

Restricting Dates

In most situations where you need to use a calendar for selection, you don't want to allow the user to select any date in the calendar.

For example, the user might be booking an appointment or choosing a delivery date—two services that are generally provided only on set days. The Calendar control makes it surprisingly easy to implement this logic. In fact, if you've worked with the date and time controls on the Windows platform, you'll quickly recognize that the ASP.NET versions are far superior.

The basic approach to restricting dates is to write an event handler for the `Calendar.DayRender` event.

This event occurs when the Calendar control is about to create a month to display to the user. This event gives you the chance to examine the date that is being added to the current month (through the `e.Day` property) and decide whether it should be selectable or restricted.

The following code makes it impossible to select any weekend days or days in years later than 2013:

```
protected void MyCalendar_DayRender(Object source, DayRenderEventArgs e)
{
    // Restrict dates after the year 2013 and those on the weekend.
    if (e.Day.IsWeekend || e.Day.Date.Year > 2013)
    {
        e.Day.IsSelectable = false;
    }
}
```

The `e.Day` object is an instance of the `CalendarDay` class, which provides various properties.

CalendarDay Property Descriptions

<code>Date</code>	The <code>DateTime</code> object that represents this date.
<code>IsWeekend</code>	True if this date falls on a Saturday or Sunday.
<code>IsToday</code>	True if this value matches the <code>Calendar.TodaysDate</code> property, which is set to the current day by default.
<code>IsOtherMonth</code>	True if this date doesn't belong to the current month but is displayed to fill in the first or last row. For example, this might be the last day of the previous month or the next day of the following month.
<code>IsSelectable</code>	Allows you to configure whether the user can select this day.