



## **Peter's Date Package v2.0**

### **What's New in Version 2**

Copyright © 2007, Peter L. Blum. All Rights Reserved

## Table of Contents

<b>INTRODUCTION</b>	<b>4</b>
<b>BREAKING CHANGES</b>	<b>4</b>
<b>CHANGES THAT AFFECT MANY CONTROLS</b>	<b>4</b>
<b>Shared Popups</b>	<b>4</b>
<b>Support for AJAX “Panels”</b>	<b>6</b>
<b>Popups can appear when the mouse points to the Toggle Button</b>	<b>6</b>
<b>Special Effects on popups and popdowns</b>	<b>7</b>
<b>Shadow effect on all popups</b>	<b>7</b>
<b>Separate style sheets for by type of control</b>	<b>8</b>
Design change – different style sheets by browser made obsolete	8
<b>Browser Sensitive Style Sheet Class Names</b>	<b>8</b>
<b>xDateTimeFormatInfo property always allows edits</b>	<b>8</b>
<b>URLs into the Appearance folder use the {APPEARANCE} token</b>	<b>9</b>
<b>Design Mode Improvements</b>	<b>10</b>
Expanded Properties Editor	10
Larger Editor for String Properties	11
Properties that should not be used are hidden from Intellisense	11
Visual Studio 2005 applies automatic style sheets in design mode	11
<b>NEW CONTROL: MULTISELECTIONCALENDAR</b>	<b>12</b>
<b>NEW CONTROLS: TIMEPICKER AND POPUPTIMEPICKER</b>	<b>13</b>
<b>DATETEXTBOX IMPROVEMENTS</b>	<b>14</b>
<b>New Date Formats</b>	<b>14</b>
<b>Smarter ShortDatePattern</b>	<b>14</b>
<b>Range Mode Improvements</b>	<b>14</b>
Auto fill in the end date of a range	14
Range minimum date	14
<b>Pop up the calendar when focus is enters the textbox</b>	<b>14</b>
<b>More flexible editing for ReadOnly textboxes</b>	<b>14</b>
<b>DateTextBoxCompareValidator now can compare to a specific date</b>	<b>15</b>
<b>The OnChange Function can be fired on any command</b>	<b>15</b>

<b>CS_CALENDAR IMPROVEMENTS</b>	<b>16</b>
<b>Multiple Months View</b>	<b>16</b>
<b>Show the date numbers in cells that are outside the min/max range</b>	<b>17</b>
<b>Hide Weekend Columns</b>	<b>17</b>
<b>Special effects on month changes</b>	<b>17</b>
<b>Several images available for month navigation buttons</b>	<b>17</b>
<b>Width of Calendar and Cells</b>	<b>17</b>
Adjusting cell widths	18
<b>Client-side script available on month view changes</b>	<b>18</b>
Related changes	18
<b>Keyboard commands are optional</b>	<b>18</b>
<b>Adding gaps around the calendar</b>	<b>19</b>
<b>Leveling week row heights</b>	<b>19</b>
<b>Breaking Change: Day Header formatting</b>	<b>19</b>
<b>SPECIALDATES CONTROL IMPROVEMENTS</b>	<b>20</b>
<b>MONTHYEARPICKER IMPROVEMENTS</b>	<b>21</b>
<b>Few Years Format</b>	<b>21</b>
<b>New Properties on the MonthYearPicker</b>	<b>22</b>
<b>Several images available for year navigation buttons</b>	<b>22</b>
<b>TIMEOFDAYTEXTBOX AND DURATIONTEXTBOX IMPROVEMENTS</b>	<b>23</b>
<b>TimePicker control</b>	<b>23</b>
<b>UnselectableTimesValidator</b>	<b>23</b>
<b>Other New Properties</b>	<b>23</b>

## Introduction

Peter's Date Package v2.0 is an extensive upgrade. There are new features introduced in nearly every control, and the introduction of several new controls, such as the MultiSelectionCalendar and TimePicker. This document will orient users of Peter's Date Package v1.x to these changes.

- ❖ To install this software into a web application that has Peter's Date Package v1.x, see the "Upgrading to Version 2" section of the **Installation Guide**.
- ❖ To install this software into a web application without Peter's Date Package, see the "First time installation" section of the **Installation Guide**.

Features that quickly enhance existing sites are marked ▲. This means that after installing you either immediately benefit from them or will benefit after a quick property change. Features without this marker may require adding controls or changing the design of your page.

## Breaking Changes

Here are the design changes that may break existing code. They are:

- These properties on **DateTextBox.xPopupCalendar** are obsolete: **xAutoSharedCalendarB**, **xSharedCalendarDateTextBoxID**, and **xSharedCalendarDateTextBox**. They have been replaced by **xSharedGroup** on **DateTextBox.xPopupCalendar.xCalendar**. See "Breaking Change – Obsolete properties".
- The CS\_Calendar provides a shadow effect in the CSCCalendar style sheet class. This has been replaced with a property, **xUseShadowEffectB**. The CSCCalendar style sheet class should be modified in your existing style sheet files. See "Breaking Change – Converting A Style Sheet from v1.x".
- The default image files for the month and year navigation buttons on CS\_Calendar and MonthYearPicker have changed. They used to default to LeftCmdBlue.PNG, LeftCmd2Blue.PNG, RightCmdBlue.PNG, and RightCmd2Blue.PNG. Now they are LeftCmd\_3DBlue.GIF, LeftCmd2\_3DBlue.GIF, RightCmd\_3DBlue.GIF, and RightCmd2\_3DBlue.GIF.

The new files have the same images as in version 1.1, except they now provide a transparent background, enabling them to work with non-white backgrounds.

This change will only affect you if you have edited the original graphic image files. To fix this, edit the URL in these properties to reflect the old filename:

In CS\_Calendar, PopupCalendar and DateTextBox.xPopupCalendar.xCalendar: **xNextMonthButtonImageUrl**, **xPrevMonthButtonImageUrl**, **xJumpForwardButtonImageUrl**, **xJumpBackButtonImageUrl**.

In MonthYearPicker and PopupMonthYearPicker: **xNextYearButtonImageUrl**, **xPrevYearButtonImageUrl**.

- The header that shows days of weeks above the calendar has had some restructuring. See "Breaking Change: Day Header formatting".
- When using a DateTextBox without a DateTextBoxValidator, but with a DateTextBoxMinMaxValidator or DateTextBoxCompareValidator, you may have found those two validators report errors when the date entered is invalid. That happens during server side validation only. It should not have worked that way. The DateTextBoxValidator should handle that exclusively. The software has been changed to use the DateTextBoxValidator for checking the date format while the other two will only report errors for what they were intended.

How does this affect you? Check for DateTextBoxes that do not have a DateTextBoxValidator but have one of the other two. Add the DateTextBoxValidator.

- When using the VAM Style Validators, they require VAM v3.0.4 or higher. You may need to get a service release or upgrade from <http://www.peterblum.com/vam/home.aspx>.

## Changes That Affect Many Controls

### Shared Popups

▲ When using two or more controls on a page that have popups, those popups (like context menu or popup calendar) usually have the same appearance and features. For example, the context menu of every DateTextBox almost always has the same

commands. The popup Calendar and popup MonthYearPicker almost always has the same appearance. With popups, since the control isn't shown until needed, one block of HTML can fully describe the popup control and the controls that invoke the popup can simply use that one block of HTML. Yet in version 1.x, these controls generated unique HTML and javascript for each popup.

Version 2.0 changes this dramatically. Every control that can popup has a new property called **xSharedGroup**. **xSharedGroup** is a name to associate a group of identical controls so one block of HTML is generated. **xSharedGroup** defaults to the empty string (""), which is a valid name. Therefore, all identical controls default to using HTML sharing. You explicitly have to supply a new name in **xSharedGroup** to let it generate its own HTML.

The **xSharedGroup** property is found on the panel that is popped up, not the toggle button or control that fires the popup. Use this table to identify a specific control and where its **xSharedGroup** property is located:

Popup Calendar on DateTextBox	DateTextBox.xPopupCalendar.xCalendar
Popup Calendar on AnniversaryTextBox	AnniversaryTextBox.xPopupCalendar.xCalendar
Popup MonthYearPicker on MonthYearTextBox	MonthYearTextBox.xPopupMonthYearPicker.xMonthYearPicker
Popup MonthYearPicker on CS_Calendar	CS_Calendar.xPopupMonthYearPicker.xMonthYearPicker.xMonthYearPicker DateTextBox.xPopupCalendar.xCalendar.xPopupMonthYearPicker.xMonthYearPicker
Popup MonthYearPicker on MultiSelectionCalendar	MultiSelectionCalendar.xPopupMonthYearPicker.xMonthYearPicker
Popup TimePicker on TimeOfDayTextBox	TimeOfDayTextBox.xPopupTimePicker.xTimePicker
Popup TimePicker on DurationTextBox	DurationTextBox.xPopupTimePicker.xTimePicker
ContextMenu on DateTextBox	DateTextBox.xContextMenu
ContextMenu on AnniversaryTextBox	AnniversaryTextBox.xContextMenu
ContextMenu on MonthYearTextBox	MonthYearTextBox.xContextMenu
ContextMenu on TimeOfDayTextBox	TimeOfDayTextBox.xContextMenu
ContextMenu on DurationTextBox	DurationTextBox.xContextMenu
ContextMenu on CS_Calendar	CS_Calendar.xContextMenu
ContextMenu on MultiSelectionCalendar	MultiSelectionCalendar.xContextMenu

Context menus are considered different for each control combination listed above. Therefore, DateTextBox cannot share its context menu with the AnniversaryTextBox, MonthYearTextBox, etc.

**Breaking Change – Obsolete properties**

The shared popup feature replaces these properties on **DateTextBox.xPopupCalendar**: **xAutoSharedCalendarB**, **xSharedCalendarDateTextBoxID**, and **xSharedCalendarDateTextBox**. Those properties still exist in the code so that you don't have any compiler errors. But they are merely placeholders and do not get used to setup sharing anymore.

When adding Peter's Date Package 2.0, you should review each page to see if there are multiple controls of the same type for which you created different popups. For context menus, this would be differences in the menu items. For popup Calendars and popup MonthYearPickers, this would be differences in their appearance. When differences are found, set the **xSharedGroup** name to a unique value for each case. For example, when you have two DateTextBoxes with differences in their context menu or popup calendar's appearance.

**Advantages of xSharedGroup over v1.x**

Immediately upon installing v2, all of your popups will employ sharing. This will immediately reduce the size and initialization time on pages that have multiple identical control types. If you removed context menus and popup MonthYearPickers to save page size, you may want to experiment by reenabling them. Yes, the page size will grow a bit, but with only one HTML block for each control type instead of as many controls as you have on that page.

Here are the typical properties from which to experiment:

- DateTextBox.xEnableContextMenuB
- AnniversaryTextBox.xEnableContextMenuB

- MonthYearTextBox.xEnableContextMenuB
- TimeOfDayTextBox.xEnableContextMenuB
- DurationTextBox.xEnableContextMenuB
- DateTextBox.xPopupCalendar.xCalendar.xEnableContextMenuB
- In Page\_Load ( ), sometimes users shut down the context menus or MonthYearPicker by using the PeterBlum.PetersDatePackage.TrueBrowser class. Here are the properties involved. *Note: These properties are global to your web application. It only makes sense to adjust them if you make those adjustments uniformly throughout your code.*
  - TrueBrowser.xSupportsContextMenuB
  - TrueBrowser.xSupportsMonthYearPickerB

In v1.1, while the DateTextBox's popup Calendar offered sharing, it could not be used when the DateTextboxes on the page had differences in these data oriented properties: **xTodaysDate**, **xSpecialDate**, **xMinDate**, **xMaxDate** or **xSpecialDatesControlID**. This restriction has been removed for v2.0. If any of those properties are different, you can still use the same name in **xSharedGroup**.

## Support for AJAX "Panels"

AJAX is a technology that allows modern browsers to communicate with your server side code without a postback. It typically is used to update values through DHTML – the object oriented framework of the browser. The page developer may replace the value of a single attribute on an HTML tag, such as the text shown on a textbox or replace a block the page's HTML.

AJAX is packaged within many third party products, including [Microsoft ASP.NET AJAX](#), [Microsoft Atlas](#), [Telerik RadControls](#) ("radCallback" and "radAJAX"), and [Magic AJAX](#). Each of these products has the capability to replace a block of the page's HTML with any web controls you desire. Peter's Date Package now supports this technology.

See the "Using these Controls with AJAX" in the **User's Guide**, page 407.

## Popups can appear when the mouse points to the Toggle Button

▲ All popups that use a toggle button, such as the popup Calendar and Help button, can automatically popup after the mouse points to them for a specified duration. Here are the properties that enable this feature and set its duration.

Control	Enable the feature	Set its duration
Popup Calendar on DateTextBox	DateTextBox.xPopupCalendar. xPopupOnMouseOverB	DateTextBox.xPopupCalendar. xPopupOnMouseOverDelay
Popup Calendar on AnniversaryTextBox	AnniversaryTextBox.xPopupCalendar. xPopupOnMouseOverB	AnniversaryTextBox.xPopupCalendar. xPopupOnMouseOverDelay
Popup MonthYearPicker on MonthYearTextBox	MonthYearTextBox.xPopupMonthYearPicker. xPopupOnMouseOverB	MonthYearTextBox.xPopupCalendar. xPopupOnMouseOverDelay
Help button on DateTextBox	DateTextBox.xHelpPopupOnMouseOverB	DateTextBox.xHelpPopupOnMouseOverDelay
Help button on AnniversaryTextBox	AnniversaryTextBox.xHelpPopupOnMouseOverB	AnniversaryTextBox.xHelpPopupOnMouseOverDelay
Help button on MonthYearTextBox	MonthYearTextBox.xHelpPopupOnMouseOverB	MonthYearTextBox.xHelpPopupOnMouseOverDelay
Help button on TimeOfDayTextBox	TimeOfDayTextBox.xHelpPopupOnMouseOverB	TimeOfDayTextBox.xHelpPopupOnMouseOverDelay
Help button on DurationTextBox	DurationTextBox.xHelpPopupOnMouseOverB	DurationTextBox.xHelpPopupOnMouseOverDelay
Popup TimePicker on TimeOfDayTextBox	TimeOfDayTextBox.xPopupTimePicker. xPopupOnMouseOverB	TimeOfDayTextBox.xPopupTimePicker. xPopupOnMouseOverDelay
Popup TimePicker on	DurationTextBox.xPopupTimePicker.	DurationTextBox.xPopupTimePicker.

DurationTextBox	xPopupOnMouseOverB	xPopupOnMouseOverDelay
Help button on CS_Calendar	CS_Calendar.xHelpPopupOnMouseOverB DateTextBox.xPopupCalendar.xCalendar. xHelpPopupMouseOverB	CS_Calendar.xHelpPopupOnMouseOverDelay DateTextBox.xPopupCalendar.xCalendar. xHelpPopupOnMouseOverDelay
Help button on MultiSelectionCalendar	MultiSelectionCalendar.xHelpPopupOnMouseOverB	MultiSelectionCalendar.xHelpPopupOnMouseOverDelay
Popup MonthYearPicker on CS_Calendar	CS_Calendar.xPopupMonthYearPicker .xPopupOnMouseOverB	CS_Calendar.xPopupMonthYearPicker .xPopupOnMouseOverDelay

## Special Effects on popups and popdowns

▲ Internet Explorer provides some nifty special effects they call [filters and transistions](#). These are non-standard style sheet items. The version 1.x CS\_Calendar has already been using two of them to form its shadow border. Now all popups support special effects that are invoked when popping up and down. The effects default to a quick fade in and fade out, using the filter called “[Fade](#)”.

This feature is enabled by default (and causes a very small script file, filter.js, to be loaded).

See “Customizing The Popup Effect” in the **User’s Guide**, page 433.

## Shadow effect on all popups

▲ The shadow effect on the popup calendar has been improved and is now available as a property setting on all popups. It is shown by default. In addition, it handles the shadow effect better. Previously, the shadow could blend into the actual popup when it was taller or wider than 200 pixels. Now it uses the size of the popup control to determine the shadow size. (It still may blend a bit, but far less than before.)

If you want to remove the effect on individual controls, change their **xUseShadowEffectB** property to `false`. This property is found on the toggle control, such as `DateTextBox.xPopupCalendar` and `CS_Calendar.xPopupMonthYearPicker`.

```
DateTextBox.xPopupCalendar.xUseShadowEffectB = false
```

```
CS_Calendar.xPopupMonthYearPicker.xUseShadowEffectB = false
```

## Breaking Change – Converting A Style Sheet from v1.x

In Peter’s Date Package v1.x, the shadow effect came from the style sheet file, in the this style sheet:

```
.CSCCalendar
{
    border-right: lightgrey ridge;
    border-top: lightgrey ridge;
    font-size: 8pt;
    filter: progid:DXImageTransform.Microsoft.dropshadow(OffX=3, OffY=3, Color=
        'gray' , Positive= 'true' ) progid:DXImageTransform.Microsoft.Alpha(
        Opacity=100, FinishOpacity=20, Style=1, StartX=96, FinishX=100, StartY=0,
        FinishY=0) progid:DXImageTransform.Microsoft.Alpha( Opacity=100,
        FinishOpacity=20, Style=1, StartX=0, FinishX=0, StartY=97, FinishY=100);
    border-left: lightgrey ridge;
    border-bottom: lightgrey ridge;
    font-family: Arial;
    background-color: white;
}
```

Notice that filter style. You should remove it in v2.0 to allow the **xUseShadowEffectB** property to do the work. You have the option of retaining it in the style sheet file, but only if you set

**DateTextBox.xPopupCalendar.xCalendar.xUseShadowEffectB** and **CS\_Calendar.xUseShadowEffectB** to `false`.

In the previous version, the `CSCCalendar` style sheet’s filter interfered with `SmartNavigation` and `IFrames` forcing users to set the `CssClass` property to an alternative style sheet, `CSCCalendarNoFilter`. While that style remains for backwards compatibility, it is no longer necessary. You can simply set **xUseShadowEffectB** to `false`.

## Separate style sheets for by type of control

In previous versions, you had a single style sheet file that contained all style sheet classes. When you were not using all controls, those styles would still be downloaded. With many new styles introduced in this version, that single file will have even a larger impact on transmission times.

In this version, you can either use one style sheet file or separate files based on the type of control. The controls will automatically add their own style sheet file's `<link>` tag to the page, if you are using ASP.NET 2.0 with `<head runat='server'>` or add this line to the `<head>` tag in any version of ASP.NET:

```
<%= PeterBlum.PetersDatePackage.CommonFunctions.GetStyleSheetLinkTags(Page) %>
```

This change will not break your existing sites. The upgrading instructions will show you how to setup your existing style sheet file so it continues to work. It will also show you how to switch to this new feature.

*Note: Peter's Date Package v2 provides different style sheet file names. It will not overwrite your existing style sheets, to allow for an easy upgrade.*

See "Using Style Sheets" in the **User's Guide**, page 435.

## Design change – different style sheets by browser made obsolete

Version 1.x provided you with three style sheet files: "StyleSheet1.css", "NS7Stylesheet.css", and "OperaStyleSheet.css". They can be used to handle the differences in browsers. Version 2 does not need these separate files. Instead, it allows different class names in the same file to handle a single style sheet property based on the browser. This is called "Browser Sensitive Style Sheet Class Names" (see the next topic.)

Version 2 preserves the built-in code that uses these old style sheets. If you want to use this built-in code, add this key to the web.config file:

```
<configuration>  
  <appSettings >  
    <add key="PDP_UseVersion1StyleSheet" value="" />  
  </appSettings>  
</configuration>
```

## Browser Sensitive Style Sheet Class Names

When you assign a style sheet class name to any of Peter's Date Package properties, you should expect that same name to be outputted to the HTML. Not every style sheet class that you define works for every browser. For example, on Netscape 7.0x browsers, there was a bug that showed border lines on tables even when the table was invisible. The best solution for that bug was to provide alternative style sheet classes for Netscape 7 that omitted the border lines. In Peter's Date Package version 1.x, this was handled by having separate style sheet files. The default was StyleSheet1.css. For Netscape 7.0, you used NS7StyleSheet1.css. For Opera 7, you used OperaStyleSheet1.css.

In Version 2, you can keep alternative style sheet classes in the same file as the default classes. With the Browser Sensitive Style Sheet feature, Peter's Date Package will switch the class name depending on the browser. This feature is very powerful, giving you the ability to map any style sheet class name to another name. For example, Peter's Date Package automatically switches "CSCWeekRowsTable" to "CSCWeekRowsTable\_NS70" on CS\_Calendar when it detects Netscape 7.0.

*Note: Numerous small changes have been made to the classes of the new style sheet files that solve some problems found by users in v1.x. This includes solutions that remove the need for separate classes on Opera 7 browsers. If you continue to use the version 1 style sheet files, it will continue to load the Netscape 7 and Opera 7 style sheet files.*

To use this feature, see the topic "Browser Sensitive Style Sheet Class Names" in the **User's Guide**, page 439.

## xDateTimeFormatInfo property always allows edits

In past versions, if you wanted to edit a property on the `xDateTimeFormatInfo` property of these controls, you would have to create a new instance of the `System.Globalization.DateTimeFormatInfo` class. This is no longer necessary.

## ***URLs into the Appearance folder use the {APPEARANCE} token***

▲ In Version 1.1, any property that specified a URL to the **Appearance** folder would default to “/aspnet\_client/PetersDatePackage/Appearance/”. Since ASP.NET 2.0 came out, Peter’s Date Package gives users the ability to easily locate the **PetersDatePackage** folder in the web application folder instead of the **/aspnet\_client** folder.

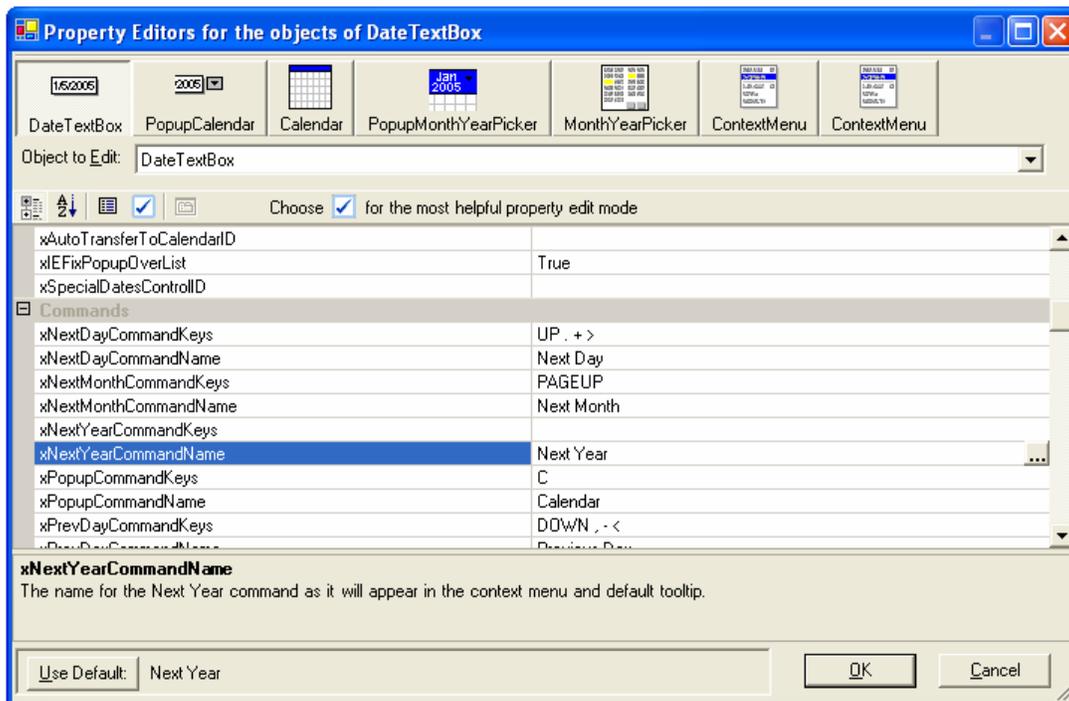
As a result, all default property values for URLs will start with “{APPEARANCE}/”. The {APPEARANCE} token will be replaced by the path known to Peter’s Date Package for the **Appearance** folder. It may be in the **[web app]/PetersDatePackage** folder, **[domain root]/aspnet\_client/PetersDatePackage/**, or a path you specify using a key in the <appSettings> section of the **web.config** file.

This change should not affect you. The software will still convert any URL starting with “/aspnet\_client/PetersDatePackage/Appearance/” to the correct location of the **Appearance** folder.

## Design Mode Improvements

### Expanded Properties Editor

▲ In design mode, the Properties Editor is used to change property values. Because many Peter's Date Package controls are really several controls combined together, an **Expanded Properties Editor** is now available. You can select it from the context/task menu on the control, from the Properties Pages icon at the top of Visual Studio.net's Properties Editor, and from a link at the bottom of Visual Studio.net's Properties Editor.



The toolbar and dropdownlist at the top let you switch to the various controls that are part of this control. Shown above is the DateTextBox. If you notice two Context Menus, it's because both the DateTextBox and its CS\_Calendar provide Context Menus.

The most powerful feature of this window is the  "Best Order" button immediately above the property names. Click it to have the properties and categories listed in the order recommended by PeterBlum.com for setting up the control.

Once the Best Order button is selected, the toolbar looks like this:



All properties have been assigned one of these states: Required, Recommended, Sometimes used, or Rarely used. Click the **Recommended Properties** button to show only the Required and Recommended properties. This is a very good way to setup a newly added control. If you want to customize which states are shown, use the **Properties Assistant** button.

Finally, along the bottom is the **Use Default** button and the default value shown to its right. This is a quick way to restore the individual property to its default.

## Larger Editor for String Properties

▲ Most String Properties now offer a modal dialog box editor from any Properties Editor. It makes editing easier for large strings. It also offers a button to restore the string to its default.



To open this:



## Properties that should not be used are hidden from Intellisense

▲ When subclassing controls, properties from the base class may no longer be used or appropriate. These properties are no longer listed in Intellisense.

## Visual Studio 2005 applies automatic style sheets in design mode

▲ When using Peter's Date Package ability to automatically add style sheets to the page, design mode did not apply the style sheets. Visual Studio 2005 users now will see their pages using the style sheets in design mode.

## New Control: MultiSelectionCalendar

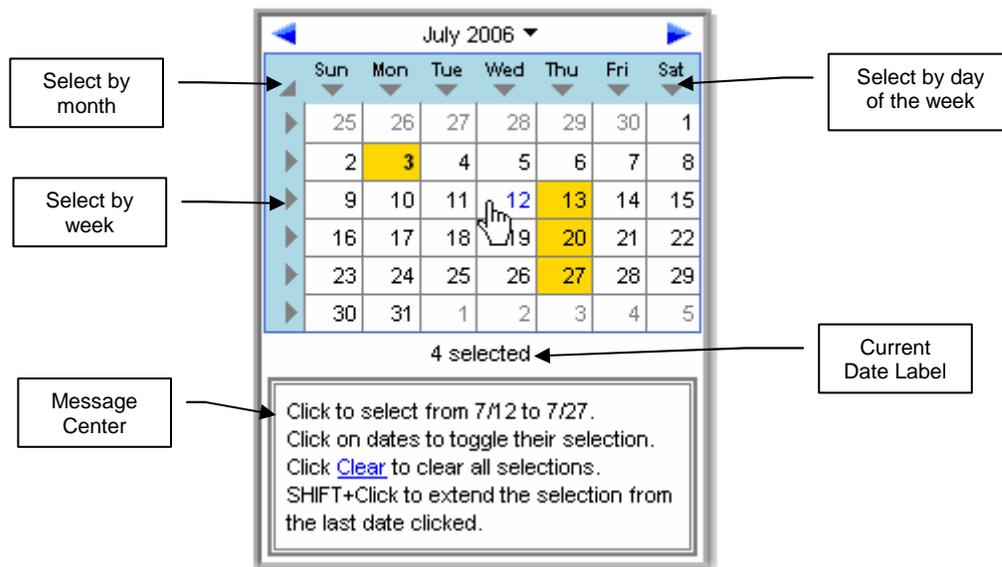
The CS\_Calendar control has a companion, MultiSelectionCalendar, which (obviously) allows multiple date cells to be selected. The two controls are very similar and share most of the same properties.

*FYI: While many calendar control developers merge a single selection and multi-selection feature into one calendar control, two separate controls allows more clarity on the purpose of properties designed around the selected values.*

If you know how to use the CS\_Calendar (documented in the **User's Guide**), you already know most of how to use the MultiSelectionCalendar.

Here are the main features of this control:

- Uses most of the same properties from the CS\_Calendar control including SpecialDates with its formatting features and Multiple Month View.
- There are several selection modes:
  - Individual dates. When turned off, you can prevent selecting single dates. This lets you use the week, month, and day of week features exclusively.
  - By week
  - By month
  - By day of the week
  - Range. Keeps a contiguous selection between the earliest and latest dates.
  - Use the SHIFT key to extend the selection from the last cell selected.
- You can limit the number of dates selection and show an alert when that limit is exceeded.
- The Current Date Label can show the number of dates selected or the start and end dates.
- On challenge users face with most multiple selection calendars is how to interact with it. This control introduces the **Message Center**, an interactive help section at the bottom of the control.



See the section "MultiSelectionCalendar Control" in the **User's Guide**, page 249.

## New Controls: TimePicker and PopupTimePicker

The TimePicker lets the user pick from a table of predefined times. You can use time of day or duration values. All you need to do is set the format in **xTimeFormat**. While time of day cannot exceed 24 hours, durations can.

The TimePicker is available in several forms. It is built into the TimeOfDayTextBox and DurationTextBox, shown when you set **xShowTimePickerB** to `true` on those controls. This is the most common use for it. It is also available as a separate control in popup or always shown form. You use the PopupTimePicker and TimePicker controls for these separate controls.



TimeOfDayTextBox with a TimePicker popped-up

The power of the TimePicker is two-fold. It lets the user pick from a visual display of times, and it can define a restricted list of the times you allow. The second case is important. Perhaps you will use it to offer times of appointments, which are separated by 30 minutes and skip your lunch period. You may even lock out editing the actual textbox by setting the **ReadOnly** and **xReadOnlyAllowsEditsB** properties to `true`. Since you define the exact times it offers, how you use it is very flexible.

When used with the TimeOfDayTextBox and DurationTextBox, a separate validator control is available: the UnselectableTimesValidator. It reports an error when the textbox contains a time that is not in the list of times you have defined.

See the section "TimePicker and PopupTimePicker" in the **User's Guide**, page 365.

For setting it up on the TimeOfDayTextBox and DurationTextBox, see the topic "Popup TimePicker" in the section "Using the TimeOfDayTextBox" in the **User's Guide**, page 322.

## DateTextBox Improvements

*Includes the AnniversaryTextBox and MonthYearTextbox.*

### New Date Formats

▲ In previous versions, you could only use the date format described by the **DateTimeFormatInfo.ShortDatePattern**. A **ShortDatePattern** separates each component of the date with the **DateSeparator**, like 05/31/2006. Even with month names shown (using **xAllowMonthNames**), the text remained a **ShortDatePattern**.

Now you can use the **LongDatePattern**, either with abbreviated or full month names. (The software still does not support the day of the week component, which is automatically ignored if found in the **LongDatePattern**.) Here's how to set this up:

- Make sure the **DateTextBox.xDateTimeFormatInfo.LongDatePattern** property reflects the desired pattern.
- Set the new **xDateFormat** property to either **Abbreviated** or **Long**.

See the section "Formatting And Parsing Text" in the **User's Guide**, page 27.

### Smarter ShortDatePattern

▲ While the **xAllowMonthNames** property still can be used to introduce an abbreviated month name into the **ShortDatePattern**, now you can use the text "MMM" inside the **ShortDatePattern**. This is a more .net-like way to setup the abbreviated month name.

### Range Mode Improvements

Range mode is when you have two DateTextBoxes and assign the **xRangeEndControlID** of the first one to the ID of the second one. That limits the second DateTextBox to be greater or equal to the first.

See the section "Date Range Properties" in the **User's Guide**, page 45.

### Auto fill in the end date of a range

▲ The end DateTextBox can automatically fill in with a date, when it is blank and the user has entered a start date. Just set **xRangeFillInBlankB** to **true**. It will use the start date + the number of days in **xRangeMinDays**. It also works in reverse, where you fill in the end date and it fills in the start DateTextBox.

### Range minimum date

▲ Normally when range mode is setup, if the user edits the end date to a value below the start date, the start date will be updated to a value below or equal to the end date. You can prevent the user from entering a date in the end date that is below the start date. This effectively imposes a minimum date on the end date textbox based on the start datetextbox's value + **xRangeMinDays**.

Set **xRangeEndSetsMinDateB** to **true** and provide an error message for an alert box in the **xRangeMinErrorMsg** property.

### Popup the calendar when focus is enters the textbox

▲ Set **xAutoPopupOnFocusB** to **true** when you want the calendar to popup as the user sets focus to the DateTextBox. Focus is set when the user tabs into the textbox or clicks in it.

*Note: You can also have the calendar autopopup when the user moves the mouse over the calendar toggle button. See "Popups can appear when the mouse points to the Toggle Button".*

### More flexible editing for ReadOnly textboxes

▲ Normally the **ReadOnly** property on a TextBox prevents edits. In previous versions, you could allow the popup calendar to still change the date by setting **xReadOnlyAllowsPopupCalendarB** to **true**. This concept has been expanded to allow editing through the context menu and help button, which offer commands like Next Day, Next Month, and Today.

The **xReadOnlyAllowsPopupCalendarB** property has been replaced by **xReadOnlyAllowsEditsB**. You do not have to change any code because the **xReadOnlyAllowsPopupCalendarB** will get and set the value of **xReadOnlyAllowsEditsB**.

If you do not want the user to use the context menu or help button, turn off those features by setting **xEnableContextMenuB** to `false` and **xShowHelpButton** to `false`.

### ***DateTextBoxCompareValidator now can compare to a specific date***

The `DateTextBoxCompareValidator` always compared one `DateTextBox` to another. Now it can also compare the `DateTextBox` to a specific date. Just set a `DateTime` object to its **xDateToCompare** property and do not use the **xControlToCompare** property.

### ***The OnChange Function can be fired on any command***

▲ If you use the **xOnChangedFunctionName** property to run javascript after the user makes an edit, you can now have it also run after commands like Next Day, Next Month, and Today. Just set **xCommandsFireOnChangeFunctionB** to `true`.

## CS\_Calendar Improvements

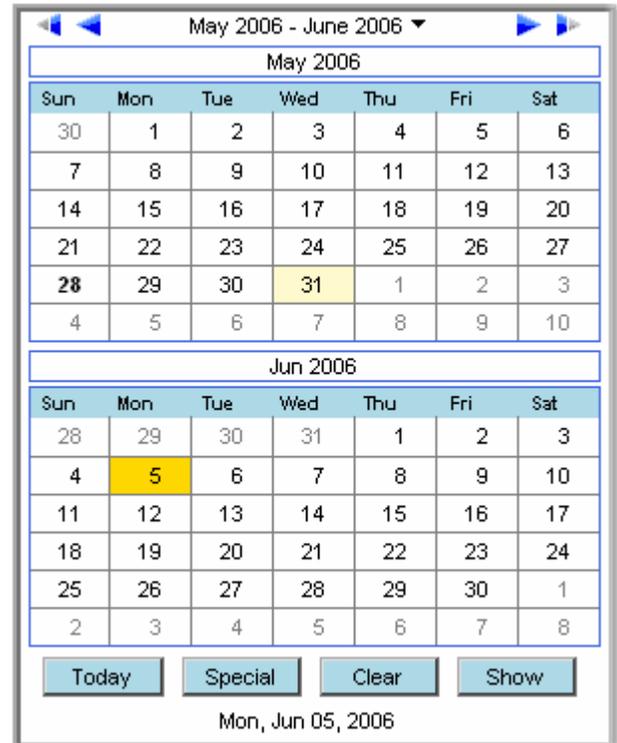
### Multiple Months View

▲ CS\_Calendar, the popup calendar of the DateTextBox, and MultiSelectionCalendar now allow you to view more than one month at a time. You define the number of rows and columns containing months in the **xMultiMonthRowCount** and **xMultiMonthColumnCount** properties. Months are shown sequentially. When the user uses the Next or Previous Month command, it jumps by the total number of months shown.

This mode allows quickly selecting a date amongst several months without moving to a different month. However, it increases the HTML downloaded to the browser, equal amounts for each additional calendar shown. So while its possible to show 12 months, be prepared for it to take much longer to load and initialize such a large calendar.



Two columns



Two rows

Looking at the above graphics, you will notice that the CS\_Calendar retains its general structure. Its header, showing the month movement commands and current month, has only changed to include the range of months shown. Its footer, shown here with most of the optional command buttons and the current date label, is the same as before. Only the area where the actual month calendar – day headers and week rows, has changed. It fits multiple months in that part and includes a new header above each month reflecting the month shown below it.

Use these properties to establish Multiple Months View:

<b>xMultiMonthRowCount</b>	The number of months shown stacked top-to-bottom. You can show between 1 and 6 months.
<b>xMultiMonthColumnCount</b>	The number of months shown side-by-side. You can show between 1 and 6 months.
<b>xMultiMonthOrdering</b>	Determines if sequential months are shown top-to-bottom or left-to-right first. For top-to-bottom, set the value to <code>Columns</code> . For left-to-right, set the value to <code>Rows</code> . It defaults to <code>Rows</code> .

There are also properties to establish the appearance. For details on all properties, see the “Multiple Month View Properties” topic in the “CS\_Calendar Properties” section of the **User’s Guide**, page 159.

## Show the date numbers in cells that are outside the min/max range

▲ In the past, date cells outside of **xMinDate** and **xMaxDate** properties were blank. Now you can show the dates by setting **xShowOutsideRangeDaysB** to `true`. The dates will not be selectable and will use a different style sheet, "CSCOutsideRangeDay" (or whatever you assign to **xOutsideRangeCSSClass**.)

See the section "Week Rows Properties" in the **User's Guide**, page 138.

## Hide Weekend Columns

▲ You can hide the two date columns of the weekend by setting **xShowWeekendDaysB** to `false`.

## Special effects on month changes

▲ By default, this upgrade provides some special effects to CS\_Calendar on Internet Explorer browsers. By using IE's [filters and transitions](#) feature, the calendar is animated as the user changes months. There are three animations: next month, previous month, and jump more than one month.

In addition, the month being replaced changes color to make the animation more visual. The default is the web color named "snow", an off-white.

It is on by default. To disable it, set **xUseMonthChangeEffectB** to `false`. When using `DateTextBox`, its **DateTextBox.xPopupCalendar.xCalendar.xUseMonthChangeEffectB**.

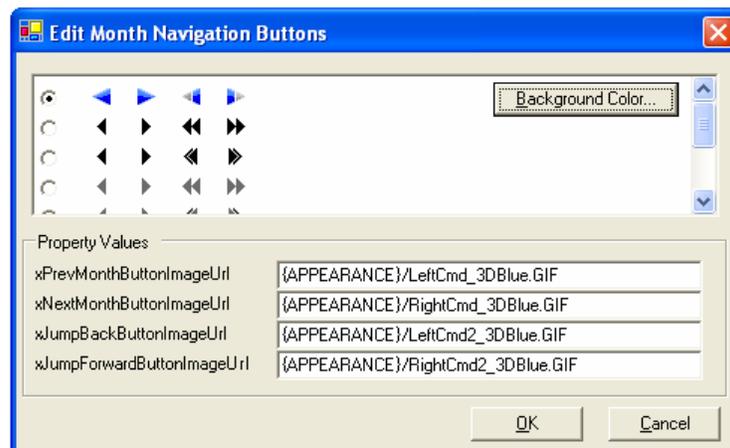
*Note: There are also effects to set the shadow and fade in/out on popup and popdown. See "Shadow effect on all popups" and "Special Effects on popups and popdowns".*

See the section "Special effects on month changes" in the **User's Guide**, page 130.

## Several images available for month navigation buttons

Version 1.x provided one set of images for the Month Navigation buttons. They are: 

Version 2 includes several additional sets of images. It also provides a design mode editor to pick them. Select the **Edit Month Navigation Buttons** command from the control's action menu or at the bottom of the Properties Editor.



## Width of Calendar and Cells

▲ In version 1.x, the width of the calendar worked best when the **Width** property was assigned to a value. As you added the week number column or SpecialDates data into the cells, that width remained unchanged, making the day cells too small. You would be forced to enter a different Width.

In version 2, the Width property can be left unassigned (which is the default setting). It will automatically calculate from a variety of elements, including the presence of the week number column, the size of text in the day header, and formatting rules for SpecialDates.

If your calendar's Width was never assigned, it will start to work this way, which means your calendar may be a different width than before.

*Note: If the Width property is assigned, it is best to set it to a pixel value. That way, the calculations can determine the day cell width and week number column width properly. Pixel values are represented by a string in this format: ##px. For example, 100px.*

## Adjusting cell widths

▲ There are two new properties to control the width of day cells and the week number column.

<b>xDayCellWidth</b>	The width of each day cell, in pixels. Leave it at 0 to have it automatically calculated. Automatic calculations take into account the size of day of weeks shown in the day header, <b>xWeekRowsCellSpacing</b> , and formatting from SpecialDates that you setup in the <b>xDayCellFormatter</b> property. Edit it if the calculated value is not right for you. Note that this is actually a minimum width. Browsers expand table cells to accommodate their contents. This is especially likely when using SpecialDates to provide formatting within a day cell.
<b>xWeekNumberCellWidth</b>	The width of the week number column, in pixels. Leave it at 0 to have it automatically calculated. Automatic calculations take into account <b>xShowWeekNumbersB</b> , the size of text in <b>xWeekNumberColTitle</b> , and toggle buttons from the MultiSelectionCalendar. Edit it if the calculated value is not right for you.

## Client-side script available on month view changes

You can add some javascript to the page that the calendar will fire on a month view change (including after the PopupMonthYearPicker is used). Put your script into the **xOnMonthViewChangedScript** property. Do not add the "javascript:" leader.

You can use these new client-side functions to return the month and year currently viewed, just by passing the **ClientID** of the CS\_Calendar to them.

```
function CSC_GetViewedMonth(clientID);
```

```
function CSC_GetViewedYear(clientID);
```

### Example

Uses the **xOnMonthViewChangedScript** to show an alert with the current month and year in the message:

```
Calendar1.xOnMonthViewChangedScript =  
  "alert('Month:' + CSC_GetViewedMonth('"+ Calendar1.ClientID +  
  "') + ' Year:' + CSC_GetViewedYear('"+ Calendar1.ClientID + "'))";
```

## Related changes

In past versions, the **xAutoPostBackOnMonthChangeB** did not work when the user used any commands that change the month viewed, like Next Month and Next Year. Now it works. It takes advantage of **xOnMonthViewChangedScript**. So you cannot use **xOnMonthViewChangedScript** when using **xAutoPostBackOnMonthChangeB**.

For consistency, the property **xOnSelectionChanged** has been renamed to **xOnSelectionChangedScript**. You will not have to make any changes as the old property remains, depreciated, and sets the new one automatically.

## Keyboard commands are optional

▲ The CS\_Calendar has always offered keyboard commands on Internet Explorer so long as the control as focus. Now you can reduce the amount of javascript loaded by turning off the keyboard commands. Set **xKeyboardControlB** to *false*. The file **CalKeyboard.js**, which contains this code, will no longer load into the page. Additionally, for browsers that don't support keyboard commands, this code is never loaded.

## ***Adding gaps around the calendar***

▲ If you want to add some space between the calendar's border and the contents of the calendar, use these properties: **xSideGaps**, **xTopGap**, and **xBottomGap**. Assign a value in pixels.

## ***Leveling week row heights***

▲ When using the SpecialDates control, you can fill in the contents of a day cell with HTML that may cause the row height to increase. Suppose the default height of a day cell is 20 pixels and one row has enough HTML to expand its height to 100 pixels. This makes for very uneven rows. To compensate, CS\_Calendar uses the height of the first row on the first month viewed to apply to all other rows. This can leave rows too big. Use the **xWeekRowsAutoCellHeightLimit** to establish the maximum size for this auto calculation.

## ***Breaking Change: Day Header formatting***

The day header - where day of week names appear - was a separate table, above the week rows in version 1.x. This caused problems where the day header cells could be misaligned with week row cells, especially when special dates was formatting the day cell.

The remedy is to make the day header the first row of the week rows table. It works very similarly, except it has these visual changes:

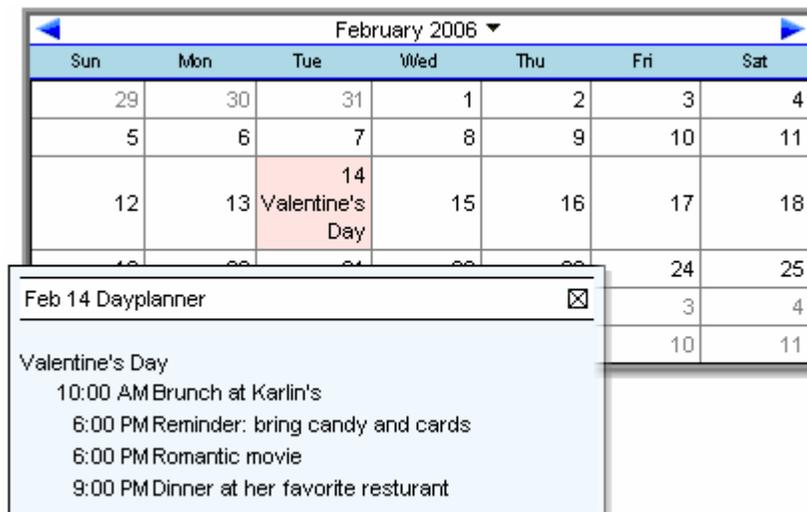
- It can no longer have a solid background when **xWeekRowsCellSpacing** is used. You will see gaps showing the background color of the week rows table, where the cell spacing is shown.
- It cannot show a bottom border because a bottom border is eliminated when the style border-collapse = collapse. That setting is used when **xWeekRowsCellSpacing** = 0.

As a result, a few style sheet classes have been changed. There is now a border around the entire table, not just the week rows. There is no border below the day header. These styles are addressed in the upgrading documentation of the **Installation Guide**.

## SpecialDates Control Improvements

The SpecialDates control has some of the most powerful improvements. They are summarized here.

- It supports multiple entries on the same date. This is usually multiple times in one day, although you might have a list of events that share a common time. These are called **Time Rows**. It can also merge the descriptions of from multiple entries on the same date if you prefer not to separate the data between date and a list of events.
- The **xDates** collection can be filled in by a Data Source. Time Rows can be read in through the same data source.
- There are simplified methods on the SpecialDates object to populate the **xDates** collection without dealing with all of those objects. In fact, the User's Guide doesn't talk much about creating SpecialSingleDate or the other objects. It has users take advantage of methods like `AddDate()` and `AddDateRange()`. Your code will continue to work, but future code may use the new methods.
- `CS_Calendar` can display the date entries using a several predefined formats. There is a format to quickly show a list of time elements. Plus there is a format to give you exact control over the look of the list of time elements. `CS_Calendar` can also popup an expanded description of the date as the mouse passes over a date cell. With this feature, you might provide a very simple description in the date cell so the cell stays small, and let the popup show the full details of that date.



- You can have all dates not defined in the SpecialDates control be considered unselectable by setting **SpecialDates.xUnknownDatesAreUnselectableB** to `true`.
- Client-side data is far more compact, reducing the the transfer times (including on AJAX-style callbacks which are already in the ASP.NET 2.0 enhanced assemblies of this product.)
- The **xDates** collection can be cached automatically so you don't have to populate it unless different data is needed.

See the section "SpecialDates Control" in the **User's Guide**, page 198.

## MonthYearPicker Improvements

The MonthYearPicker has several improvements. The first is most interesting, an alternative style designed around faster entry at the expense of a smaller list of available months to choose from.

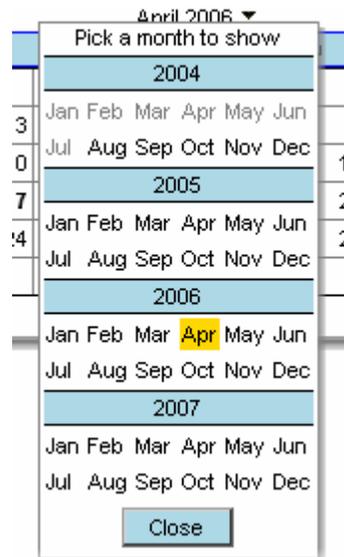
### Few Years Format

▲ The MonthYearPicker now has two formats, selectable by the **xFormat** property. They are:

- **ManyYears** – This is the original format. It allows a very large date range because the user can scroll through the years area to find the right year.
- **FewYears** – This new format shows every possible month in one view. As a result, you use it when there is only a short list of months, perhaps 1-5 years worth. It provides a one-click selection too, as compared to 3 or more clicks on the ManyYears format. It does not support keyboard selection though.

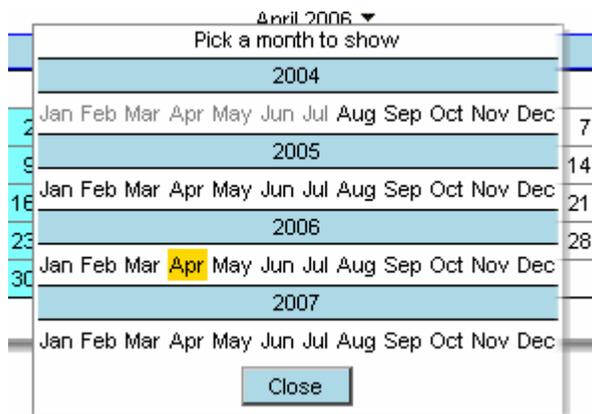


ManyYears format

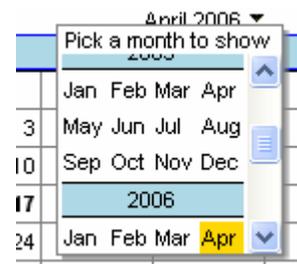


FewYears

In the above examples, the FewYears format takes up more screen real-estate. It generally will, to show all months and years. You can control the number of columns it shows, from 3 to 12, and add a vertical scrollbar.



12 columns



4 columns with scrollbars shown

The FewYears mode can have a lighter footprint on the page outputted. It's javascript is much smaller. If you show only 24 months, its HTML is smaller too. So it's a good choice when trying to reduce the transfer time of your page.

See the "Setting Few Years Format" topic in the "MonthYearPicker and PopupMonthYearPicker Controls" section of the **Users Guide**, page 274.

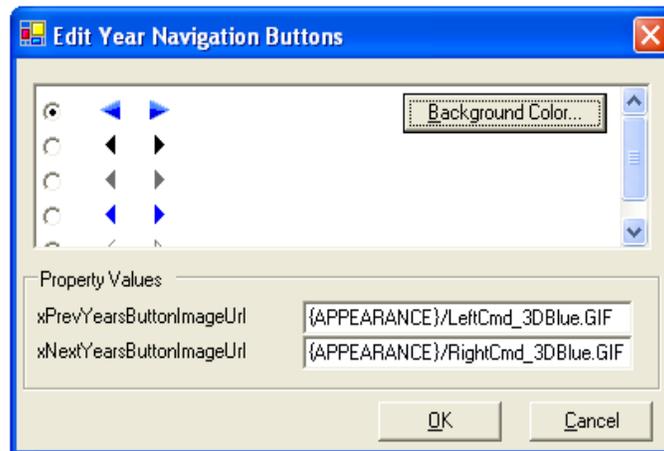
## New Properties on the MonthYearPicker

<b>▲ xFormat</b>	Determines the format. Choose one of these <code>PeterBlum.PetersDatePackage.MonthYearPickerFormat</code> choices: <ul style="list-style-type: none"><li>• <code>ManyYears</code> – Allows a long range of years and provides keyboard support. While generally more compact visually, it can take more clicks to operate.</li><li>• <code>FewYears</code> – Allows a maximum of 10 years and lets the user select a value with one click. It lacks keyboard support.</li></ul>
<b>▲ xHeaderTitle</b>	An optional title to appear in a header on the control. The appearance is controlled by the style sheet class <code>MYPHeader</code> or by assigning a different class to <code>xHeaderCSSClass</code> . The alignment is set by <code>xHeaderHorizontalAlign</code> , which defaults to <code>Left</code> .
<b>▲ xKeyboardControlB</b>	The <code>ManyYears</code> format supports keyboard control. If you don't need that feature, set this to <code>false</code> . It will remove a small javascript file, reducing the transmission time for the page.
<b>xDateTime</b>	Get and set the month and year using a <code>DateTime</code> object. This is an alternative to using the <code>xMonth</code> and <code>xYear</code> properties to get and set the value.

## Several images available for year navigation buttons

Version 1.x provided one set of images for the Year Navigation buttons. They are:  

Version 2 includes several additional sets of images. It also provides a design mode editor to pick them. Select the **Edit Year Navigation Buttons** command from the control's action menu or at the bottom of the Properties Editor.



## TimeOfDayTextBox and DurationTextBox Improvements

### *TimePicker control*

▲ The `PopupTimePicker` control is an option of these textboxes. See the topic “Popup TimePicker” in the section “Using the TimeOfDayTextBox” in the **User's Guide**, page 322.

### *UnselectableTimesValidator*

With the `TimePicker`, there is now a list of times assigned to these textbox controls. With the `UnselectableTimesValidator`, you can demand that the user only enter types from the list. It works even if you do not show the `TimePicker`. You only need to set up time values.

It can also evaluate times based on an `Time Interval Rule`, where you define a time increment and it reports errors if the user does not enter a time that works with that increment. For example, if the increment is set to 15, the user must enter times on 15 minute intervals from 0:00 (midnight) or `xMinTime` if `xFromMinTimeB` is `true`.

To use it, add it to the page and assign its `ControlToValidate` property to a `TimeOfDayTextBox` or `DurationTextBox`. Set the `xUseTimePickerTimesB` to `true` for `TimePicker` times. Set the `xTimeIncrement` to the number of seconds between valid times to use the `Increment Rule`. It will calculate increments from 0:00:00 unless you setup the textbox's `xMinTime` property and set it's own `xFromMinTimeB` property to `true`. Then it will use the `xMinTime` as the start time.

Professional Validation And More users also have a VAM compatible `UnselectableTimesValidator`.

### *Other New Properties*

▲ <b>xTimeFormat</b>	<p><code>TimeOfDayTextBox</code> only. Determines whether to show 12 or 24 hour time format. The enumerated type <code>PeterBlum.PetersDatePackage.TimeFormat</code> provides these values:</p> <ul style="list-style-type: none"><li>• <code>Auto</code> – Use <code>xDateTimeFormatInfo.ShortTimePattern</code>. If it has an uppercase H, it's 24 hr. Otherwise, it is 12.</li><li>• <code>Hr12</code> – Use 12 hour format.</li><li>• <code>Hr24</code> – Use 24 hour format.</li></ul>
▲ <b>xReadOnlyAllowsEditsB</b>	<p>You can prevent the user from typing in a time, but still allow the <code>TimePicker</code>, arrow buttons, and context menu commands to edit it. This assures you that specific dates are entered. Set both <code>ReadOnly</code> and this property to <code>true</code>.</p>
▲ <b>xBlankStartsAt</b>	<p>When the text box is blank and the user uses the <code>Next</code> or <code>Previous Hour</code> or <code>Minute</code> commands, this defines the initial value. It can now start at the current time of day or a value you pick.</p>