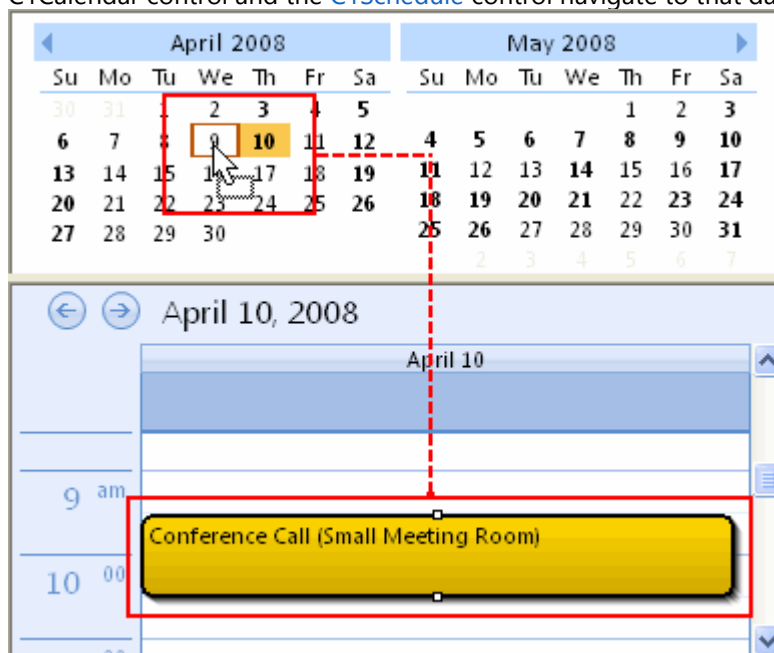
At run time, users can:

- **Drag appointments from one C1Schedule control to another.**
Users can perform a drag-and-drop operation at run time to move an appointment from one C1Schedule control to another if the C1Schedule. **AllowDrop** property for each C1Schedule control has been set to **True**.



- **Drag appointments onto a C1Calendar control.**

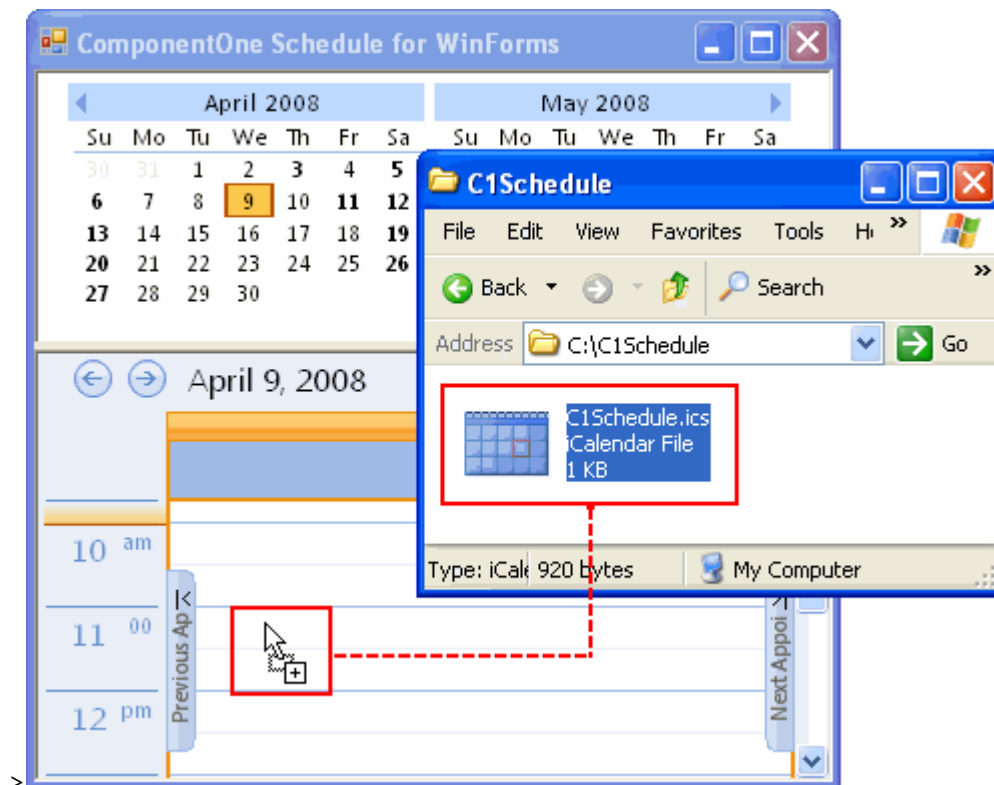
Users can drag appointments onto a [C1Calendar](#) control at run time if [C1Calendar.AllowDrop](#) is **True** and the [Schedule](#) property is not null. The [Start](#) property will be changed according to the date selected in the [C1Calendar](#) control and the [C1Schedule](#) control navigate to that date.



Note: if there are several [C1Schedule](#) controls in application, appointment will be moved to the [C1Schedule](#) control specified in [Schedule](#) property.

- **Drag .ics, .dat and .xml files with saved schedule data onto a C1Schedule control.**

Users can drag files with saved schedule data onto a [C1Schedule](#) control at run time if [C1Schedule.AllowDrop](#) is **True**.



Scheduler for WinForms Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos, which may make use of other development tools included with the ComponentOne Studio.

Please refer to the pre-installed product samples through the following path:

Documents\ComponentOne Samples\WinForms

Click one of the following links to view a list of **ComponentOne Scheduler for WinForms** samples:

Visual Basic Samples

Sample	Description
C1ScheduleDemo	The C1Schedule control displays appointments data and responds to end-users input, while the C1Calendar control lets end-users change the date or the set of dates for which the C1Schedule control currently shows its appointments data. This selection is passed to the assigned C1Schedule control, which then loads all the necessary data and represents them using appropriate view.
CustomForms	<p>The CustomData application demonstrates using the C1Schedule and C1Calendar controls for creating and keeping a timetable for a Fitness Center. The sample shows how to:</p> <ul style="list-style-type: none"> • Bind to custom data. • Use custom Appointment Form. • Work with components from code. <p>The sample shows timetable in a WeekView. It prevents C1Schedule from switching to other views. It does not show built-in context menus or a ReminderForm. The ExerciseForm class represents a form that is used instead of the built-in AppointmentForm. Users can:</p> <ul style="list-style-type: none"> • Create new exercises by double-clicking on desired day and time. • Edit exercises in-place or in ExerciseForm. • Copy timetable information from one day/week to another one. • Clear current week timetable. <p>On closing, the application saves all timetable information to the FitnessClub database, which is included into solution.</p>
DragDrop	This sample demonstrates a drag and drop operation from FlexGrid for WinForms - to Scheduler for WinForms . After opening the sample, type any information in the C1FlexGrid cell and drag this cell onto C1Schedule . This sample uses the C1Flexgrid component as well as the C1Schedule component.
MultiUser	This sample demonstrates how to assign owners to Appointments and filter data according to the currently selected user. The MultiUser application consists of the C1Schedule and C1Calendar controls with the AppointmentStorage and ContactStorage storages bound to data from the C1NWind database (which is included into solution). Other storages in the sample are used in unbound mode. The Appointments table from C1NWind.mdb contains an Owner column for keeping information about Appointment's owner. This column may contain key values from the Employees table. The currently selected user (an owner) is shown in the toolbar combobox. At changing the owner the application applies a filter to the

Sample	Description
	C1Schedule's datasource to show only the appointments for the selected owner. At the same time, it sets default value for the Appointment table Owner column (to save Owner the index along with other appointment information).
UnBound	This sample demonstrates using C1Schedule in unbound mode. When the application loads, it loads data from an XML file and saves data to the same XML file at exit. If there is no previously saved data, the application creates a new XML file in the working folder. The end-user can import and export data at any time by using the Import/Export items from the context menu of the C1Schedule component.

C# Samples

Sample	Description
C1ScheduleDemo	The C1Schedule control displays appointments data and responds to end-users input, while the C1Calendar control lets end-users change the date or the set of dates for which the C1Schedule control currently shows its appointments data. This selection is passed to the assigned C1Schedule control, which then loads all the necessary data and represents them using appropriate view.
CustomData	<p>The CustomData application demonstrates using the C1Schedule and C1Calendar controls for creating and keeping a timetable for a Fitness Center.</p> <p>The sample shows how to:</p> <ul style="list-style-type: none"> • Bind to custom data. • Use custom Appointment Form. • Work with components from code. <p>The sample shows timetable in a WeekView. It prevents C1Schedule from switching to other views. It does not show built-in context menus or a ReminderForm. The ExerciseForm class represents a form that is used instead of the built-in AppointmentForm.</p> <p>Users can:</p> <ul style="list-style-type: none"> • Create new exercises by double-clicking on desired day and time. • Edit exercises in-place or in ExerciseForm. • Copy timetable information from one day/week to another one. • Clear current week timetable. <p>On closing, the application saves all timetable information to the FitnessClub database, which is included into solution.</p>
FullBound	<p>The FullBound application consists of the C1Schedule and C1Calendar controls, with all C1Schedule storages are bound to data from the C1NWind database (which is included into solution). The C1NWind.mdb contains 6 tables (one table per storage):</p> <ul style="list-style-type: none"> • Appointments - the sample uses it to store Appointments' data. • Categories - the sample uses it to store Categories' data. • Employees - the sample uses it to store Contacts' data. • Labels - the sample uses it to store Labels' data. • Products - the sample uses it to store Resources' data. • Statuses - the sample uses it to store Statuses' data.

Sample	Description
	<p>The C1Schedule control displays appointments data and responds to end-users input, while the C1Calendar control lets end-users change the date or the set of dates for which the C1Schedule control currently shows its appointments data. This selection is passed to the assigned C1Schedule control, which then loads all the necessary data and represents them using appropriate view.</p>
MultiUser	<p>This sample demonstrates how to assign owners to Appointments and filter data according to the currently selected user. The MultiUser application consists of the C1Schedule and C1Calendar controls with the AppointmentStorage and ContactStorage storages bound to data from the C1NWind database (which is included into solution). Other storages in the sample are used in unbound mode.</p> <p>The Appointments table from C1NWind.mdb contains an Owner column for keeping information about Appointment's owner. This column may contain key values from the Employees table.</p> <p>The currently selected user (an owner) is shown in the toolbar combobox. At changing the owner the application applies a filter to the C1Schedule's datasource to show only the appointments for the selected owner. At the same time, it sets default value for the Appointment table Owner column (to save Owner the index along with other appointment information).</p>
PrintDocTemplates	<p>Demonstrates creating C1.C1Preview.C1PrintDocument templates for printing C1Schedule's views.</p>
UnBound	<p>This sample demonstrates using C1Schedule in unbound mode. When the application loads, it loads data from an XML file and saves data to the same XML file at exit. If there is no previously saved data, the application creates a new XML file in the working folder. The end-user can import and export data at any time by using the Import/Export items from the context menu of the C1Schedule component.</p>

Scheduler for WinForms Task-Based Help

The task-based help assumes that you are familiar with programming in Visual Studio, and know how to use bound and unbound controls in general. If you are a novice to the **Scheduler for WinForms** product, please see the [Scheduler for WinForms Quick Start](#) first.


Each topic provides a solution for specific tasks using the **Scheduler for WinForms** product. By following the steps outlined in the help, you will be able to create projects demonstrating a variety of **Scheduler for WinForms** features.

Each task-based help topic also assumes that you have created a new .NET project.

Creating a Bindable Microsoft Access Database

To create a bindable Microsoft Access database:

1. Create a new Microsoft Access database.
2. Add tables with the fields and settings listed below.

 **Note:** This topic assumes that you are familiar with creating a database in Microsoft Access and setting up tables. If you are a novice to Microsoft Access, please see the Microsoft Office Help for more information.

Appointments Table

The following fields and settings are for the **Appointments** table:

Field Name	Data Type	Field Size	Required	Indexed	Primary Key
AppointmentID	AutoNumber	Replication ID	Yes	Yes (No Duplicates)	X
Subject	Text	255	Yes	No	--
Location	Text	255	No	No	--
Start	Date/Time	--	No	No	--
End	Date/Time	--	Yes	No	--
Body	Memo	--	No	No	--
Properties	Memo	--	No	No	--

Categories Table

The following fields and settings are for the **Categories** table:

Field Name	Data Type	Field Size	Required	Indexed	Primary Key
CategoryID	AutoNumber	Long Integer	Yes	Yes (No Duplicates)	X
CategoryName	Text	255	Yes	Yes (No Duplicates)	--

Contacts Table

The following fields and settings are for the **Contacts** table:

Field Name	Data Type	Field Size	Required	Indexed	Primary Key
ContactID	AutoNumber	Long Integer	Yes	Yes (No Duplicates)	X
Name	Text	255	Yes	Yes (No Duplicates)	--


Resources Table

The following fields and settings are for the **Resources** table:

Field Name	Data Type	Field Size	Required	Indexed	Primary Key
ResourceID	AutoNumber	Long Integer	Yes	Yes (No Duplicates)	X
ResourceName	Text	255	Yes	Yes (No Duplicates)	--

Mapping the Microsoft Access Database to a C1Schedule Control

To map the Microsoft Access database to a [C1Schedule](#) control, set the mapping properties in the **Data Source settings** dialog box in the **C1Schedule Smart Designer**. For more information on accessing the **C1Schedule Smart Designer**, see [C1Schedule Smart Designer](#).

 **Note:** The following data mappings use the Microsoft Access database created in the [Creating a Bindable Microsoft Access Database](#) topic.

1. Open the **Data Source settings** dialog box in the [C1Schedule Smart Designer](#).
2. On the **AppointmentStorage** tab, map the **Appointments** table by selecting the following field names from the drop-down lists:

Data Mapping Property	Field Name
IdMapping	AppointmentID
IndexMapping	(none)
Properties	Properties
Body	Body
End	End
Location	Location
Start	Start
Subject	Subject

3. On the **CategoryStorage** tab, map the **Categories** table by selecting the following field names from the drop-down lists:

Data Mapping Property	Field Name
IdMapping	(none)
IndexMapping	CategoryID
CaptionMapping	CategoryName
TextMapping	CategoryName

- On the **ContactStorage** tab, map the **Contacts** table by selecting the following field names from the drop-down lists:

Data Mapping Property	Field Name
IdMapping	(none)
IndexMapping	ContactID
CaptionMapping	Name
ColorMapping	(none)
TextMapping	Name

- On the **ResourceStorage** tab, map the **Resources** table by selecting the following field names from the drop-down lists:

Data Mapping Property	Field Name
IdMapping	(none)
IndexMapping	ResourceID
CaptionMapping	ResourceName
ColorMapping	(none)
TextMapping	ResourceName

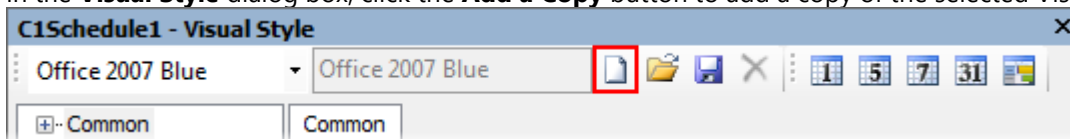
- Close the **Data Source settings** dialog box.

Creating a Custom Visual Style

To create a custom Visual Style, add a copy of the selected Visual Style in the **Visual Style** dialog box:

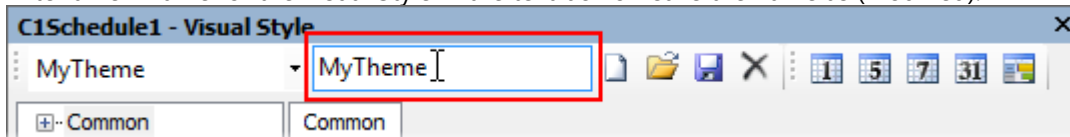
Note: Selecting **Customize** from the drop-down menu in the **C1Schedule Tasks** menu, **C1Calendar Tasks** menu, or the Properties window opens the **Visual Style** dialog box.

- In the **Visual Style** dialog box, click the **Add a Copy** button to add a copy of the selected Visual Style.



A new Visual Style will be created based on the predefined visual style, in this case on the **Office 2007 Blue** visual style.

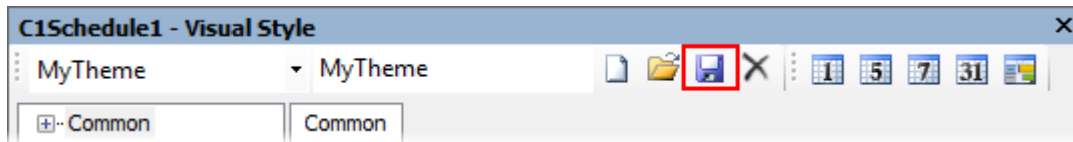
- Enter a new name for the Visual Style in the text box or leave the name as (modified).



- The new Visual Style will now appear in the **Visual Style** drop-down list.

Saving Custom Visual Style

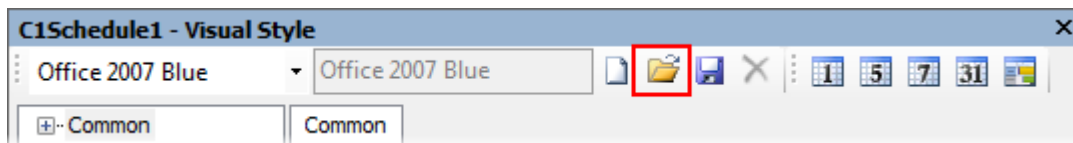
To save a custom Visual Style, click the **Save** button in the **Visual Style** dialog box.



Note: This will save all of the custom Visual Styles added to the Visual Style dialog box to an XML file.

Importing Saved Visual Style

To import a saved Visual Style, click the **Load** button in the **Visual Style** dialog box.



Note: This will load all of the custom Visual Styles in the XML file.

Deleting a Custom Visual Style

To delete a custom Visual Style, click the **Delete** button in the **Visual Style** dialog box.



Note: This will delete the current Visual Style. Predefined Visual Styles cannot be deleted.

Formatting the Border Style

Formatting the border style allows you to customize the appearance of the **Scheduler for WinForms** controls. The border style can be set for both the [C1Schedule](#) and the [C1Calendar](#) controls.

Formatting the Border Style of the C1Schedule Control

To format the border style of the control, set the [BorderStyle](#) property to **Fixed3D**, **FixedSingle**, or **None**. This property can be set either in the designer or in code.

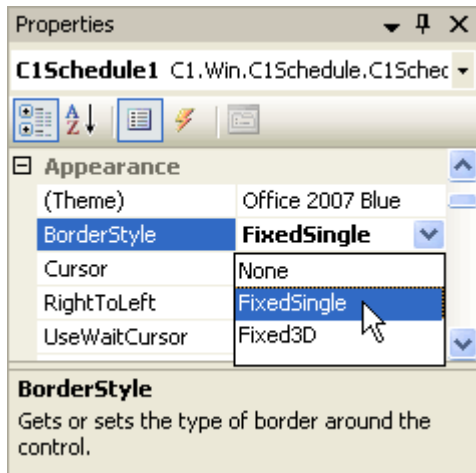
The following table describes each of the border styles:

Border	Description
Fixed3D	A three-dimensional border. This is the default setting.
FixedSingle	A single line border.
None	No border.

In the Designer

Locate the **BorderStyle** property in the Properties window and set it to **Fixed3D**, **FixedSingle**, or **None**.

In this example, the **BorderStyle** property is set to **FixedSingle**:



In Code

Add code to the **Form_Load** event to set the **BorderStyle** property to **Fixed3D**, **FixedSingle**, or **None**. The following code sets the **BorderStyle** property to **FixedSingle**:

To write code in Visual Basic

Visual Basic

```
Me.C1Schedule1.BorderStyle = BorderStyle.FixedSingle
```

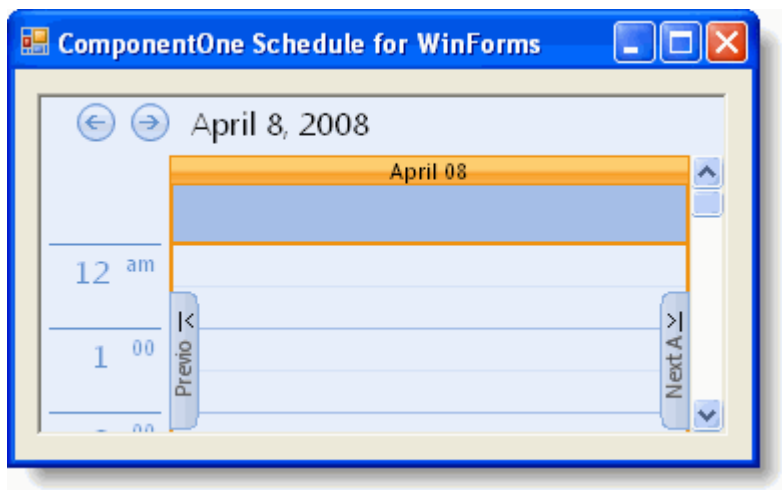
To write code in C#

C#

```
this.c1Schedule1.BorderStyle = BorderStyle.FixedSingle;
```

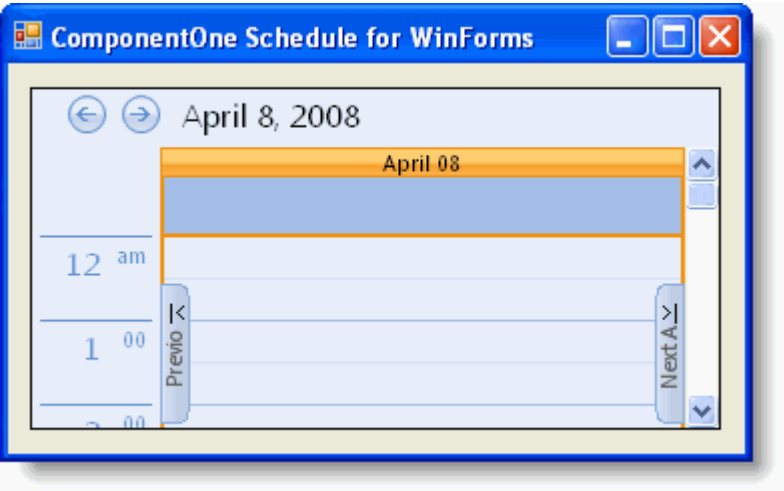
Three-Dimensional Border

If the **BorderStyle** property is set to **Fixed3D**, the border will be appear three-dimensional:



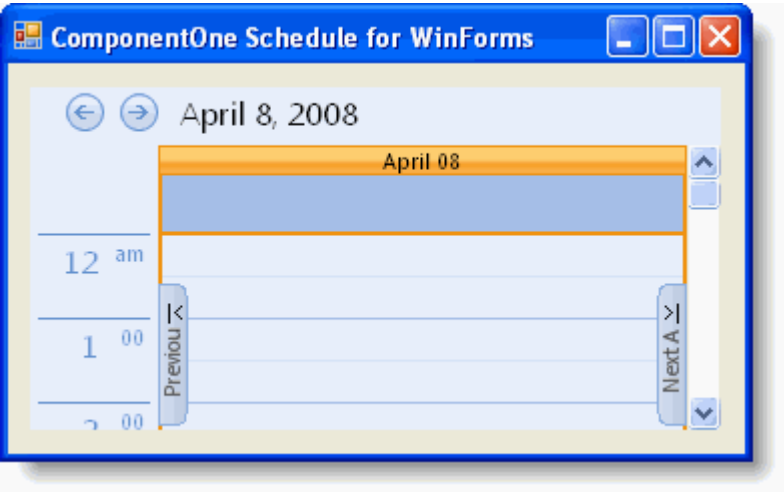
Single Line Border

If the `BorderStyle` property is set to **FixedSingle**, the border will appear as a single line:



No Border

If the `BorderStyle` property is set to **None**, no border will be visible:



Formatting the Border Style of the C1Calendar Control

To format the border style of the control, set the **BorderStyle** property to **Fixed3D**, **FixedSingle**, or **None**. This property can be set either in the designer or in code.

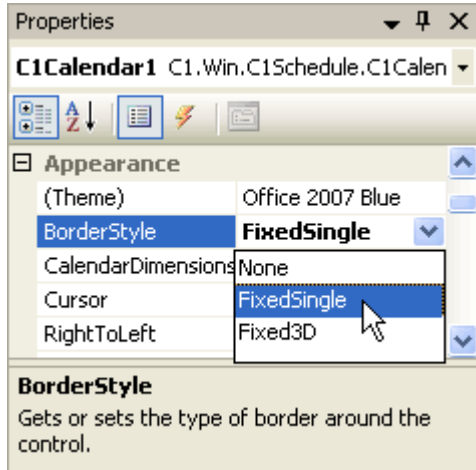
The following table describes each of the border styles:

Border	Description
Fixed3D	A three-dimensional border. This is the default setting.
FixedSingle	A single line border.
None	No border.

In the Designer

Locate the **BorderStyle** property in the Properties window and set it to **Fixed3D**, **FixedSingle**, or **None**.

In this example, the **BorderStyle** property is set to **FixedSingle**:



In Code

Add code to the **Form_Load** event to set the **BorderStyle** property to **Fixed3D**, **FixedSingle**, or **None**. The following code sets the **BorderStyle** property to **FixedSingle**:

To write code in Visual Basic

Visual Basic

```
Me.C1Calendar1.BorderStyle = BorderStyle.FixedSingle
```

To write code in C#

C#

```
this.c1Calendar1.BorderStyle = BorderStyle.FixedSingle;
```

Three-Dimensional Border

If the **BorderStyle** property is set to **Fixed3D**, the border will be appear three-dimensional:



Single Line Border

If the `BorderStyle` property is set to **FixedSingle**, the border will appear as a single line:



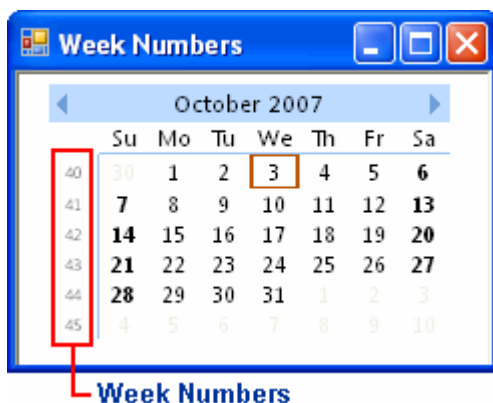
No Border

If the `BorderStyle` property is set to **None**, no border will be visible:



Hiding the Week Numbers

By default week numbers appear on the `C1Calendar` control to the left of the calendar dates.

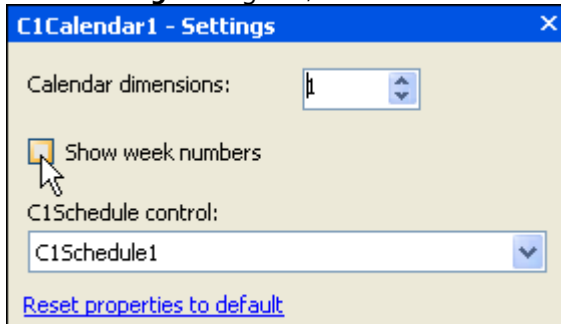


To hide the week numbers at design time, set the `ShowWeekNumbers` property to **False** either in the Smart Designer, Tasks menu, Properties window, or in code.

In the Smart Designer

To hide the week numbers, set the ShowWeekNumbers property to **False** in the Smart Designer:

1. On the **C1Calendar Smart Designer**, click the **Settings** button. For more information on accessing the **C1Calendar Smart Designer**, see [C1Calendar Smart Designer](#).
2. In the **Settings** dialog box, uncheck the **Show week numbers** check box.

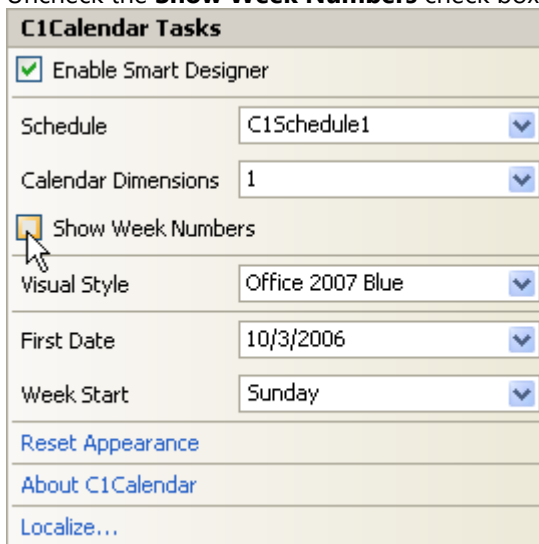


3. Close the dialog box.

In the Tasks Menu

To hide the week numbers, set the ShowWeekNumbers property to **False** in the Tasks menu:

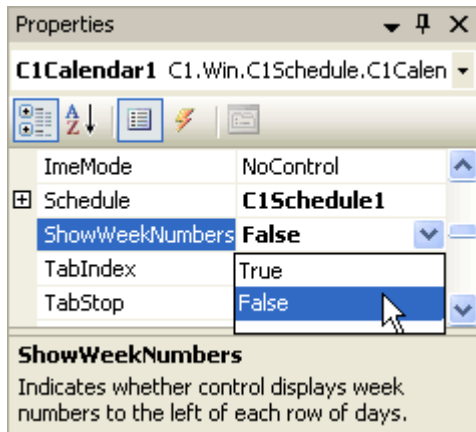
1. Open the **C1Calendar Tasks** menu. For more information on accessing the **C1Calendar Tasks** menu, see [C1Calendar Tasks Menu](#).
2. Uncheck the **Show Week Numbers** check box.



3. Close the **C1Calendar Tasks** menu.

In the Properties Window

To hide the week numbers, locate the ShowWeekNumbers property in the [C1Calendar](#) Properties window and set it to **False**:



In Code

To hide the week numbers, add the following code to the **Form_Load** event to set the [ShowWeekNumbers](#) property to **False**:

To write code in Visual Basic

Visual Basic

```
Me.C1Calendar1.ShowWeekNumbers = False
```

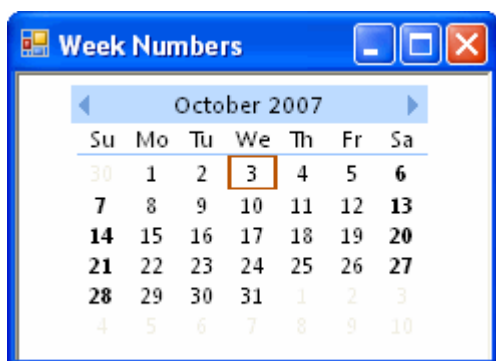
To write code in C#

C#

```
this.c1Calendar1.ShowWeekNumbers = false;
```


Run the program and observe:

When the ShowWeekNumbers property is set to **False**, the week numbers will no longer be visible:



Adding Importing and Exporting

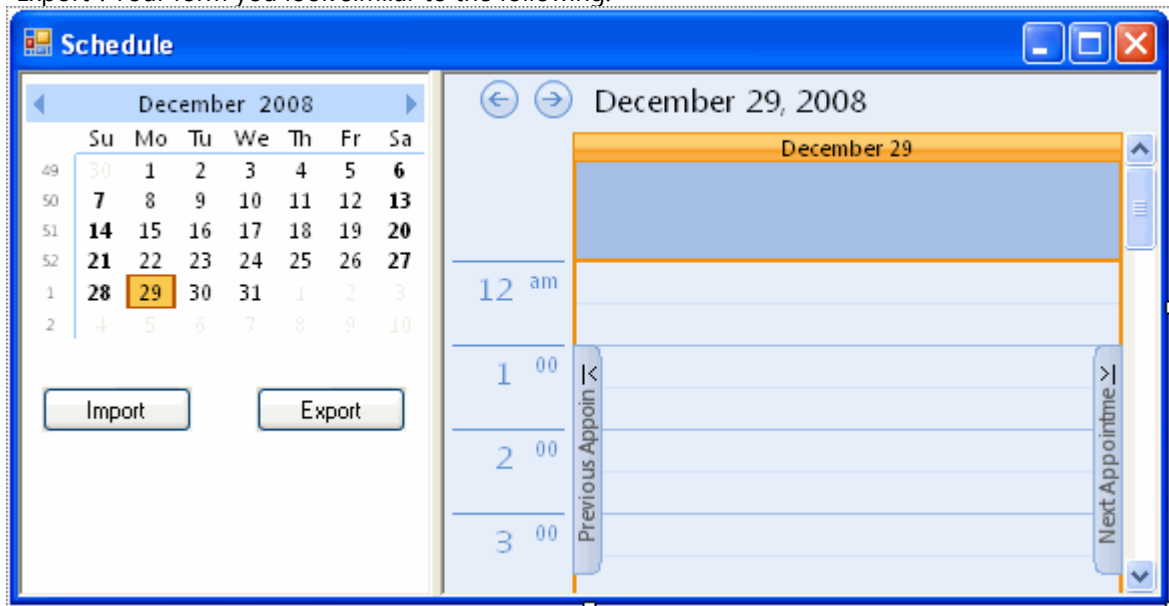
You can add importing and exporting functionality to your scheduling application using the [Import](#) and [Export](#) methods. The Import method opens the **OpenFileDialog** dialog box allowing you to import a XML, iCal, or binary (.dat) file. The Export method opens the **SaveFileDialog** dialog box allowing you to export a XML, iCal, or binary (.dat) file.

 **Note:** You can also use built-in importing and exporting features, see [Importing and Exporting Data](#) for details.

In the following example, [C1Calendar](#) and [C1Schedule](#) controls are added to the form along with **Import** and **Export** buttons. When the application is run, you'll be able to import and export data into the C1Schedule control.

Complete the following steps to add importing and exporting functionality to a scheduling application:

1. Create a new .NET application.
2. Resize the form, and from the Visual Studio Toolbox add a **SplitContainer** control to the form.
3. Click in the left pane of the **SplitContainer** and double-click the **C1Calendar** control in the Visual Studio Toolbox, to add it to the form.
4. In the Properties window, set **C1Calendar1.Dock** property to **Fill**.
5. Click in the right pane of the **SplitContainer** and double-click the **C1Schedule** control in the Visual Studio Toolbox, to add it to the form.
6. In the Properties window, set **C1Schedule1.Dock** property to **Fill**.
7. In the Properties window, set **C1Calendar1.Schedule** property to **C1Schedule1**.
8. Click in the left pane of the **SplitContainer** and from the Visual Studio Toolbox add two **Button** controls to the form on top of the C1Calendar control.
9. In the Properties window, set **Button1's Text** property to "Import" and set **Button 2's Text** property to "Export". Your form you look similar to the following:



10. Double-click the **Import** button. The Code Editor will open and the **Button1_Click** event will be added.
11. Add code to the **Button1_Click** event so that it looks like the following:

To write code in Visual Basic

Visual Basic

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    C1Schedule1.Import("Import a File")
End Sub
```

To write code in C#

C#

```
private void button1_Click(object sender, EventArgs e)
{
```

```
c1Schedule1.Import("Import a File");
}
```

This will enable importing files.

12. In Design view, double-click the **Export** button. The Code Editor will open and the **Button2_Click** event will be added.
13. Add code to the **Button2_Click** event so that it looks like the following:

To write code in Visual Basic

Visual Basic

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    C1Schedule1.Export("", "Export a File")
End Sub
```

To write code in C#

C#

```
private void button2_Click(object sender, EventArgs e)
{
    c1Schedule1.Export("", "Export a File");
}
```

This will enable exporting files.

Run your application and observe:

The **Import** and **Export** buttons will enable you to import and export appointments into your scheduling application.

1. Add a few appointments to the schedule.
2. Click the **Export** button. The **Export a File** dialog box will open enabling you to export a XML, iCal, or binary file. Export the file as an XML file and name it "myFile.xml".
3. Delete the appointments that you just added to the schedule.
4. Click the **Import** button. The **Import a File** dialog box will open enabling you to import a XML, iCal, or binary file.
5. Import the **myFile.xml** file you just saved and observe that the appointments you added are restored in the schedule.

Customizing Printing and Previewing

The following topics detail how to customize printing and previewing in **Scheduler for WinForms**. Printing in [C1Schedule](#) is based on using the C1.C1Preview.C1PrintDocument component and the C1.Win.C1Preview.C1PrintPreviewDialog form via reflection. For information about printing at run time see [Printing and Previewing Data](#).



Note: You must have build 2.0.20082.50035 or later of **Reports for WinForms** or **Preview for WinForms** for printing to function.

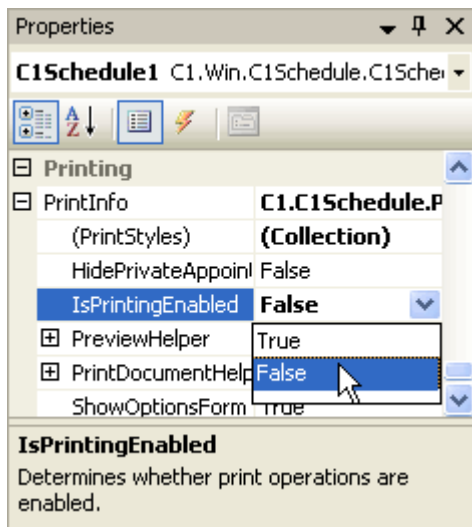
Disabling Printing

By default printing and previewing is enabled in the [C1Schedule](#) control. If you prefer that end-users not be able to print and preview schedule information or your application does not require printing functionality, you can disable printing by using the [IsPrintingEnabled](#) property. If [IsPrintingEnabled](#) is set to **False** the [C1Schedule](#) control will not try to find and load the **Preview for WinForms** assemblies and will not show printing and print preview options at run time in context menus and in the **AppointmentForm** toolbar.

To disable printing, set the [IsPrintingEnabled](#) property to **False** either in the Properties window or in code.

In the Properties Window

Locate the [IsPrintingEnabled](#) property in the [C1Schedule](#) Properties window and set it to **False**:



In Code

Add the following code to the **Form_Load** event to set the [IsPrintingEnabled](#) property to **False**:

To write code in Visual Basic

Visual Basic

```
Me.C1Schedule1.PrintInfo.IsPrintingEnabled = False
```

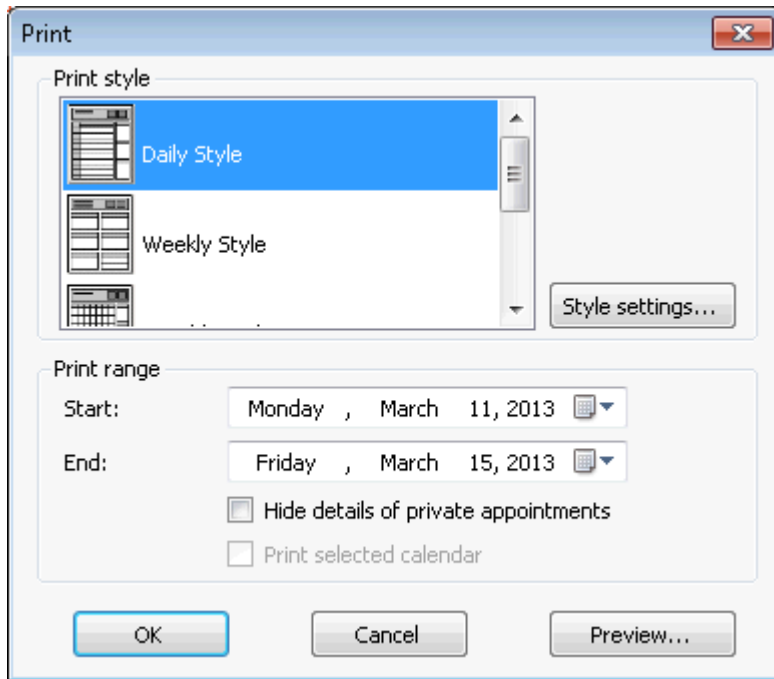
To write code in C#

C#

```
this.c1Schedule1.PrintInfo.IsPrintingEnabled = false;
```

Disabling the Print Options Form

By default, a **Print** dialog box displaying various options is displayed at run time before printing or previewing printing data in **Scheduler for WinForms**:

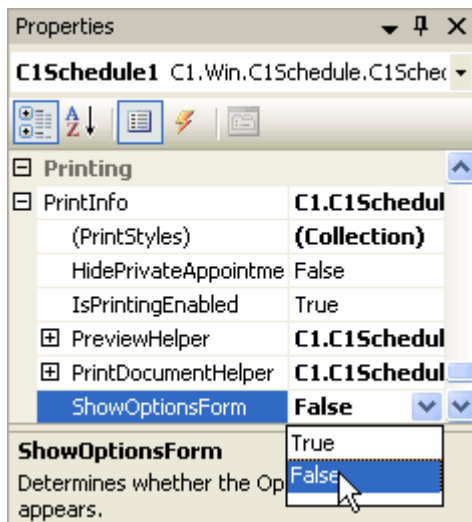


If you prefer that end-users not see this form, you can disable it using the [ShowOptionsForm](#) property. If ShowOptionsForm is set to **False** the [C1Schedule](#) control will not display the **Print** dialog box.

To disable the options form, set the ShowOptionsForm property to **False** either in the Properties window or in code.

In the Properties Window

Locate the ShowOptionsForm property in the C1Schedule Properties window and set it to **False**:



In Code

Add the following code to the **Form_Load** event to set the ShowOptionsForm property to **False**:

To write code in Visual Basic

Visual Basic

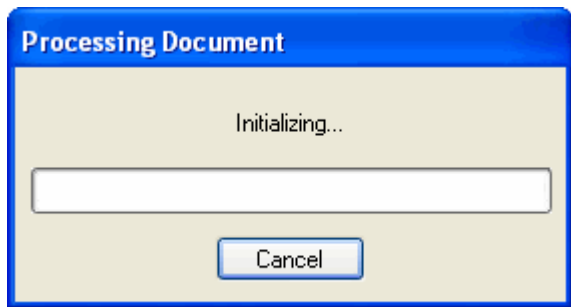
```
Me.C1Schedule1.PrintInfo.ShowOptionsForm = False
```


To write code in C#

```
C#
this.c1Schedule1.PrintInfo.ShowOptionsForm = false;
```

Disabling the Print Progress Form

By default, a **Processing Document** dialog box is displayed at run time before printing or previewing data in **Scheduler for WinForms**:

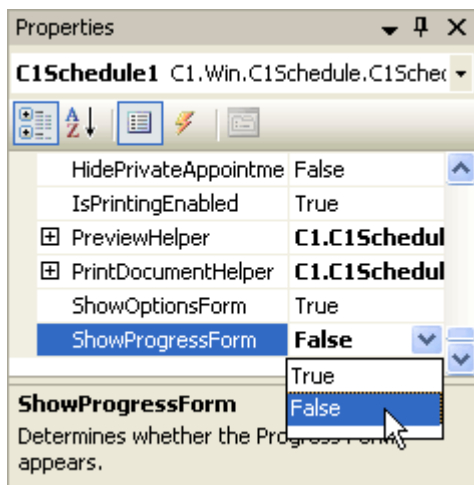


If you prefer that end-users not see this dialog box, you can disable it using the [ShowProgressForm](#) property. If [ShowProgressForm](#) is set to **False** the [C1Schedule](#) control will not display the **Processing Document** dialog box.

To disable the print progress form, set the [ShowProgressForm](#) property to **False** either in the Properties window or in code.

In the Properties Window

Locate the [ShowProgressForm](#) property in the [C1Schedule](#) Properties window and set it to **False**:



In Code

Add the following code to the **Form_Load** event to set the [ShowProgressForm](#) property to **False**:

To write code in Visual Basic

```
Visual Basic
```

```
Me.C1Schedule1.PrintInfo.ShowProgressForm = False
```

To write code in C#

```
C#
this.c1Schedule1.PrintInfo.ShowProgressForm = false;
```

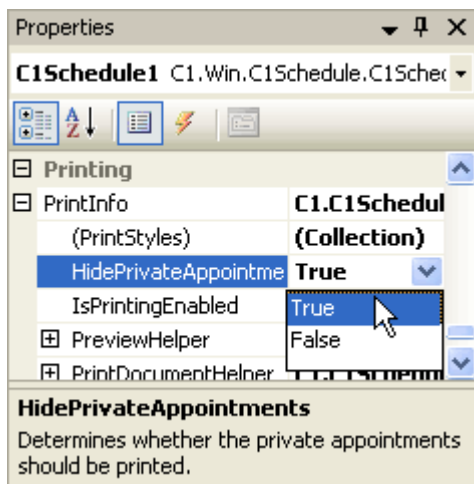
Hiding Private Appointments

By default, **Scheduler for WinForms** prints private appointments when data is printed. If you prefer that end-users not print private appointments, you can change this using the [HidePrivateAppointments](#) property. If [HidePrivateAppointments](#) is set to **True** the [C1Schedule](#) control will not print private appointments when printing the schedule.

To hide private appointments, set the [HidePrivateAppointments](#) property to **True** either in the Properties window or in code.

In the Properties Window

Locate the [HidePrivateAppointments](#) property in the [C1Schedule](#) Properties window and set it to **True**:



In Code

Add the following code to the **Form_Load** event to set the [HidePrivateAppointments](#) property to **True**:

To write code in Visual Basic

```
Visual Basic
Me.C1Schedule1.PrintInfo.HidePrivateAppointments = True
```

To write code in C#

```
C#
this.c1Schedule1.PrintInfo.HidePrivateAppointments = true;
```