WebScheduler 3.0 White Paper

This white paper discusses the new features and enhancement introduced in WebScheduler 3.0.

Table of Contents

XHTML Support

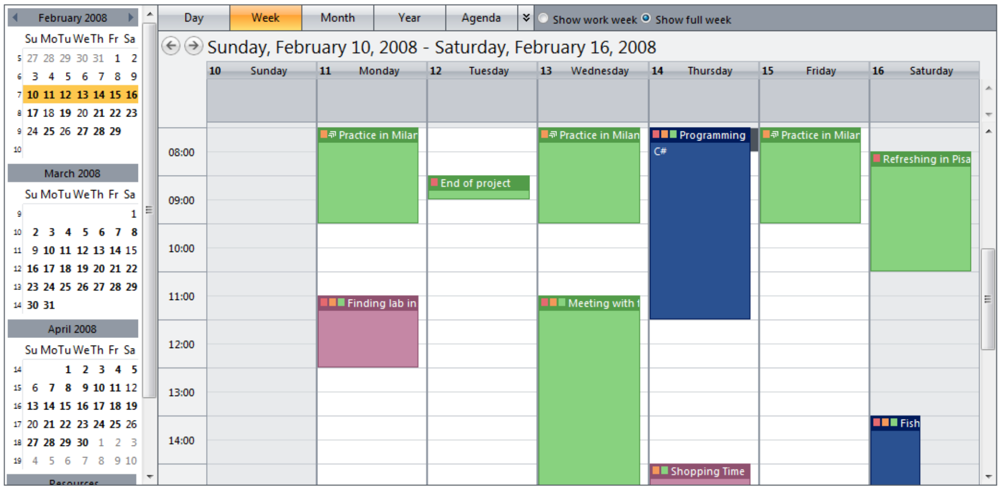
With the comprehensive W3C XHTML standard support, WebScheduler offers precise rendering capability in every modern application. This substantial enhancement gives you the power to develop a new XHTML enterprise solution or adding a scheduling module to your classic HTML application - even it means a thorough enhancement in WebScheduler's core engine has to be made. Its smart detection engine automatically tracks the document type used and applied the necessary adjustment to its entire styles and behaviors – without developer's intervention.

Following are the enhanced and optimized features for efficient XHTML output.

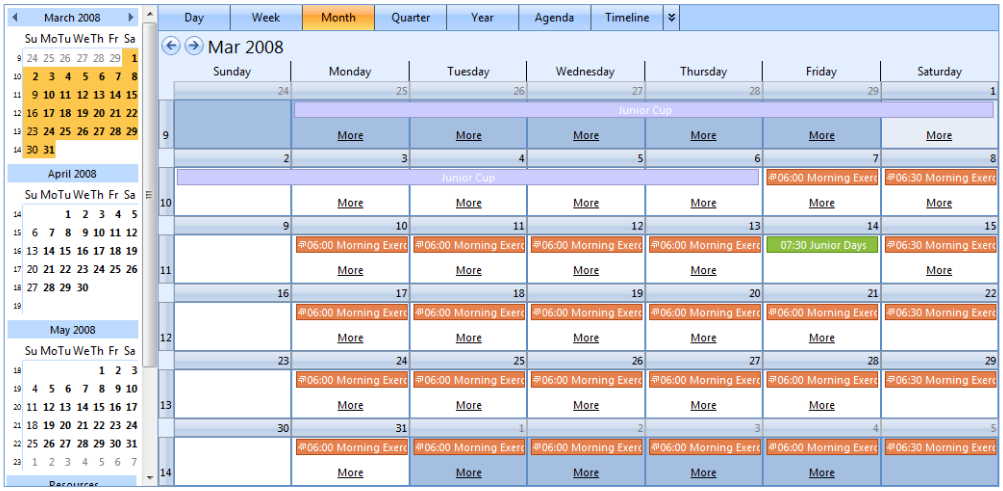
- Core User Interface, including the complete layout management on the scheduler and navigation interface.
- UX Components, including all user interaction elements, such as calendar, callout, dialog box, drop shadow, list box and many more.
- Scheduler views, including Day, Week, Month, Agenda, Quarter, Year, Timeline, and Split view.
- Built-in Editing form.

The extensive XHTML support is designed and thoroughly tested to work on all future-proof browsers.

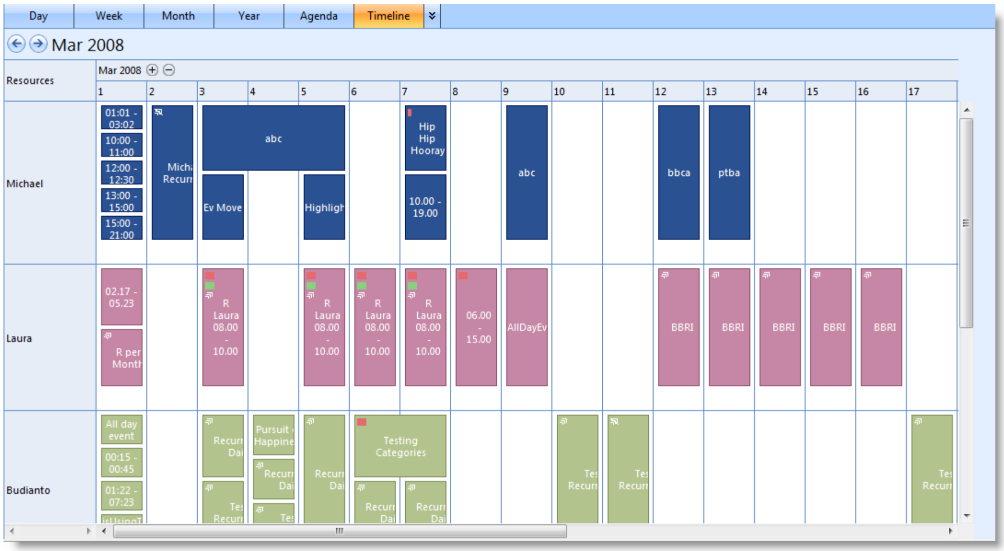
- Microsoft Internet Explorer 7 Standards Mode
- Microsoft Internet Explorer 8 Standards Mode
- Mozilla Firefox 3+
- Apple Safari 4+
- Opera 10+
- Google Chrome



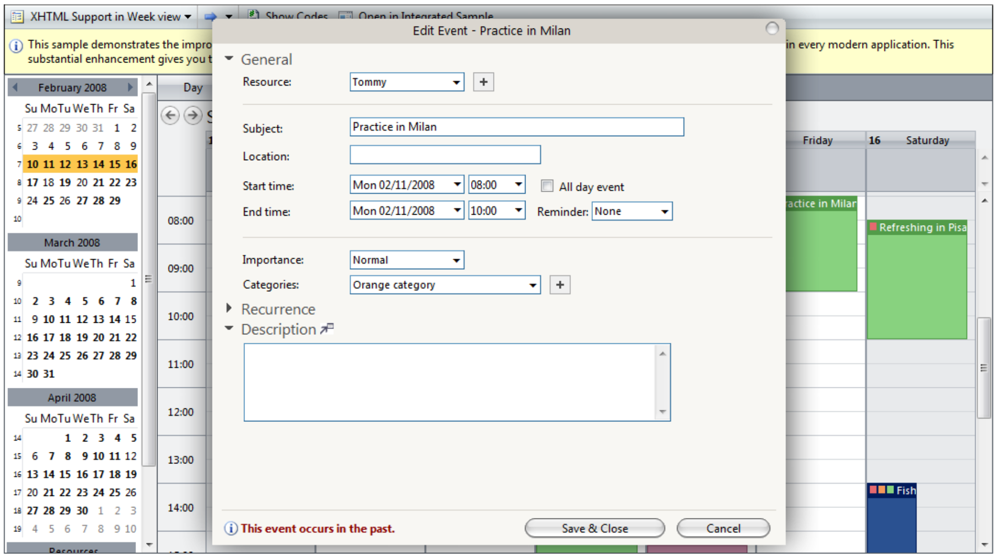
Week view in XHTML mode



Month view in XHTML mode



Timeline view in XHTML mode



Edit Event - Practice in Milan

General

Resource: Tommy

Subject: Practice in Milan

Location:

Start time: Mon 02/11/2008 08:00 □ All day event

End time: Mon 02/11/2008 10:00 Reminder: None

Importance: Normal

Categories: Orange category

Recurrence

Description

This event occurs in the past.

Built-in Editing Form in XHTML mode

Performance Improvements

WebScheduler supports advanced enterprise scheduling scenarios such as displaying thousands of data in a single load. In this new release, WebScheduler is finely tuned to display large datasets without significant performance hits enabling developers to achieve rich scheduling solution with great scalability and extensibility.

Key performance improvements include:

Data Transfer

WebScheduler 3 incorporates JSON technology to reduce the amount of data being transferred on each AJAX callback, making the data transfer faster and more efficient.

In large data scenario, the overall performance is improved by 50% – 70% when **EnableFastJsonSerialization** property is enabled.

If fast JSON serialization is enabled and extensibility feature is used in WebScheduler, additional implementation is required to process the custom fields during serialization process. Simply add the custom fields to *DataDictionary* object.

```
protected void WebScheduler_OnJsonSerialization(object sender,
WebSchedulerJsonSerializationArgs e)
{
        switch (e.Type)
        {
                case WebSchedulerObjectType.Event:
                        Dictionary<string, object> eventDict = (Dictionary<string,
object>)e.DataDictionary;
                        EOEvent obj = (EOEvent)e.DataObject;
```

```
                  eventDict["Notes"] = (string)obj.Notes;
                  eventDict["TotalAttendees"] = (int)obj.TotalAttendees;

                  break;

            case WebSchedulerObjectType.RecurringEvent:
                  Dictionary<string, object> recDict = (Dictionary<string,
object>)e.DataDictionary;
                  EORecurringEvent obj1 = (EORecurringEvent)e.DataObject;

                  recDict["Notes"] = (string)obj1.Notes;
                  recDict["TotalAttendees"] = (int)obj1.TotalAttendees;

                  break;
      }
}
```

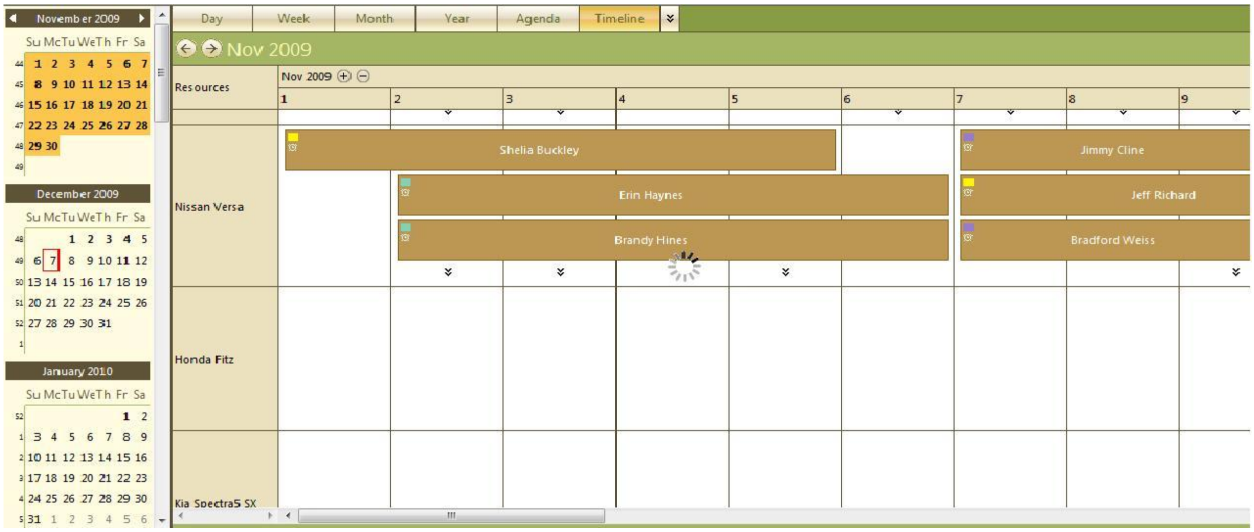
Rendering

The rendering process in WebScheduler 3 has been further optimized to meet high performance standards in enterprise scheduling scenarios. Combined with the new client paging feature, WebScheduler **performs 4x faster** compared to its previous version. The client paging will display only certain number of data in the first load and load the rest of data when users demand it – making efficient use of resources and screen real estate.

Server Paging in Timeline view

Concept

Designed as an enterprise-ready scheduling component, WebScheduler 3 provides an option to control the data retrieval mechanism. Server Paging is an innovative feature built to eliminate sluggish performance due to large data download in Timeline view mode. With server paging, WebScheduler downloads only a small chunk of data based on the viewport during initial load and intelligently requests more data as users scroll downward. This translates to reduced page load time and increased overall application responsiveness. Combined with the smart client paging rendering technology, powerful JSON response format, and script modularization, WebScheduler is the world’s fastest and most advanced scheduling solution.



Data loaded based on the viewable viewport

Implementation

Simply enable ***EnableServerPaging*** property in ViewSettings >> TimelineView.

Note: Server paging feature is only applicable when the client paging is set to ***ViewPort*** or ***Both***. For certain scenarios with smaller data (less than a thousand of records in a month view), the server paging may not be necessary enabled so that the complete data can be transmitted in a single page load. In such scenarios, the client paging will perform data rendering immediately.

Timeline (Gantt Chart) View

WebScheduler 3 takes the performance in Timeline view to the next level by introducing advanced paging mechanism. Thanks to the innovative ViewPort™ paging technology, the Timeline view is now able to display several thousands of events in a fraction of second, which significantly increasing application responsiveness and delivering superior user experiences.

Script Modularization

WebScheduler 3 now **loads 2x faster** with improved script modularization. When certain views or features are disabled, WebScheduler automatically excludes all scripts associated to those features from being loaded, thus reducing the time and resources required to instantiate the WebScheduler control.

Client Paging

Sleek performance and snappy user interface never goes hand in hand with large data. This is the challenge that every enterprise scheduling solution has to face. WebScheduler 3 elegantly addresses the scalability issue with its sophisticated client paging feature – significantly improving overall application response regardless of the data size.

Client paging can be enabled in Day, Week, Timeline, and Split view. In Month view, it is enabled by default, since the layout structure requires it to be implemented when large data is used.

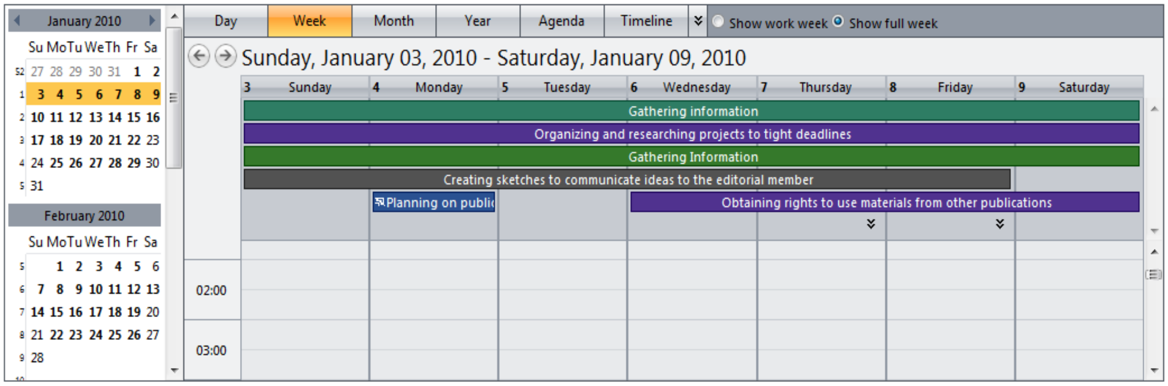
Day view, Week view, and Split view

Client paging can be enabled via ***EnabledClientPaging*** property, accessible inside the ***ViewSettings >> DayView/WeekView***. Sharing the same configuration and properties, Split view will be affected on any settings made for Day view. This means the client paging is enabled in Split view as well. Client paging is view independent, meaning that you can control which view to have client paging enabled.

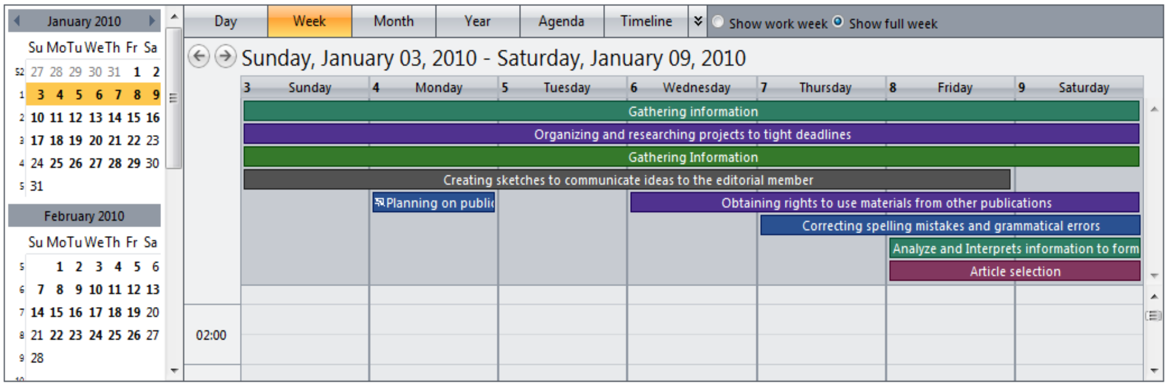
The scheduler area is divided into two, all-day area and hour area. Located on the top, all-day events are rendered in here. Time-based events are rendered in the larger hour area. Client paging feature is implemented for both area and controllable using these following properties:

1. AllDayEventPageSize
   AllDayEventPageSize property controls the maximum rows in the all-day area. The default value is 3, which means that only all-day events rendered within the 3 rows are visible. Click on the paging indicator to display all all-day events.

   The value of the custom page size can be defined in ***ViewSettings >> DayView/WeekView >> AllDatEventPageSize***. Set it to -1 means that client paging feature is disabled and all all-day events will be displayed.



All-day events are paged in Week view


January 2010
Su Mo Tu We Th Fr Sa
February 2010
Day
Week
Month
Year
Agenda
Timeline
Show work week
Show full week
Sunday, January 03, 2010 - Saturday, January 09, 2010
3 Sunday
4 Monday
5 Tuesday
6 Wednesday
7 Thursday
8 Friday
9 Saturday
Gathering information
Organizing and researching projects to tight deadlines
Gathering Information
Creating sketches to communicate ideas to the editorial member
Planning on publi
Obtaining rights to use materials from other publications
Correcting spelling mistakes and grammatical errors
Analyze and Interprets information to form
Article selection
02:00

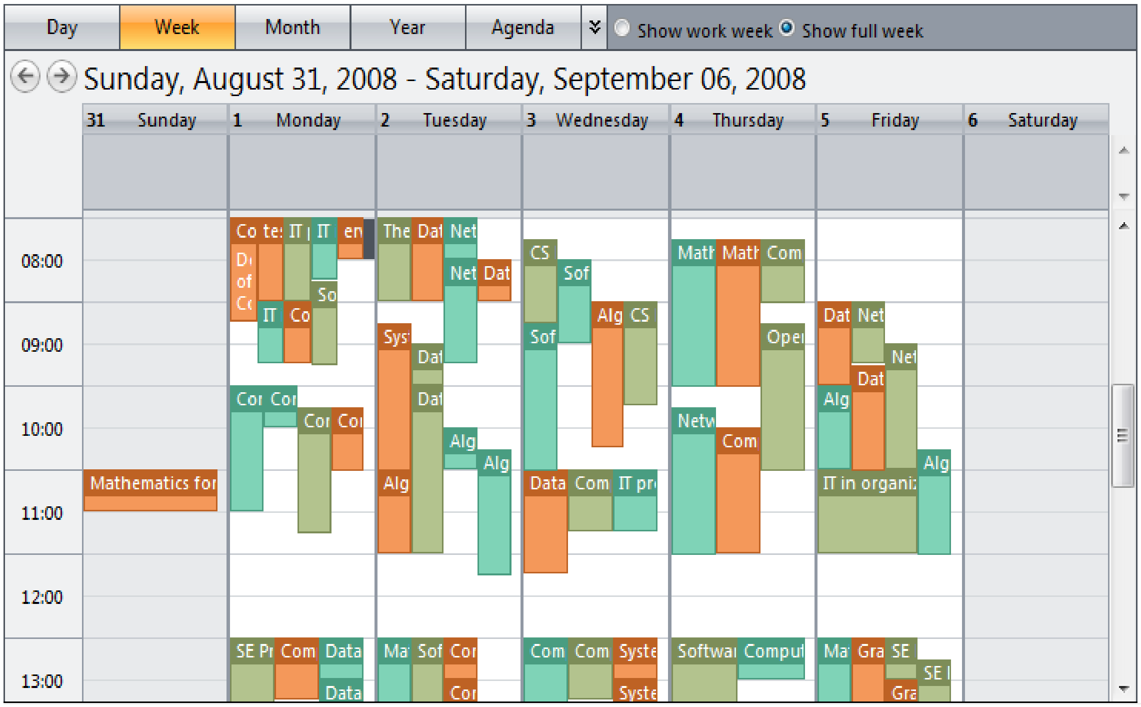
All-day events are displayed when paging arrow indicator is clicked

In the above picture, ***AllDayEventPageSize*** is set to 5, so only all-day events rendered in the first 5 rows are displayed. Click on the paging indicator to reveal all all-day events.

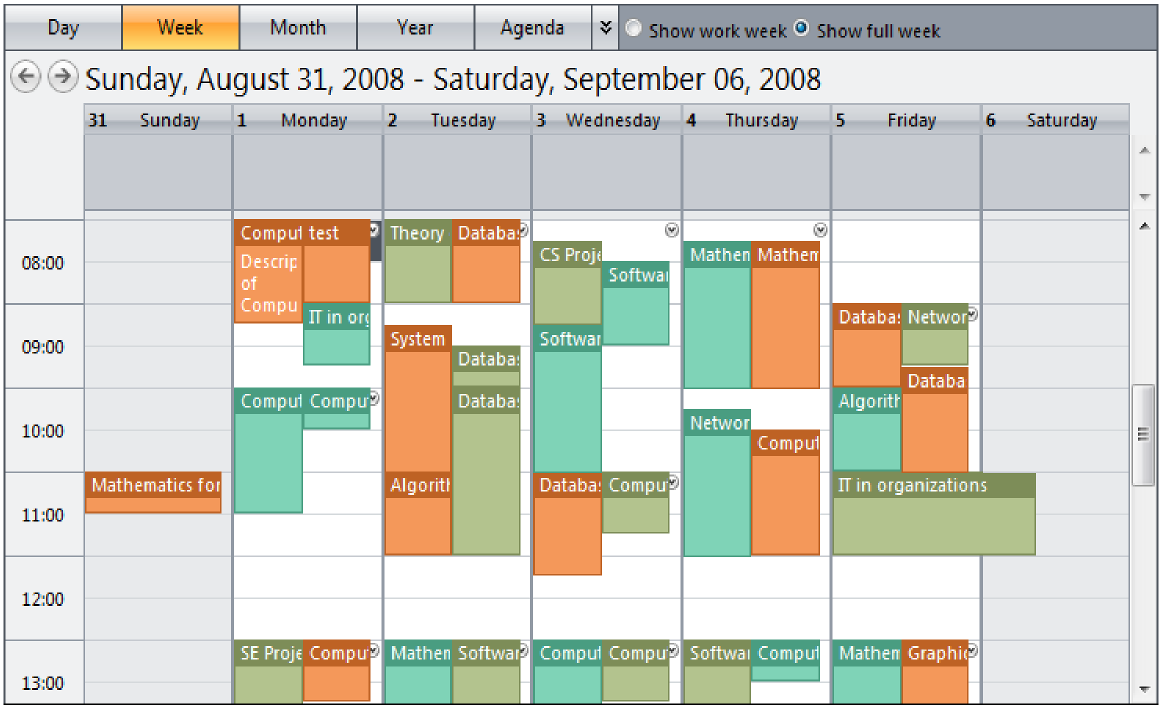
2. EventBlockSize

EventBlockSize property determines the maximum number of overlapping blocks rendered in an event block. The default value is 3. This means only events in the first three blocks is rendered. Click on the paging indicator to show all events in an elegant detail box.

The value of the custom page size can be defined in ***ViewSettings >> DayView/WeekView >> AllDatEventPageSize***. Set it to -1 means that client paging feature is disabled and all all-day events will be displayed.

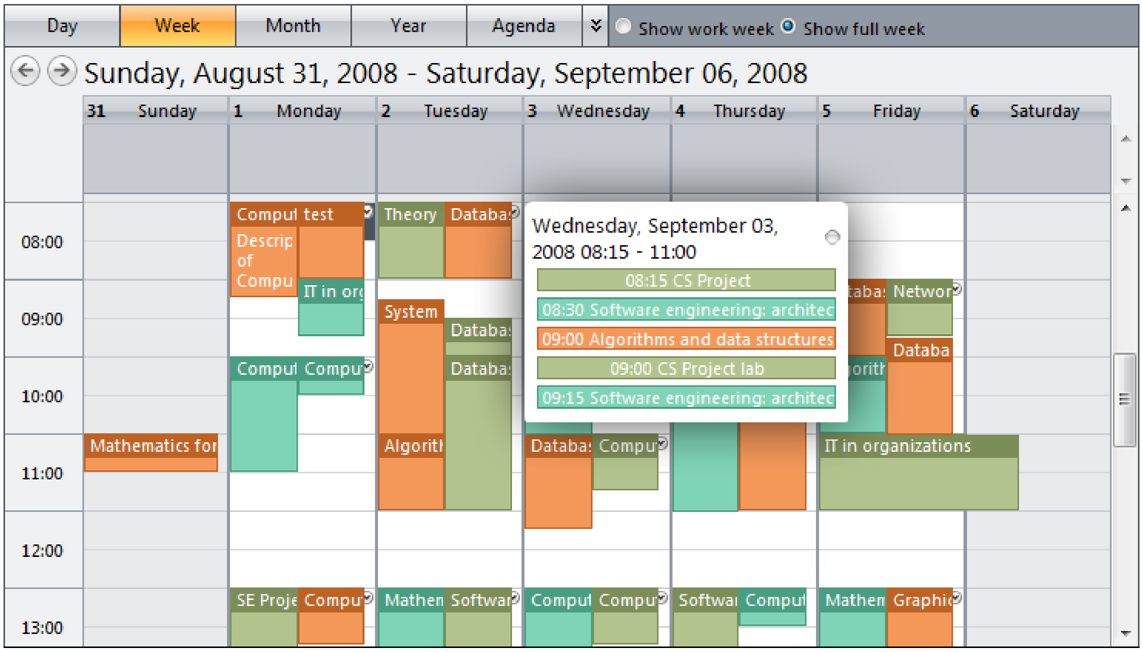


All time-based events displayed in Week view



Time-based events are paged in Week view

As seen in above screenshots, on September 3rd 2008 8:15 AM – 11: 00 AM, there are 5 events rendered in 4 overlapping blocks. When the ***EventBlockSize*** property is set to 2, those 5 events will be paged and rendered as 3 events in 2 overlapping blocks and the others will be hidden. A paging arrow indicator will be displayed in the paged blocks to indicate these hidden events.



Wednesday, September 03, 2008 08:15 - 11:00

08:15 CS Project
08:30 Software engineering: architec
09:00 Algorithms and data structures
09:00 CS Project lab
09:15 Software engineering: architec

When paging arrow indicator is clicked, the hidden events will be rendered in the Detail Box.

Timeline view

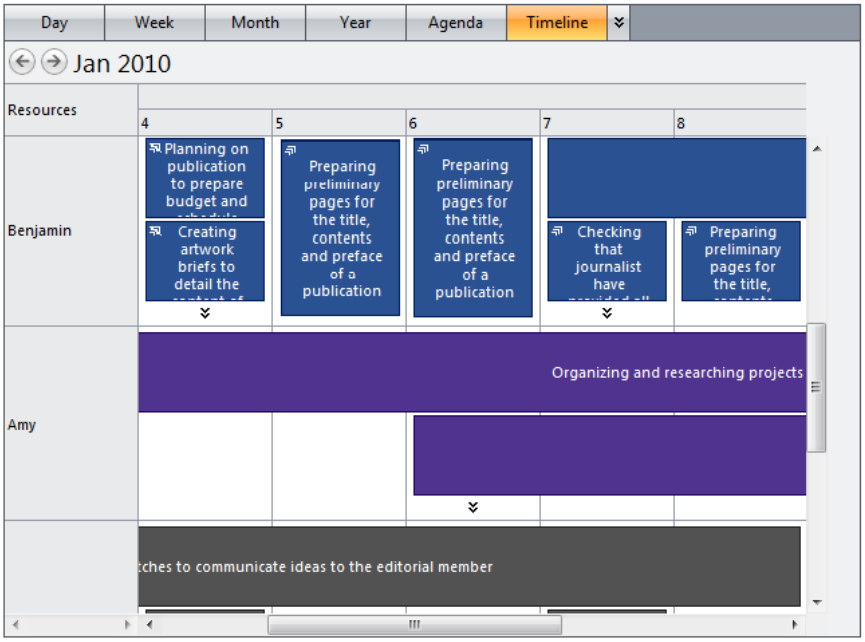
To enable client paging in Timeline view, ***ViewSettings >> TimelineView >> EnableClientPaging*** property needs to be set to *True*.

There are three client paging modes introduced in Timeline view:

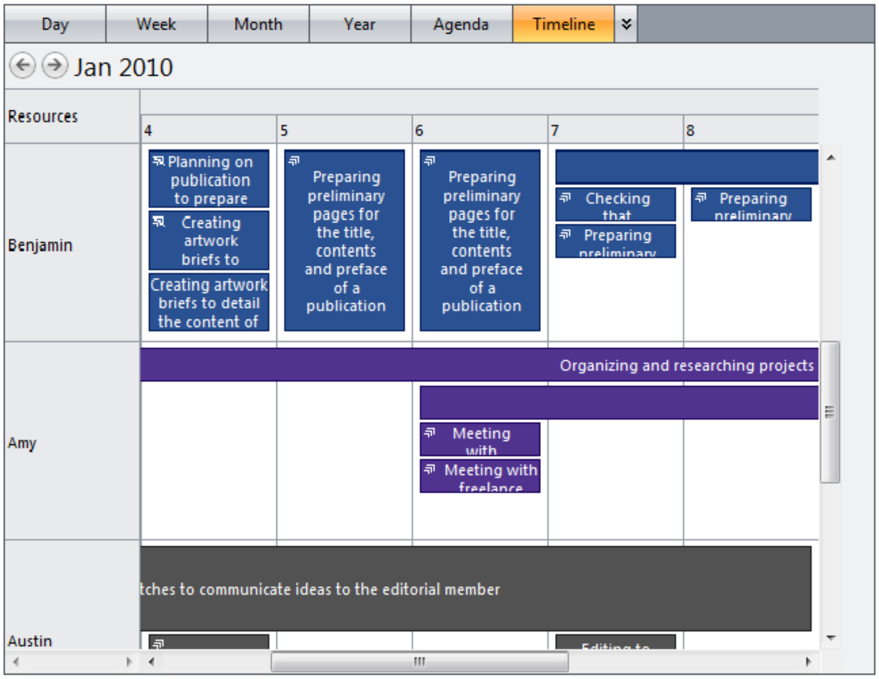
1. EventPageSize

   The ***EventPageSize™*** mode restricts numbers of event displayed in each Timeline cell. If the total number of events exceeds the value of the EventPageSize property, paging arrow indicator will appear signifying that some events are hidden. A simple click will reveals the complete events in the Timeline cell.

   To enable this mode, set ***ViewSettings >> TimelineView >> ClientPagingMode*** property to *EventPageSize*.


Day
Week
Month
Year
Agenda
Timeline
Jan 2010
Resources
4
5
6
7
8
Benjamin
Planning on publication to prepare budget and
Creating artwork briefs to detail the
Preparing preliminary pages for the title, contents and preface of a publication
Preparing preliminary pages for the title, contents and preface of a publication
Checking that journalist have
Preparing preliminary pages for the title,
Amy
Organizing and researching projects
ches to communicate ideas to the editorial member

Events are paged in Timeline view

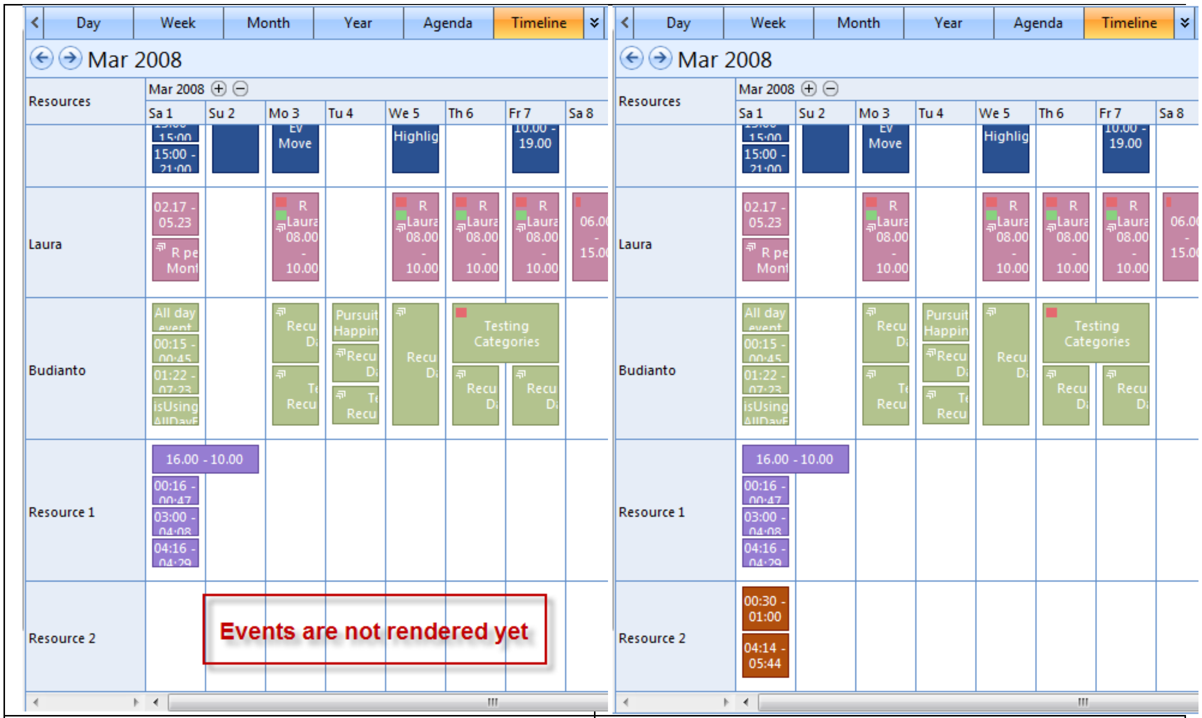

Day
Week
Month
Year
Agenda
Timeline
Jan 2010
Resources
4
5
6
7
8
Benjamin
Planning on publication to prepare
Creating artwork briefs to
Creating artwork briefs to detail the content of
Preparing preliminary pages for the title, contents and preface of a publication
Preparing preliminary pages for the title, contents and preface of a publication
Checking that
Preparing preliminary
Preparing preliminary
Amy
Organizing and researching projects
Meeting with
Meeting with freelance
Austin
tches to communicate ideas to the editorial member
Editing to

Timeline events are displayed when paging arrow indicator is clicked

2. ViewPort
   When activated, ViewPort™ paging mode will render all events based on the visible viewport. This advanced paging mode intelligently detects the screen's real estate and display the events as user scroll down the page. The previously displayed events will also be cached to improve the overall user experiences.

   For instance, when WebScheduler is loaded and only Resources A and Resources B are visible, it will only display all Resources A's and Resources B's events. The events for other resources will be displayed on demand as the viewport is scrolled.

   To enable this mode, set the ***ViewSettings >> TimelineView >> ClientPagingMode*** property to *ViewPort*.



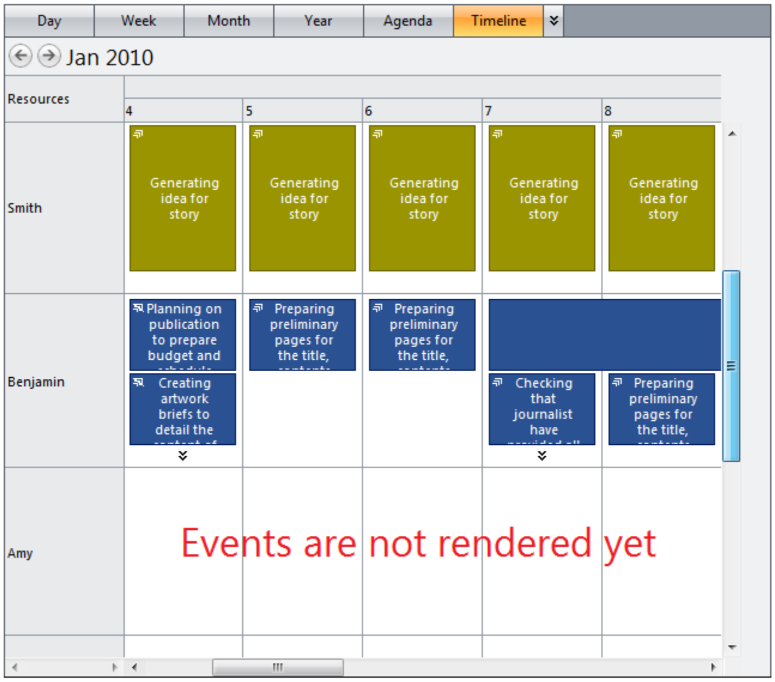
Events are not yet rendered in initial load

Events are rendered when users scroll down

ViewPort Client Paging Mode

3. Both
   This mode combines the best of both **EventPageSize™** and **ViewPort™** client paging for the most efficient event rendering. This concept renders events based on visible viewport and restricts them when exceeding the numbers of allowed event.

   To enable this mode, set ***ViewSettings >> TimelineView >> ClientPagingMode*** property to *Both*.


Day
Week
Month
Year
Agenda
Timeline
Jan 2010
Resources
4
5
6
7
8
Smith
Generating idea for story
Generating idea for story
Generating idea for story
Generating idea for story
Generating idea for story
Benjamin
Planning on publication to prepare budget and
Preparing preliminary pages for the title,
Preparing preliminary pages for the title,
Creating artwork briefs to detail the
Checking that journalist have
Preparing preliminary pages for the title,
Amy
Events are not rendered yet

Both client paging mode implemented in Timeline view

Show all Timeline events in client-side

When client paging is enabled in Timeline view and EventPageSize mode is applied, ShowAllEvents client-side function can be used to show all resources' hidden events.

```
<script language="javascript">
function ShowAllEvents()
{
      var s = ISGetObject("WebScheduler1");
      s.ShowAllEvents();
}
</script>
```

Data editing when Client paging is enabled

WebScheduler's event rendering behavior is based on each event's duration. An event with the longest duration in an event block is rendered and vice versa, an event with shortest duration will stay invisible inside the detail box – depending on the event block size value.

When a new event is added, WebScheduler will automatically detect the duration of the event. If the duration is longer than all events displayed in the event block, the new event will be added to the event block. An existing event with the shortest duration in the event block will be shifted inside the detail box. If the newly added event's duration is the shortest among all events displayed in the event block, the newly added event will be shifted inside the detail box.

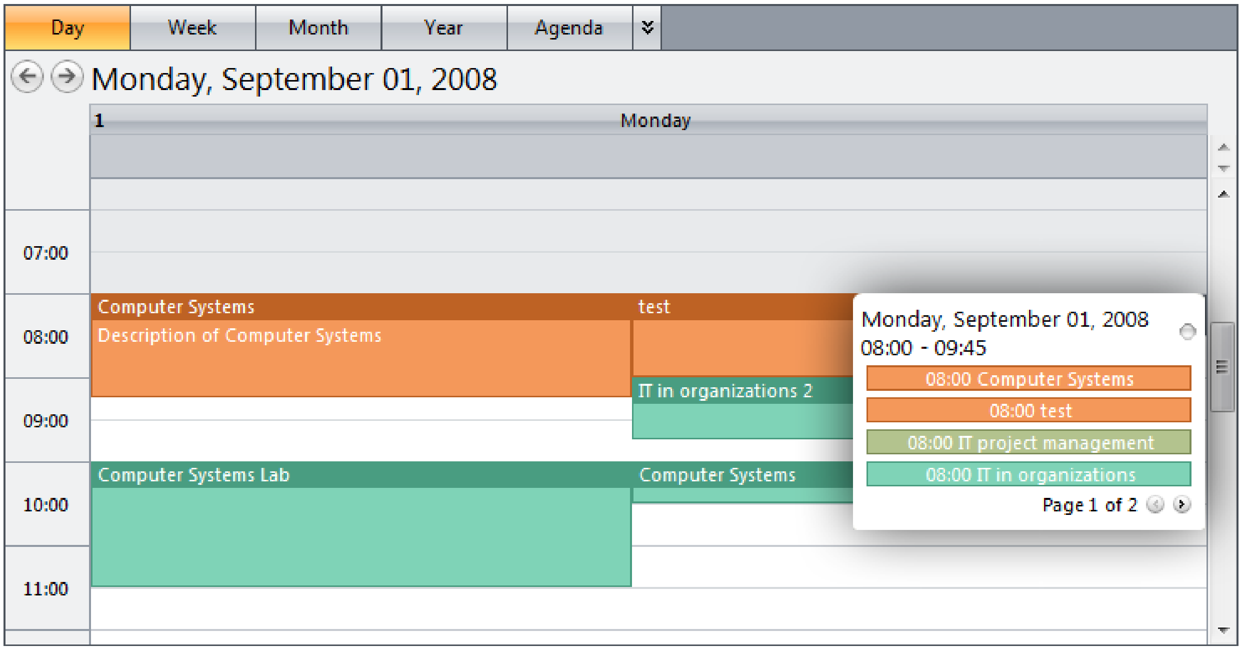
This is also applicable to delete and update an event. When an event is deleted from the event block, only the event block is re-rendered. During the re-rendering process, WebScheduler will detect the total events. If it is smaller than the maximum events set through the ***EventPageSize/EventBlockSize*** property, the paging indicator will be removed.

Detail Box Paging

When client paging is enabled in Day, Week, and Split view, a detail box will be displayed when user clicks on the paging indicator on the top right corner of an event block. The detail box also offers paging capability. The detail box's header will display the date range of the current event block.

Set the ***EnableDetailBoxPaging*** to *True* to enable the detail box paging feature. You can also specify the page size via ***DetailBoxPageSize*** property. The default value is 5. This means the detail box will hold maximum 5 events on every page.

Day, Week, Month, and Split view will be affected by this global configuration.



Detail Box Paging with Page Size set to 4

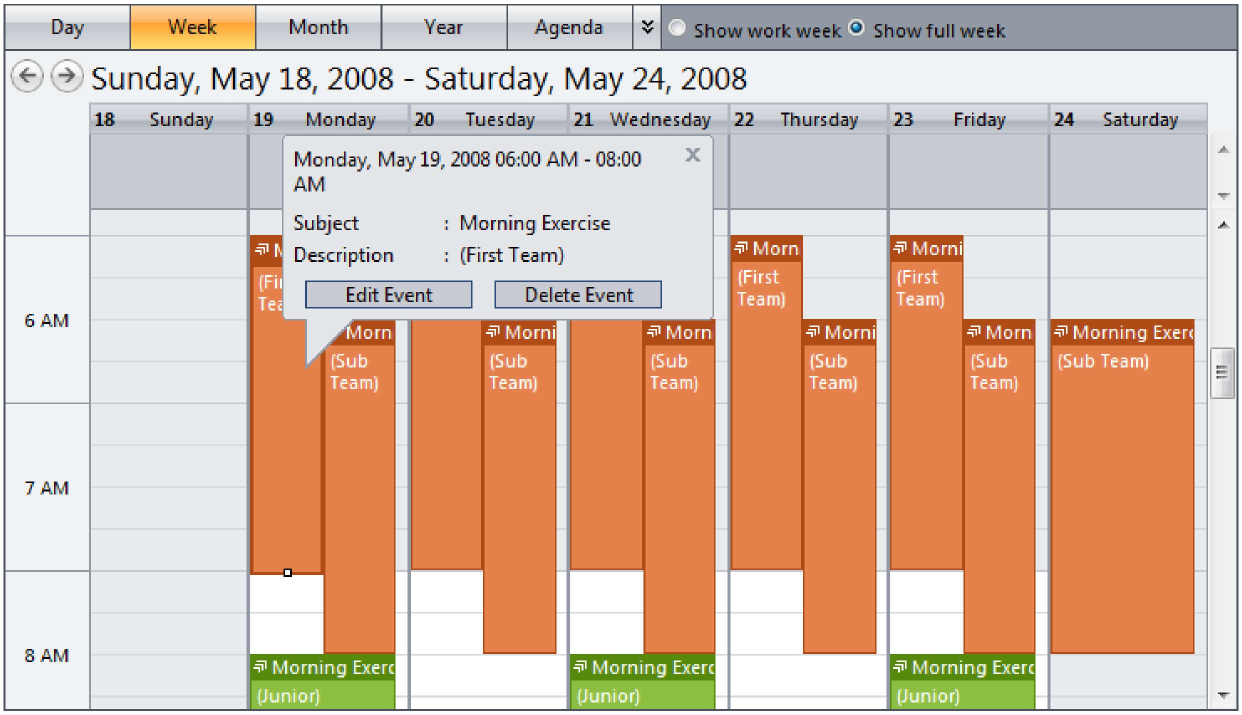
Time Interval

Day, Week, and Split view have two scheduler areas: the all-day area and the hour area. Previously, the hour area is displayed and fixed to 30-minutes interval. In version 3, WebScheduler allows you to alter this setting with various predefined time interval.

Various Time Interval Options

The available time interval options for all Day, Week, and Split views are: *Minutes10*, *Minutes15*, *Minutes30*, and *Mintues60*. You can set it from ***ViewSettings >> DayView/WeekView >> TimeInterval***.

You can also customize to have each view uses different time interval. Since Split view shares the same configuration as Day view, any settings made are applicable to both views.



○ Show work week ● Show full week

Monday, May 19, 2008 06:00 AM - 08:00 AM

Subject : Morning Exercise

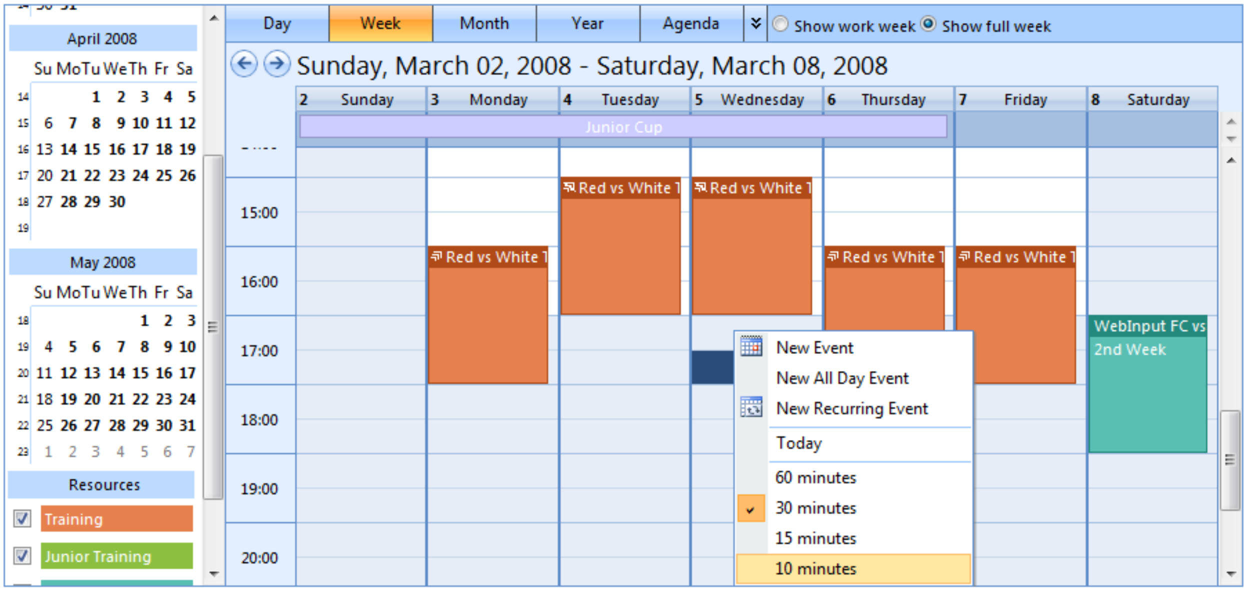
Description : (First Team)

15-minutes Time Interval in Week view

When time interval is set to 60-minutes duration, the row height is multiplied by 2, allowing users to see short duration events clearly. Additionally, all event UI elements related will be adjusted according to the active time interval.

Included in Context Menu

User can also control the time interval in runtime by easily selecting the time interval option for the context menu. Once it's changed, WebScheduler will automatically re-render its scheduler view and its events, adjusting to the selected time interval. You can control user's access to time interval via context menu in the ***IncludeTimeIntervalInMenu*** property.



Time interval options included in context menu

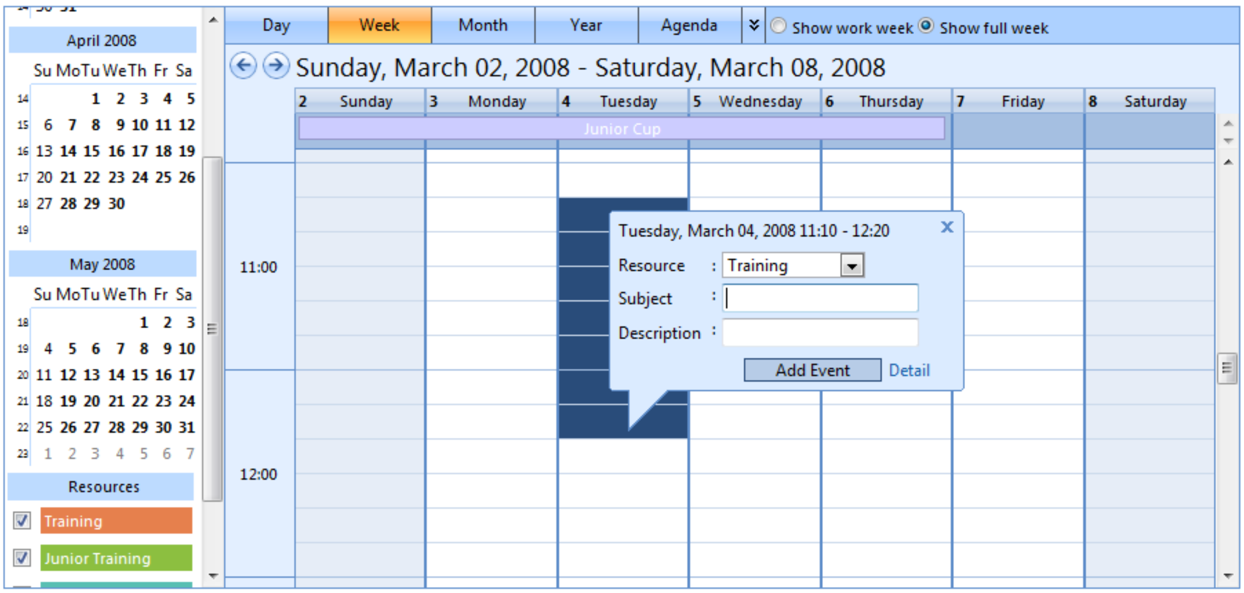
Set Time Interval Programmatically

Time interval can be set via client-side API, the **SetTimeInterval** function. The following code snippet will change the time interval to 15 minutes interval.

```
<script type="text/javascript">
function SetInterval()
{
      var s = ISGetObject("WebScheduler1");
      s.SetTimeInterval("Minutes15");
}
</script>
```

Integrated Multiple Cell Selection Support

Multiple cell selection is an innovative feature which enables user to block a certain date range and quickly create a new event based on the selection. Now, it is enhanced further to work in conjunction with the new time interval option. The duration of the selected cells is adjusted according to the active time interval. The duration can later be seen in the callout's header and new event can be added directly.



Tuesday, March 04, 2008 11:10 - 12:20

Resource : Training

Subject :

Description :

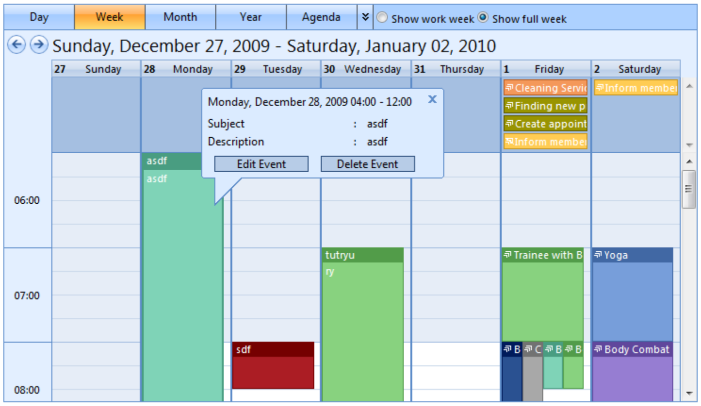
Date range of selected multiple cell adjusted with 10-minutes time interval

Visible Hours

By default, the hour area in Day, Week, and Split view starts from 00:00 AM to 11:30 PM. WebScheduler 3 gives you the flexibility to specify your own visible hour by customizing the start and end time.

The ***VisibleStartTime*** and ***VisibleEndTime*** properties are located inside the ***ViewSettings***. WebScheduler will render the visible hour based on the given value.

Event rendering will be adjusted according to these settings. For example: the visible start time is set to 06:00 AM and visible end time is set to 07:00 PM. All events that fall within the date range will be rendered. Any events that start at 04:00 AM and ends at 12:00 AM will be rendered from 06:00 AM to 12:00 following the start time. The original duration of that event is not altered. It can still be seen from the callout.



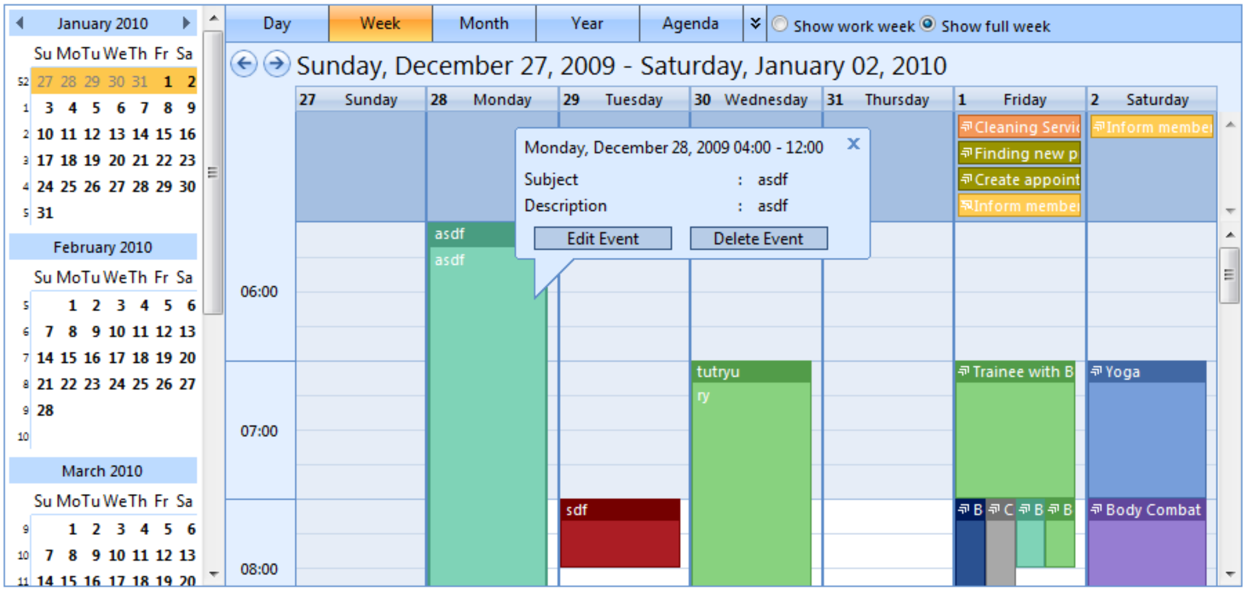
Monday, December 28, 2009 04:00 - 12:00

Subject : asdf

Description : asdf

Week view starts from 06:00 AM and ends at 07:00 PM

Although only events within the visible hours are rendered, the calendar dates will still show all events - including the currently invisible one. This behavior is implemented to allow users to see the summary of all events in the active month.



Monday, December 28, 2009 04:00 - 12:00

Subject : asdf

Description : asdf

Event indicator in calendar counts all events in the active month

Disabled Time

Disabled time is a unique feature which enables developer to specify a portion of time when event creation is disabled without requiring complex efforts or codes. When the disabled time is specified, user will not be able to add a new or edit an existing event and set it within the range of specified disabled time period.

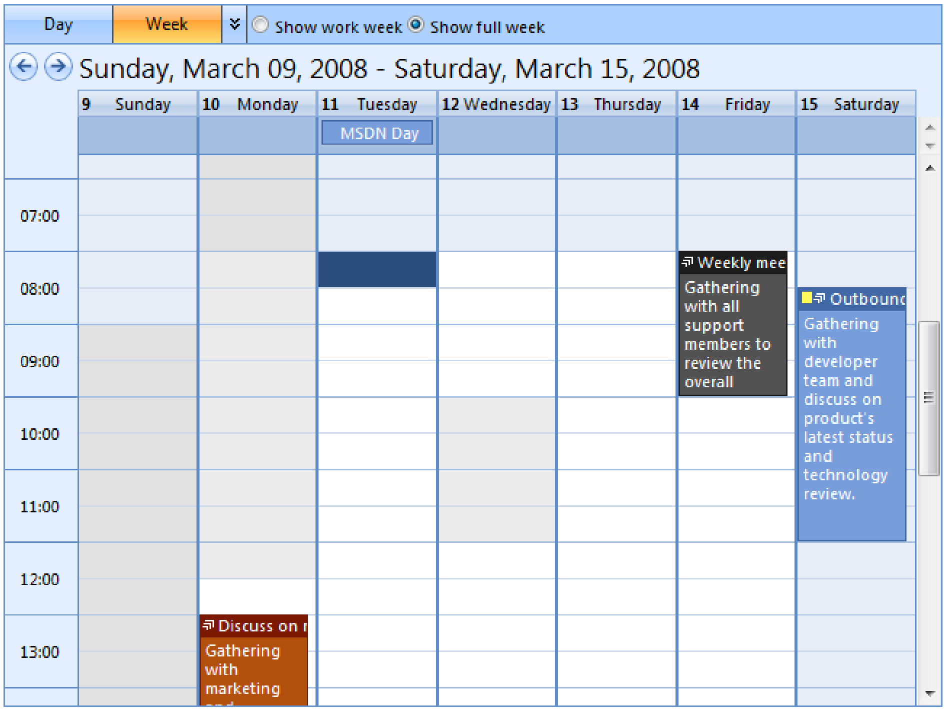
Since data operation cannot be performed in Quarter and Year view and Agenda view is text-based event information, disabled time feature is not implemented in these 3 views.

Styles and Behavior

The disabled time feature affects the following features:

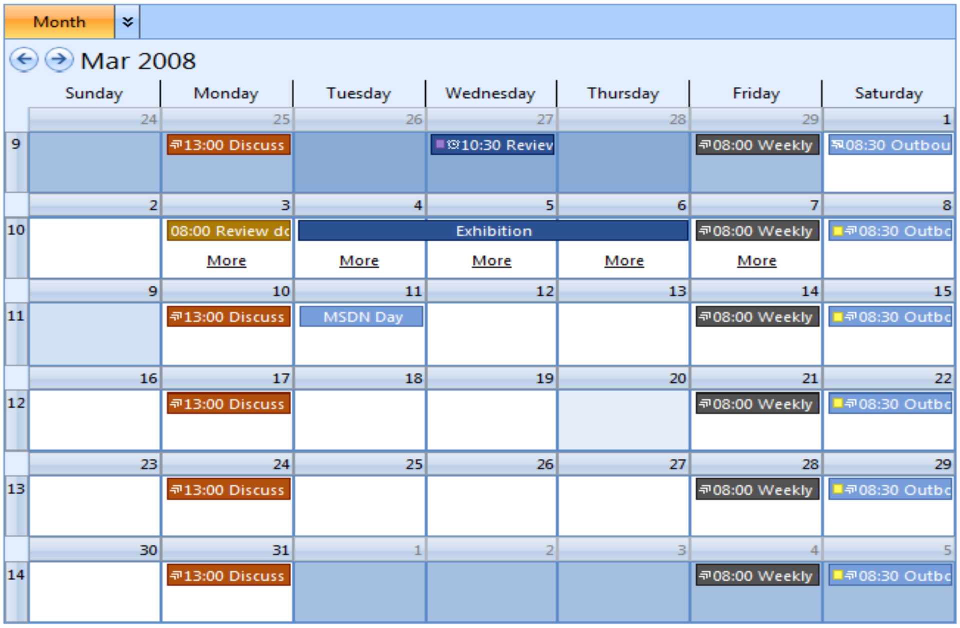
1. Cell styles in scheduler view
   In addition to blocked new event creation and editing, the disabled time cells are marked with different background color. Several styles properties are exposed for further customization on disabled time styles:

   - Day, Week, and Split view: **DisabledAllDayEventAreaStyle**, **DisabledWorkHourStyle**, and **DisabledNonWorkHourStyle**.


Day
Week
○ Show work week ● Show full week
Sunday, March 09, 2008 - Saturday, March 15, 2008
9 Sunday
10 Monday
11 Tuesday
12 Wednesday
13 Thursday
14 Friday
15 Saturday
MSDN Day
07:00
08:00
09:00
10:00
11:00
12:00
13:00
Weekly mee
Gathering with all support members to review the overall
Outbound
Gathering with developer team and discuss on product's latest status and technology review.
Discuss on r
Gathering with marketing

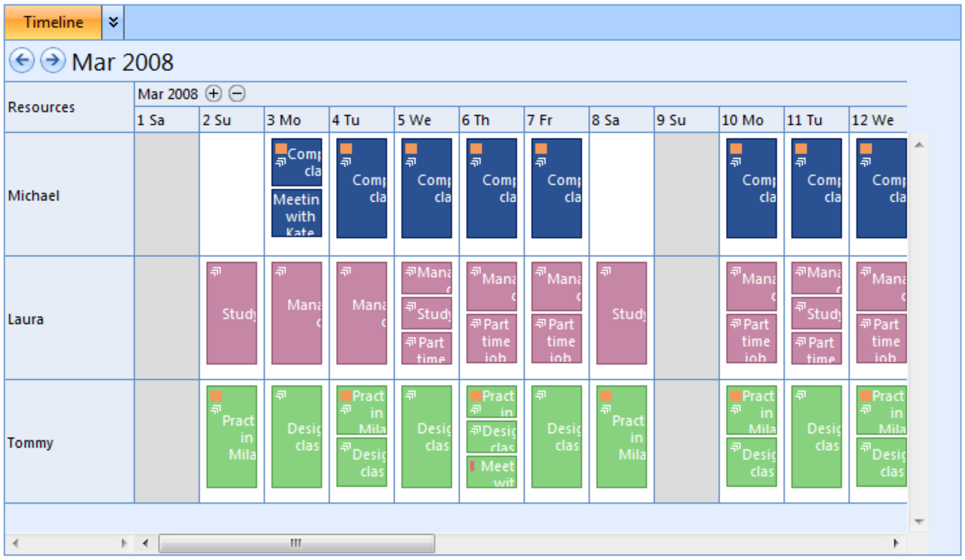
Disabled time style in Day, Week, and Split view

- Month view: **DisabledMonthOutboundCellStyle** and **DisabledMonthInboundCellStyle**.



Disabled time style in Day, Week, and Split view

- Timeline view: **TimelineDisabledTimeCellStyle.**



Disabled time style in Timeline view

2. Editing Form and CallOut

   CallOut or Editing Form will not appear when clicking on a disabled time cell – preventing user from inserting a new event. Additionally, when user tries to update an existing event, an alert box will pop out.

3. Context Menu

   New Event, New Recurring Event, and New All Day Event options in context menu will be disabled when clicking on disabled time cell.

Disabled Time Options

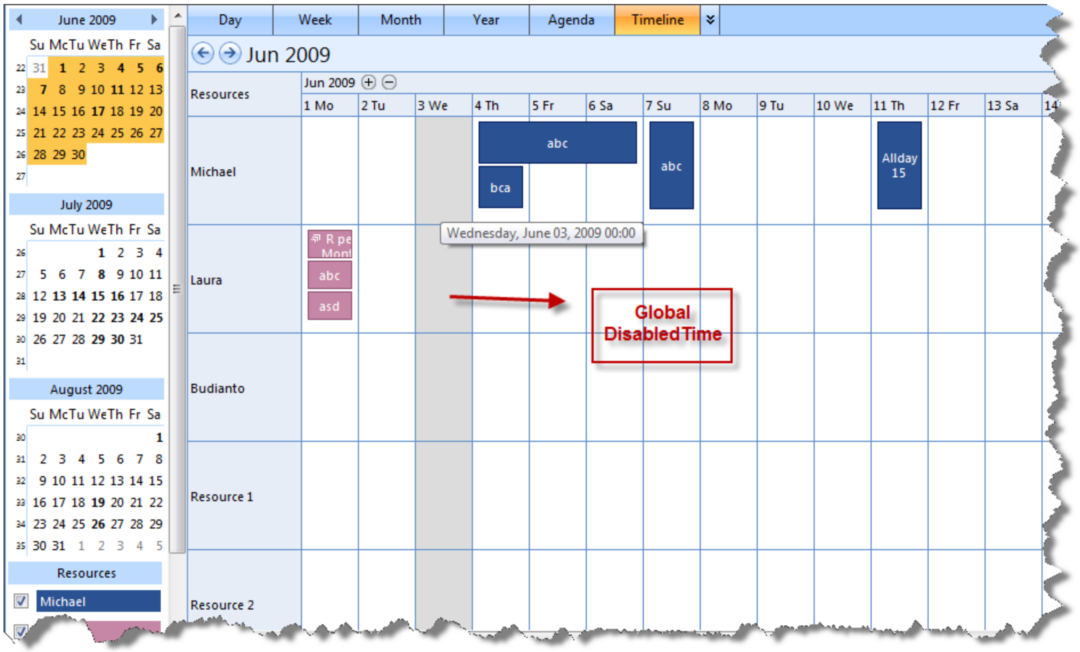
There are 3 disabled time options.

1. Global disabled time

   As the name implies, Global disabled time is applied to certain date and time and applied to all views. In Timeline view, the disabled time is applied to all resources.

   In Month and Timeline view, the following configuration sets June 3rd 2009 as disabled time. Notice that June 4th 2009 is not disabled because it is specified as the end of disabled time range. If you want to June 4th to be disabled as well, the ***EndTime*** property must be set to June 5th.

```
<DisabledTime>
   <ISWebScheduler:WebSchedulerDisabledTime EndTime="2009-06-04"
   StartTime="2009-06-03">
   </ISWebScheduler:WebSchedulerDisabledTime>
</DisabledTime>
```

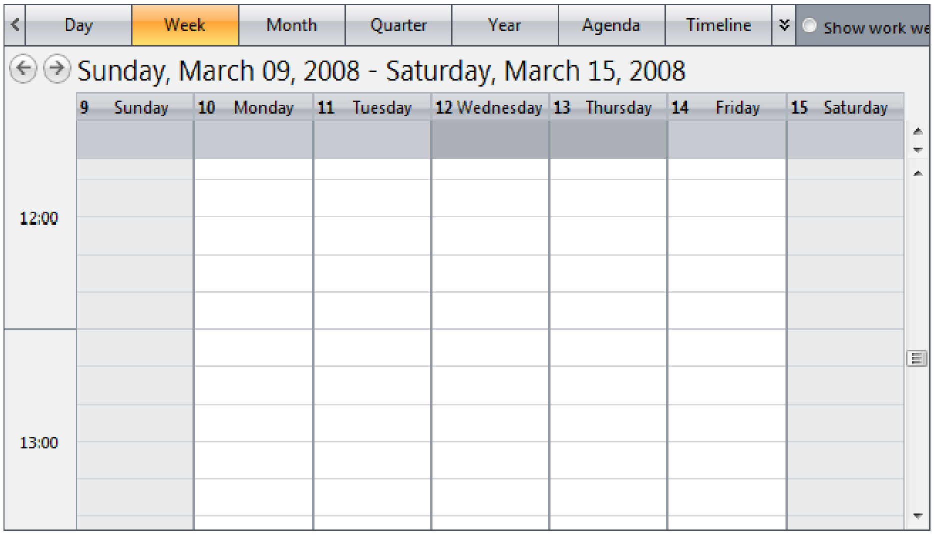


Global disabled time in Timeline View mode

Day, Week, and Split view contain two scheduler areas: the all-day area and hour area. The start time and end time of each disabled time will be used to indicate whether the all day area or hour area will be disabled.

The following configuration will set the 12th – 13th of March 2008 as disabled time. The start time and end time definition does not include time part, thus all-day area will be disabled.

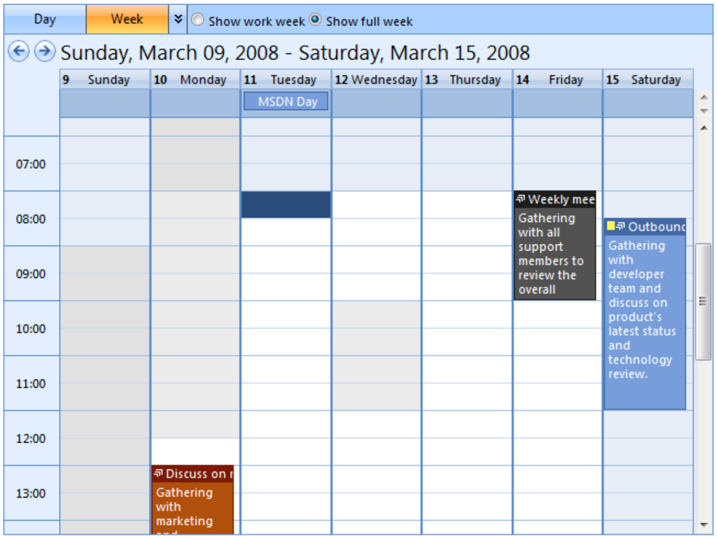
```
 <DisabledTime>
      <ISWebScheduler:WebSchedulerDisabledTime StartTime="2008-03-12"
EndTime="2008-03-13" >
      </ISWebScheduler:WebSchedulerDisabledTime>
</DisabledTime>
```



Global disabled all-day area in Week view

To disable the hour area, the complete date and time range must specified as disabled time collection. Below is an example on how to disable 10:00 AM – 12:00 PM on March 12th 2008.

```
<DisabledTime>
      <ISWebScheduler:WebSchedulerDisabledTime StartTime="2008-03-12
10:00:00" EndTime="2008-03-12 12:00:00">
      </ISWebScheduler:WebSchedulerDisabledTime>
</DisabledTime>
```



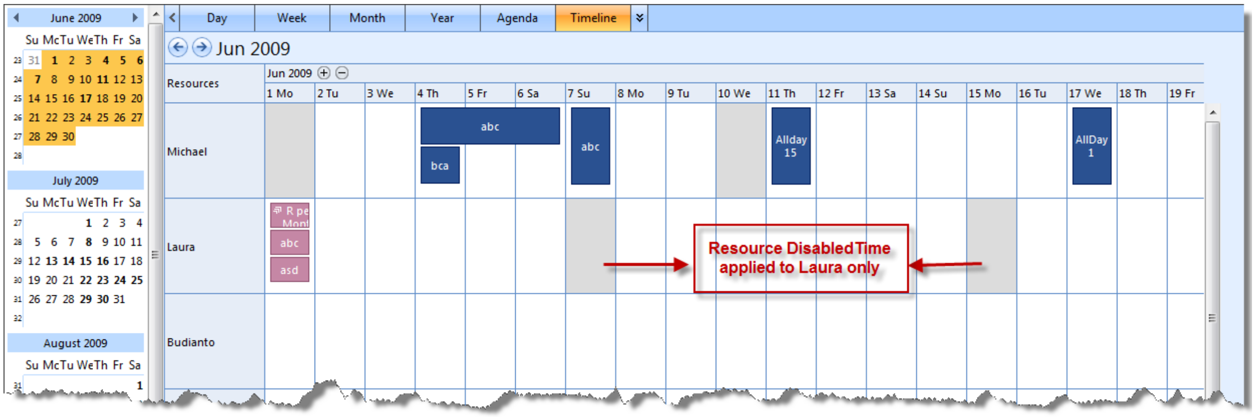
Global disabled hour area in Week view

2. Resource Disabled Time

   Resource Disabled Time is a resource specific disabled time. This option lets developer specify a specific rule for each resource. Resource disabled time can only be set from code behind in ***OnDataBound*** server-side event. The disabled time collection will then be populated to each resource object using ***PopulateDisabledTime*** server-side method.

   The following is the code snippet to add resource disabled time in **On*DataBound*** server side event.

```
protected void WebScheduler1_DataBound(object sender, 
ISNet.WebUI.WebScheduler.WebSchedulerDataBoundDataArgs e)
{
      if (e.Type == 
ISNet.WebUI.WebScheduler.WebSchedulerObjectType.Resource)
      {
            WebSchedulerResource data = e.DataObject as 
WebSchedulerResource;

            if (data.ResourceID == 1) //add disabled time to Resource 1
            {
                DataTable dt = new DataTable("DisabledTime");
                DataColumn dc = new DataColumn("StartTime", 
Type.GetType("System.DateTime"));
                DataColumn dc2 = new DataColumn("EndTime", 
Type.GetType("System.DateTime"));

                dt.Columns.Add(dc);
                dt.Columns.Add(dc2);

                DataRow dr = dt.NewRow();
                dr["StartTime"] = new DateTime(2009, 6, 1);
                dr["EndTime"] = new DateTime(2009, 6, 2);
                dt.Rows.Add(dr);

                dr = dt.NewRow();
                dr["StartTime"] = new DateTime(2009, 6, 10);
                dr["EndTime"] = new DateTime(2009, 6, 11);
                dt.Rows.Add(dr);

                WebSchedulerResource schedulerResource = e.DataObject 
as WebSchedulerResource;
                schedulerResource.PopulateDisabledTime(dt, "StartTime", 
"EndTime");
            }
        }
    }
```



Resource DisabledTime applied to Laura only

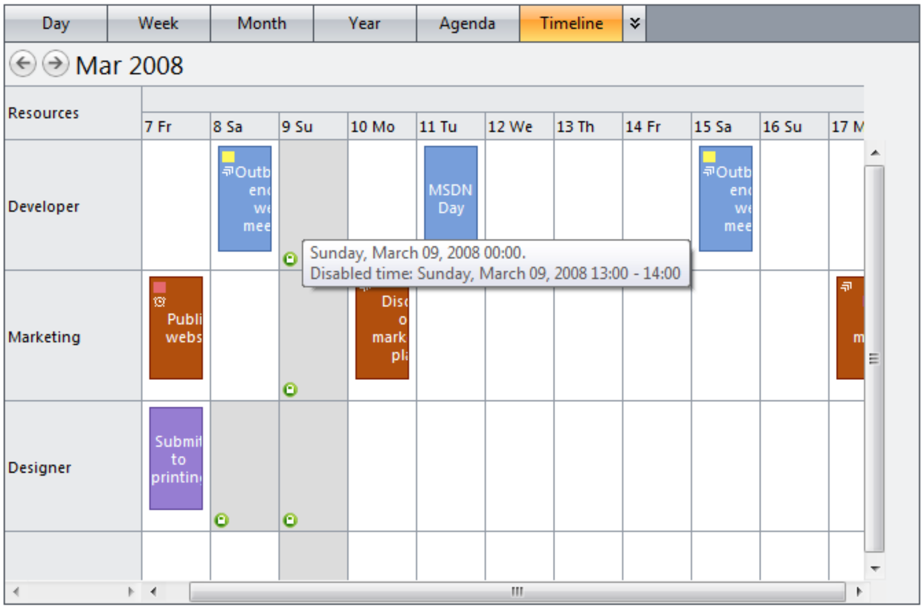
Resource specific Disabled Time in Timeline View mode

Resource disabled time can also be specified in Split view.

3. Partial Disabled Time

Partial disabled time is an extension of the disabled time concept in which not the entire day is disabled. Unlike the standard disabled time behavior, the Partial disabled time allows developer to specify certain disabled hours. To differentiate it from the global disabled time, a small padlock is added as a visual hint, which can be customized via ***ImagesSettings >> PartialTimelineDisabledTimeIcon*** property.

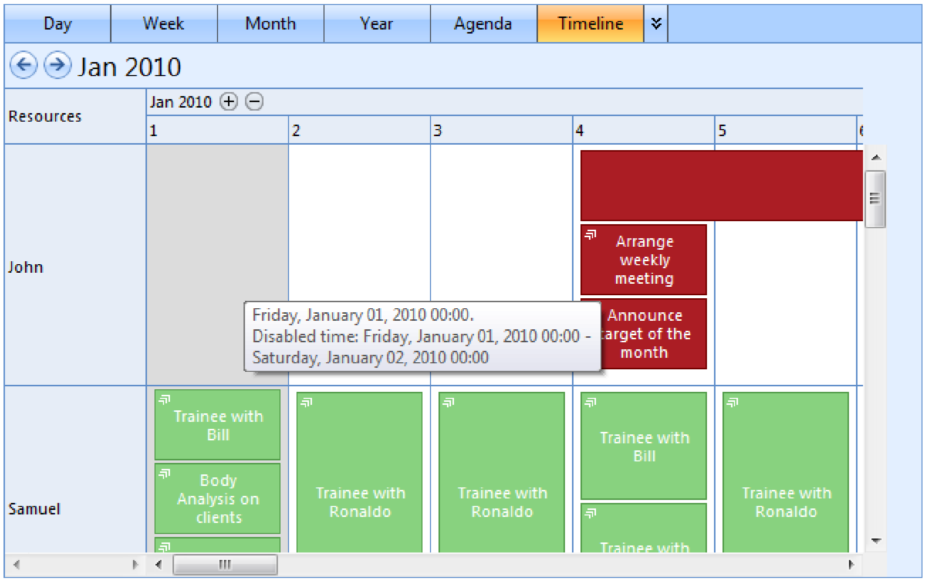
In the following screenshot, 01.00 PM – 02.00PM of March 9th 2009 is disabled. Notice the padlock indicator on the bottom left corner of the each cell. This indicate that the cells are partially disabled.



Sunday, March 09, 2008 00:00.
Disabled time: Sunday, March 09, 2008 13:00 - 14:00

Partial disabled time in Timeline view

In Timeline view, the partial disabled time range can be displayed as tooltip when user hovers through a disabled cell. To enable this feature, set ***ViewSettings >> TimelineView >> DisplayDisabledTimeTooltip*** property to *true*.



Friday, January 01, 2010 00:00.
Disabled time: Friday, January 01, 2010 00:00 - Saturday, January 02, 2010 00:00

Disabled time range is displayed as tooltip when disabled cell is hovered

Disabled Time Validation Concept

Validation is one of the core functionalities in WebScheduler Disabled Time feature. Cells are automatically disabled according to the specified date and time range. Editing functionalities, including the interactive callout, are also disabled. Update, move, and resize operations are prevented if the new date and time falls within the disabled time range.

Custom Disabled Time Validation

The built-in disabled time validation engine prevents event insertion that starts and ends within the disable time area. If the new event start at non-disabled cells and ends inside the disabled time area, the new event will be created and saved. The same thing applies to an event that starts before the disabled time range and ends after the disabled time range.

Developer is also given a complete freedom to manually code the validation logic. This process must be done in ***OnValidateDisabledTime*** client-side event. The return must be Boolean value, determining the custom validation result.

The parameters are *controlId*, *startTime*, *endTime*, *resourceID* and *elementType*. An event that returns true means that it is set within the disabled time range, and vice versa.

When ***OnValidateDisabledTime*** client-side event is implemented, WebScheduler will ignore the built-in disabled time validation and use the custom validation instead.

Create Disabled Time in Client-side

Disabled time can be specified in runtime. In client-side, ***WebSchedulerDisabledTime*** object should be created and added to the ***DisabledTime*** collection. After all is done, WebScheduler must be refreshed using ***Refresh*** function.

The following code will add disabled time on March 1st – 2nd 2010 and March 7th– 8th 2010.

```
function CreateDisabledTime()
{
      var s = ISGetObject("WebScheduler1");

      //add to global DisabledTime
      var dt = new WebSchedulerDisabledTime();
      dt.StartTime = new Date(2010, 2, 1);
      dt.EndTime = new Date(2010, 2, 2);

      s.DisabledTime.Add(dt);

      //add to resource DisabledTime
      var dt2 = new WebSchedulerDisabledTime();
      dt2.StartTime = new Date(2010, 2, 7);
      dt2.EndTime = new Date(2010, 2, 8);

      var resource = s.GetResourceById("1");
      resource.DisabledTime.Add(dt2);

      s.Refresh();
}
```

Overwrite Changes Support

Concurrency control is a data storing technique, controlling how data is read and changed in the store when multiple clients are accessing and manipulating the same data. ***ConflictDetection*** property determines whether to pass the new values to the data operation method or both old and new values are passed, thus determining which conflict detection approach is used when conflict occurs. This version also supports *OverwriteChanges* conflict detection.

There are two types of ***ConflictDetection***: *CompareAllValues* and *OverwriteChanges*. When developers create a new DataSet in Visual Studio 2005, *CompareAllValues* conflict detection is used by default. But in Visual Studio 2008, OverwriteChanges conflict detection is used by default when developers create a new DataSet.

WebScheduler 3.0 now supports both CompareAllValues and OverwriteChanges conflict detection options.

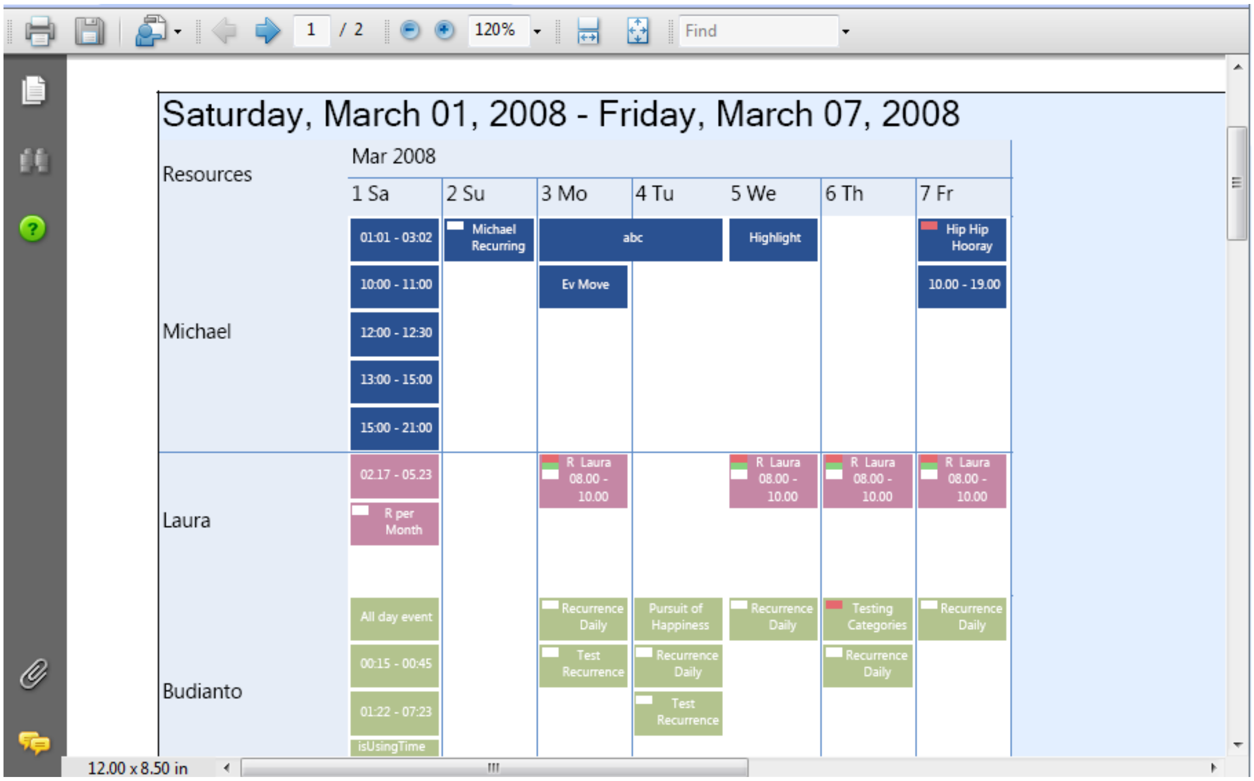
Several Enhancements in Timeline view

Real Time Operation editing

WebScheduler 3 adds numerous enhancements to its powerful Real Time Operation technology, such as optimized data editing operation and faster transaction performance. These enhancements enables partial rendering in Timeline view. Only effected events are updated – minimizing the time and overhead during data update.

Exporting

WebScheduler's exceptional exporting feature lets you easily disseminate your information in PDF format or syncing group calendars to every team member's Microsoft Outlook 2007. In version 3, WebScheduler improves its exporting functionality to support Timeline view. This comprehensive exporting covers all Timeline View modes, such as Day15Minutes, DayHourly, Week, Month, and Quarter.



Saturday, March 01, 2008 - Friday, March 07, 2008

Resources	Mar 2008 1 Sa	2 Su	3 Mo	4 Tu	5 We	6 Th	7 Fr
Michael	01:01 - 03:02 10:00 - 11:00 12:00 - 12:30 13:00 - 15:00 15:00 - 21:00	Michael Recurring	abc Ev Move		Highlight		Hip Hip Hooray 10.00 - 19.00
Laura	02.17 - 05.23 R per Month		R Laura 08.00 - 10.00		R Laura 08.00 - 10.00	R Laura 08.00 - 10.00	R Laura 08.00 - 10.00
Budianto	All day event 00:15 - 00:45 01:22 - 07:23 isUsingTime		Recurrence Daily Test Recurrence	Pursuit of Happiness Recurrence Daily Test Recurrence	Recurrence Daily	Testing Categories Recurrence Daily	Recurrence Daily

Timeline export to PDF format

Timeline exporting customization

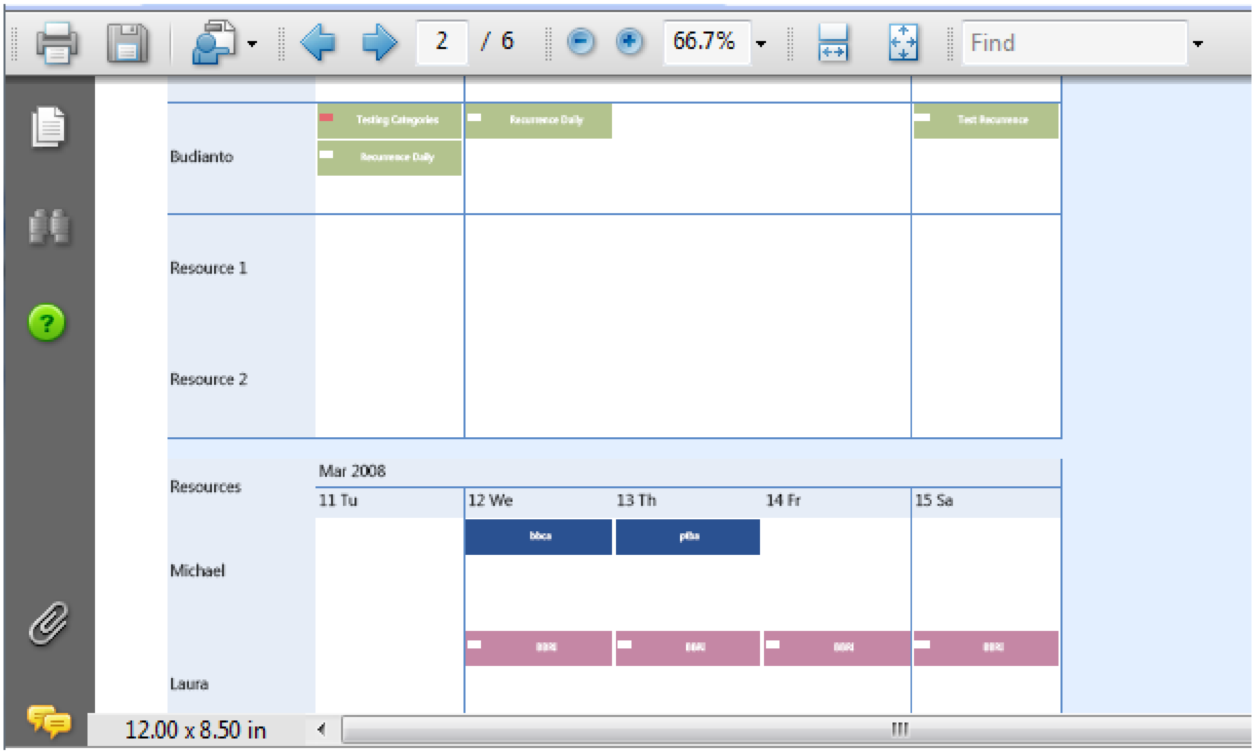
Customization on the Timeline exporting behavior is added to cover wider business scenarios, for example: customize the exported Timeline column's length.

This feature is very handy in certain scenarios that have many columns. Let's say we have 30 columns in Timeline Month view. While this won't be any problem when viewed in WebScheduler, exporting it will be a problem due to the paper size. To address this issue, WebScheduler allows developer to manually alter the column's length.

Below are properties that can be used to customize Timeline column's length in exporting.

1. QuarterExportColumnLength , to customize Timeline Quarter view column's length.
2. MonthExportColumnLength, to customize Timeline Month view column's length.
3. WeekExportColumnLength, to customize Timeline Week view column's length.
4. DayHourlyExportColumnLength, to customize Timeline DayHourly view column's length.
5. Day15MinutesExportColumnLength, to customize Timeline Day15Minutes view column's length.

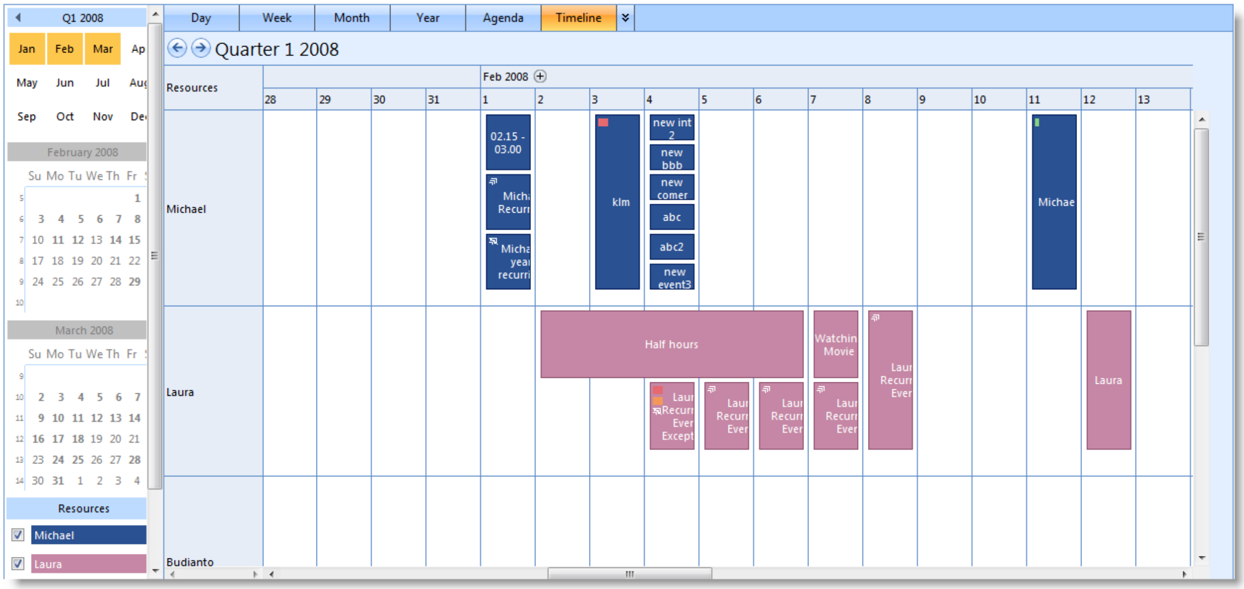
Timeline export customization properties can be found in ViewSettings > TimelineView.



Manually specifying the column's length when exporting a Timeline mode WebScheduler

Quarter mode

The new Quarter mode adds the broad view selections in Timeline View. It allows users to see their schedule in quarter-to-quarter period – providing larger perspective of event in a quarter.

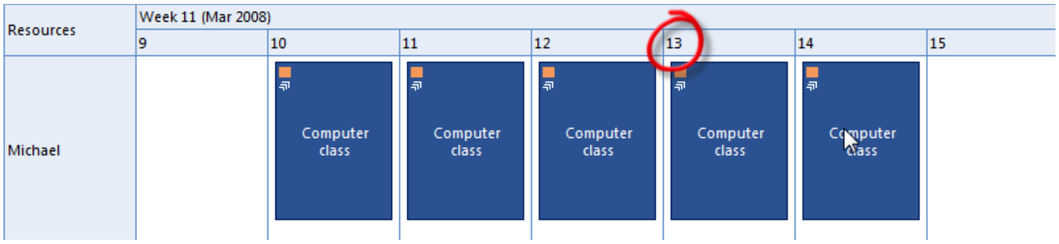


The new Quarter mode in Timeline view

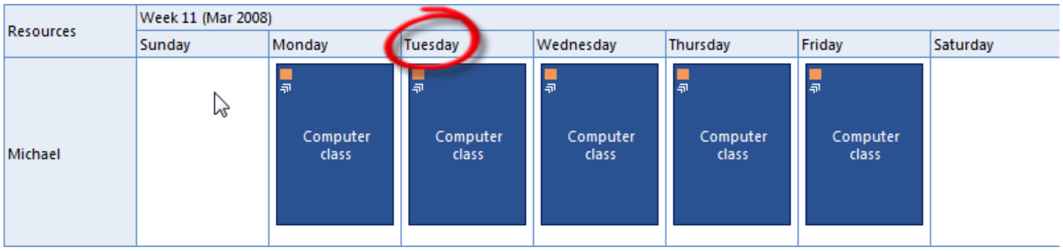
Customizable date header

WebScheduler 3 now introduces 6 various date header modes in Timeline view.

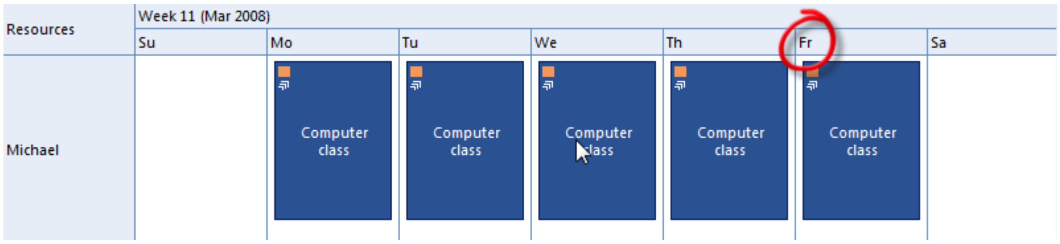
1. DateNumber



2. DayName

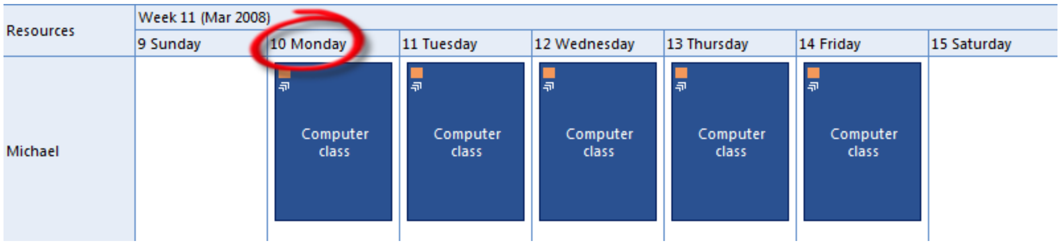


3. ShortestDayName



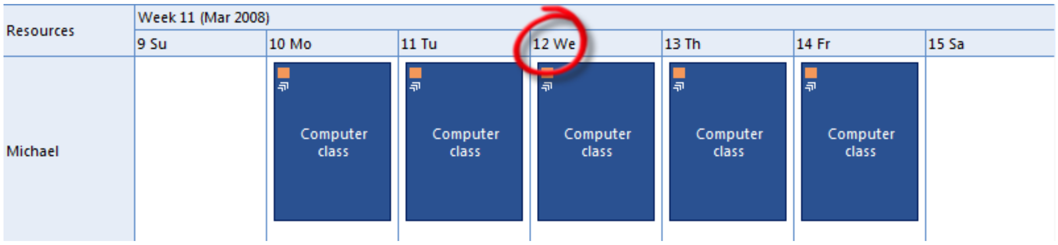
Resources	Week 11 (Mar 2008)						
	Su	Mo	Tu	We	Th	Fr	Sa
Michael		Computer class	Computer class	Computer class	Computer class	Computer class	

4. DateNumberAndDayName



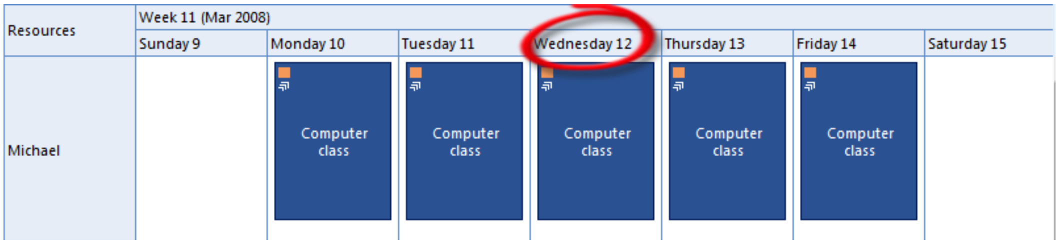
Resources	Week 11 (Mar 2008)						
	9 Sunday	10 Monday	11 Tuesday	12 Wednesday	13 Thursday	14 Friday	15 Saturday
Michael		Computer class	Computer class	Computer class	Computer class	Computer class	

5. DateNumberAndShortestDayName



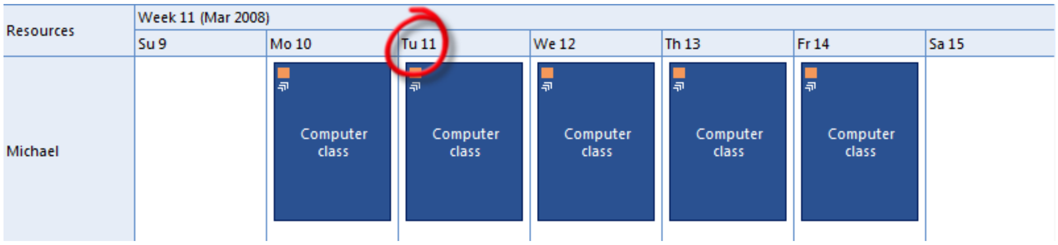
Resources	Week 11 (Mar 2008)						
	9 Su	10 Mo	11 Tu	12 We	13 Th	14 Fr	15 Sa
Michael		Computer class	Computer class	Computer class	Computer class	Computer class	

6. DayNameAndDateNumber



Resources	Week 11 (Mar 2008)						
	Sunday 9	Monday 10	Tuesday 11	Wednesday 12	Thursday 13	Friday 14	Saturday 15
Michael		Computer class	Computer class	Computer class	Computer class	Computer class	

7. ShortestDayNameAndDateNumber



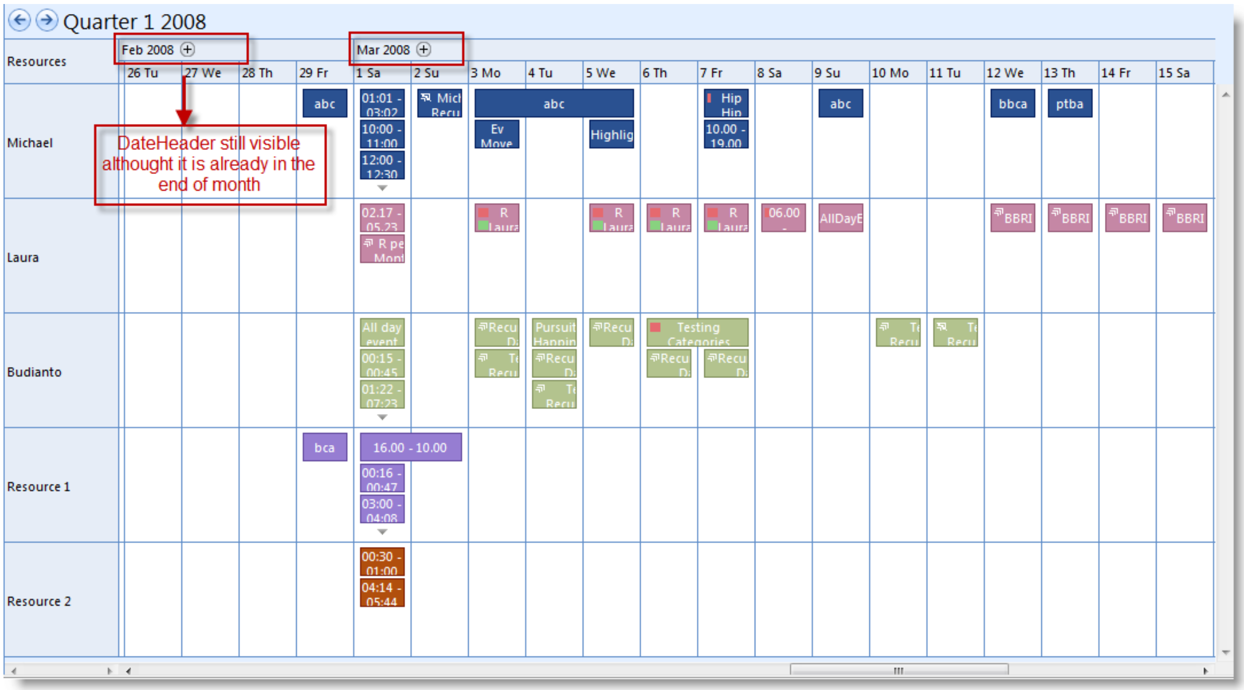
Resources	Week 11 (Mar 2008)						
	Su 9	Mo 10	Tu 11	We 12	Th 13	Fr 14	Sa 15
Michael		Computer class	Computer class	Computer class	Computer class	Computer class	

Keep date header visible

Timeline view is a powerful feature for displaying multiple progressive events - keeping track on every schedule and meeting the deadline in time fashion way. With its zooming feature, user can easily zoom-in/out using the +/- sign next to the date header.

Due to the nature of Timeline data presentation, user will need to scroll horizontally to browse all events. When it's scrolled all the way to the right, the date header will be hidden – making it difficult for user to perform zoom-in/out.

WebScheduler resolves this usability issue by introducing a new property, **ViewSettings >> TimelineView >> *KeepHeaderVisible***. When enabled, the date header position will be displayed in fixed position – enabling users to scroll all the way to the right while keeping the zooming button on their sight.

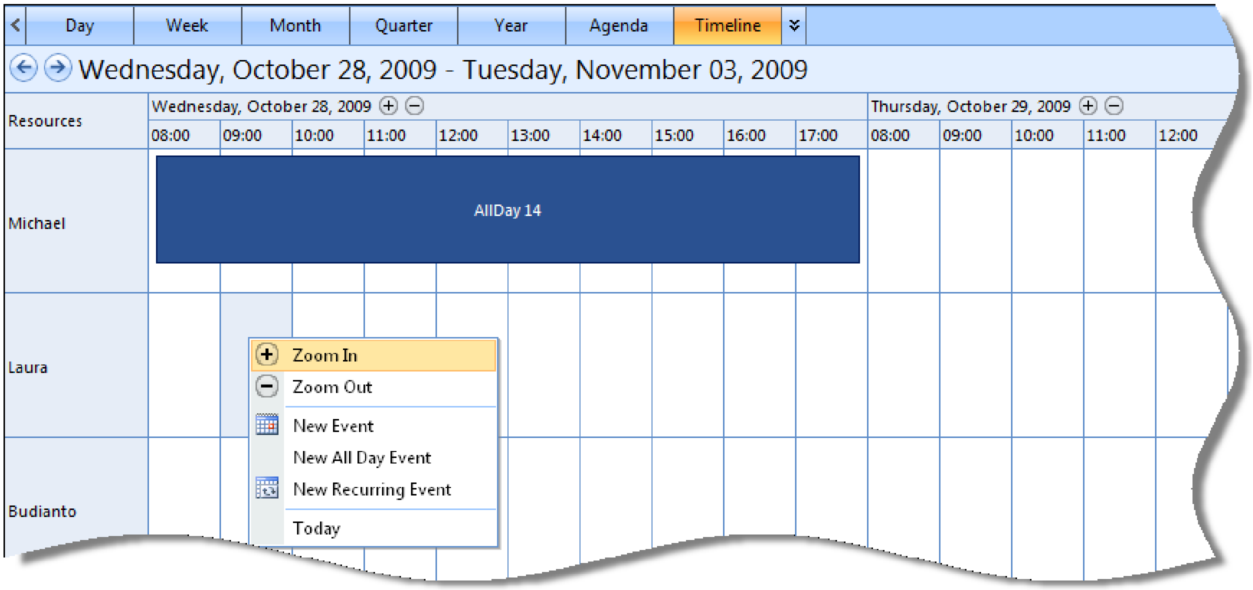


DateHeader still visible althought it is already in the end of month

Keep date header visible in Timeline View mode

Zoom-in and Zoom-out items in Context Menu

Context-menu provides hassle-free access to the most basic features in WebScheduler. WebScheduler 3 is enhanced further to support Timeline's zooming feature. In addition to the default zooming buttons on the date header, user can now access the zooming functions through context-menu.

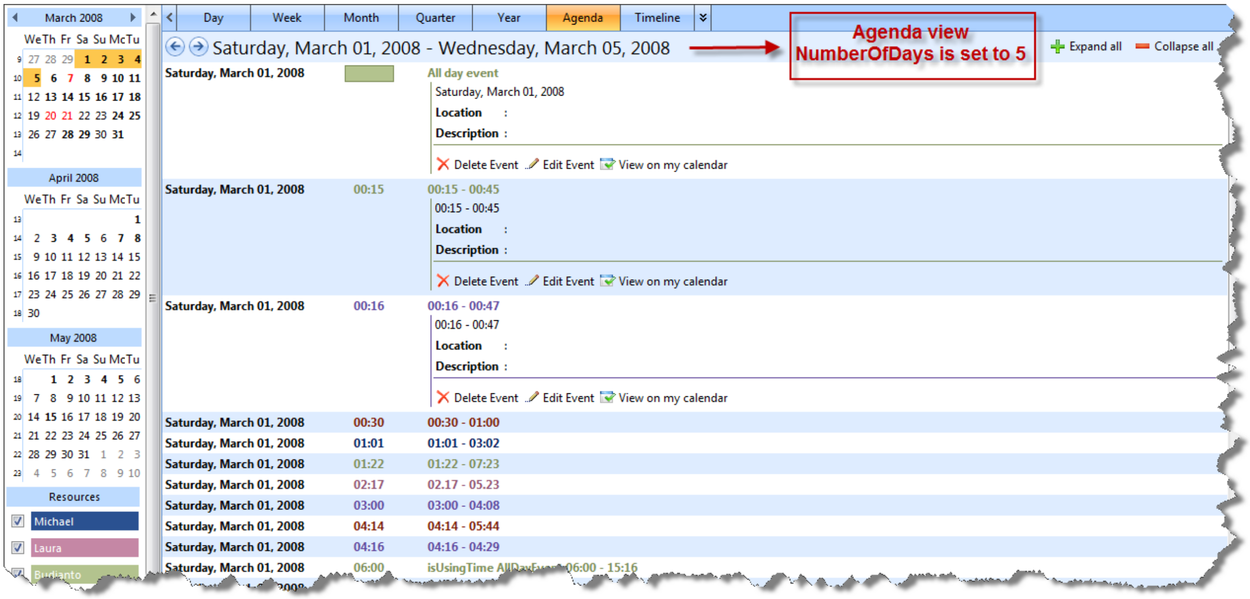


Zooming in and out via context-menu

Customizable number of days in Agenda view

This new enhancement gives developer more control in Agenda view mode to set the number of days.

By default the number of days included in the Agenda view is 10. Events that occur within the date ranges will be displayed in text-based information. The number of days selected in Agenda view can be customized in ***ViewSettings >> AgendaView >> NumberOfDays*** property.



Saturday, March 01, 2008 - Wednesday, March 05, 2008

Agenda view
NumberOfDays is set to 5

Saturday, March 01, 2008	🟩	All day event Saturday, March 01, 2008 Location : Description : Delete Event Edit Event View on my calendar
Saturday, March 01, 2008	00:15	00:15 - 00:45 00:15 - 00:45 Location : Description : Delete Event Edit Event View on my calendar
Saturday, March 01, 2008	00:16	00:16 - 00:47 00:16 - 00:47 Location : Description : Delete Event Edit Event View on my calendar
Saturday, March 01, 2008	00:30	00:30 - 01:00
Saturday, March 01, 2008	01:01	01:01 - 03:02
Saturday, March 01, 2008	01:22	01:22 - 07:23
Saturday, March 01, 2008	02:17	02.17 - 05.23
Saturday, March 01, 2008	03:00	03:00 - 04:08
Saturday, March 01, 2008	04:14	04:14 - 05:44
Saturday, March 01, 2008	04:16	04:16 - 04:29
Saturday, March 01, 2008	06:00	isUsingTime AllDayEvent 06:00 - 15:16

Customizable maximum height of all-day area

All-day events are displayed in all-day area. The height of all-day area is adjusted to the number of events in all-day area. If a lot of all-day event occurs in the active view, the all-day area will be enlarged to the show all events, often exceed the visible view port.

In current version, the all-day area has been restructured to have a maximum height. If the total number of events has exceeded the maximum height specified, vertical scrollbar will appear.

This configuration can be customized in ***LayoutSettings >> MaxAllDayEventAreaHeight*** property.

Client-side function to retrieve cell element(s)

In scenarios where specific cell(s) need to be retrieved, specific client side functions are added to WebScheduler.

Retrieve a cell element

To retrieve a single cell element, use ***GetCellElement*** client-side function. Three parameters used in the function are:

- *date*: the date of the cell to be retrieved.
- *resourceId* (optional): specifies the resource id of the cell to be retrieved. This parameter is used in Split and Timeline view in particular. When specified, the cell of the specified row only will be retrieved.
- *getAllDayArea (optional)*: determines whether hour area or all-day area should be retrieved. This parameter is used in Day, Week, and Split view only.

The following code will set red background color to 18th of March 2008 cell element. In Split and Timeline view, the cell of resourceId 2 will be set.

```
function GetCellElementByDate()
{
      var s = ISGetObject("WebScheduler1");
      var el = s.GetCellElement(new Date(2008, 2, 18, 9, 10, 10), 2, true);

      if (el != null)
            el.style.backgroundColor = "red";
}
```

The following code will set red background color to 18th of March 2008 06:00 AM cell element. In Split view and Timeline view, the cell of resourceId 2 will be set.

```
function GetCellElementByDateTime()
{
      var s = ISGetObject("WebScheduler1");
      var el = s.GetCellElement(new Date(2008, 2, 18, 6, 0, 0), 2, false);

      if (el != null)
      {
            if (s.ViewSettings.SelectedViewMode == "Timeline")
                  el.style.backgroundColor = "red";
            else
                  el.children[0].style.backgroundColor = "red";
      }
}
```

Retrieve multiple cell elements

To retrieve multiple cell elements, you can use ***GetCellElements([WebSchedulerDateRange collection])*** client-side function. You can specify multiple date range as WebSchedulerDateRange collection and GetCellElements will return the cell elements of the date range collection.

WebSchedulerDateRange has 4 properties:

- StartTime: specifies the start time of the date range
- EndTime: specifies the end time of the date range
- ResourceID: specifies the resource id of the cell to be retrieved. This parameter is used in Split and Timeline view in particular. When specified, the cell of the specified row only will be retrieved.
- DatePartType: specifies which part of the date should be used to retrieve the cell elements. The value options of this property are Date, DateTime, and Time.

When *Date* type is used, only the date part will be used to retrieve the cells. In Day, Week, and Split view, this indicates that all-day area cells will be retrieved. The following code will retrieve cell elements that start from 18th of March 2008 to 20th of March 2008. Although the start time and end time includes time part, only the date part will be used in the retrieving process.

```
function GetCellElementsByDate()
{
      var s = ISGetObject("WebScheduler1");

      var arr = new ISArray();
      arr.Add(new WebSchedulerDateRange(new Date(2008, 2, 18, 6, 0, 0), new
Date(2008, 2, 20, 7, 0, 0), 2, "Date"));

      var elements = s.GetCellElements(arr);
      SetBackgroundColor(elements);
}
```

When *DateTime* type is used, the date and time part will be used to retrieve the cells. In Day, Week, and Split view, this indicates that hour area cells will be retrieved. The following code will retrieve cell elements that start from 19th of March 2008 06:00 AM to 20th of March 2008 08:00 AM.

```
function GetCellElementsByDateTime()
{
      var s = ISGetObject("WebScheduler1");

      var arr = new ISArray();
```

```
     arr.Add(new WebSchedulerDateRange(new Date(2008, 2, 19, 6, 0, 0), new
Date(2008, 2, 20, 8, 0, 0), 2, "DateTime"));

     var elements = s.GetCellElements(arr);
     SetBackgroundColor(elements);
}
```

When *Time* type is used, only the time part will be used to retrieve the cells. In Day, Week, and Split view, cell elements of all dates in the active view within the date range will be retrieved. The following code will set the background color of cell elements that starts from 10:00 AM to 12: 00 PM, no matter what the date are. This is especially useful when you want to retrieve cells of specific time range regardless of its date.

```
function GetCellElementsByTime()
{
      var s = ISGetObject("WebScheduler1");

      var arr = new ISArray();
      arr.Add(new WebSchedulerDateRange(new Date(2008, 2, 19, 10, 0, 0), new
Date(2008, 2, 20, 12, 0, 0), 2, "Time"));

      var elements = s.GetCellElements(arr);
      SetBackgroundColor(elements);
}
```

Timeline view

In Timeline view, DatePartType parameter will not be included in the cell retrieval process. The date range input will be adjusted to the active Timeline view mode. For example: when Week, Month, or Quarter mode in Timeline view is active, only the date part will be used to retrieve the cell elements.

```
function GetCellElementsByTime()
{
      var s = ISGetObject("WebScheduler1");

      var arr = new ISArray();
      arr.Add(new WebSchedulerDateRange(new Date(2008, 2, 19, 10, 0, 0), new
Date(2008, 2, 20, 12, 0, 0), 2));

      var elements = s.GetCellElements(arr);
      SetBackgroundColor(elements);
}
```

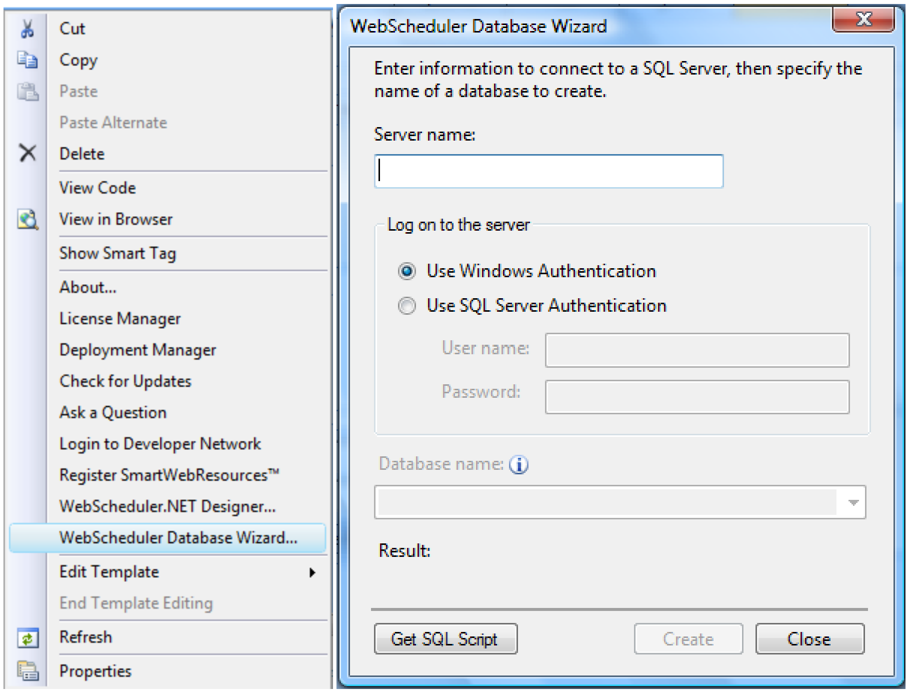
However, when DayHourly or Day15Minutes mode in Timeline view is active, the date and time part will be used to retrieve the cell elements.

WebScheduler Database Wizard

Stepping-up to version 3, WebScheduler introduces "WebScheduler Database Wizard", an automatic tool for WebScheduler's database creation. By simply specifying the SQL Server name and the database name, the wizard automatically generates the database. Next is to bind it to a WebScheduler instance in Visual Studio.

Right click on a WebScheduler instance and select “WebScheduler Database Wizard” option to use this embedded tool.

Follow the onscreen instruction and you’ll get your database in no time.



Enter information to connect to a SQL Server, then specify the name of a database to create.

Server name:

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

User name:

Password:

Database name:

Result:

WebScheduler Database Wizard

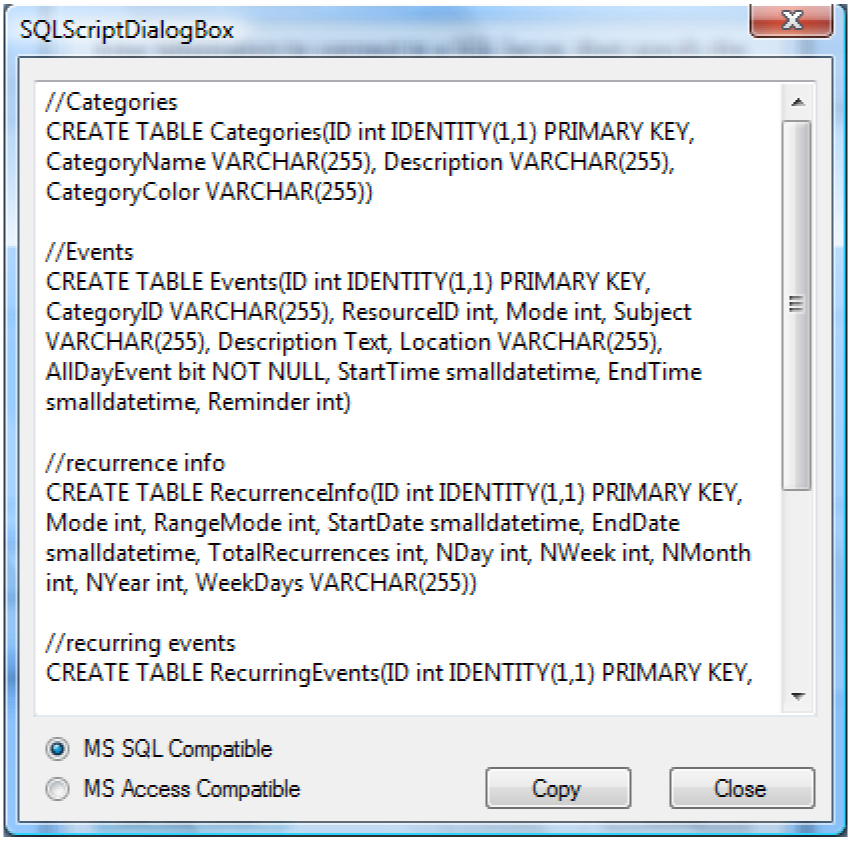
Database name drop down list

When a successful connection to the designated SQL Server has been made, you specify a new database by typing the name on the textbox – or you can select one of the existing database from the drop down list. Press “Create” and all required tables for WebScheduler will be defined automatically.

Get SQL Script button

A more advanced option to obtain SQL script for the table definition is also provided. Developer can use the Get SQL Script button to obtain the complete SQL statements and apply it in their own database, for example a Microsoft Access database.

Below is a sample when developer clicks the *Get SQL Script* button



```
//Categories
CREATE TABLE Categories(ID int IDENTITY(1,1) PRIMARY KEY,
CategoryName VARCHAR(255), Description VARCHAR(255),
CategoryColor VARCHAR(255))

//Events
CREATE TABLE Events(ID int IDENTITY(1,1) PRIMARY KEY,
CategoryID VARCHAR(255), ResourceID int, Mode int, Subject
VARCHAR(255), Description Text, Location VARCHAR(255),
AllDayEvent bit NOT NULL, StartTime smalldatetime, EndTime
smalldatetime, Reminder int)

//recurrence info
CREATE TABLE RecurrenceInfo(ID int IDENTITY(1,1) PRIMARY KEY,
Mode int, RangeMode int, StartDate smalldatetime, EndDate
smalldatetime, TotalRecurrences int, NDay int, NWeek int, NMonth
int, NYear int, WeekDays VARCHAR(255))

//recurring events
CREATE TABLE RecurringEvents(ID int IDENTITY(1,1) PRIMARY KEY,
```

● MS SQL Compatible
○ MS Access Compatible

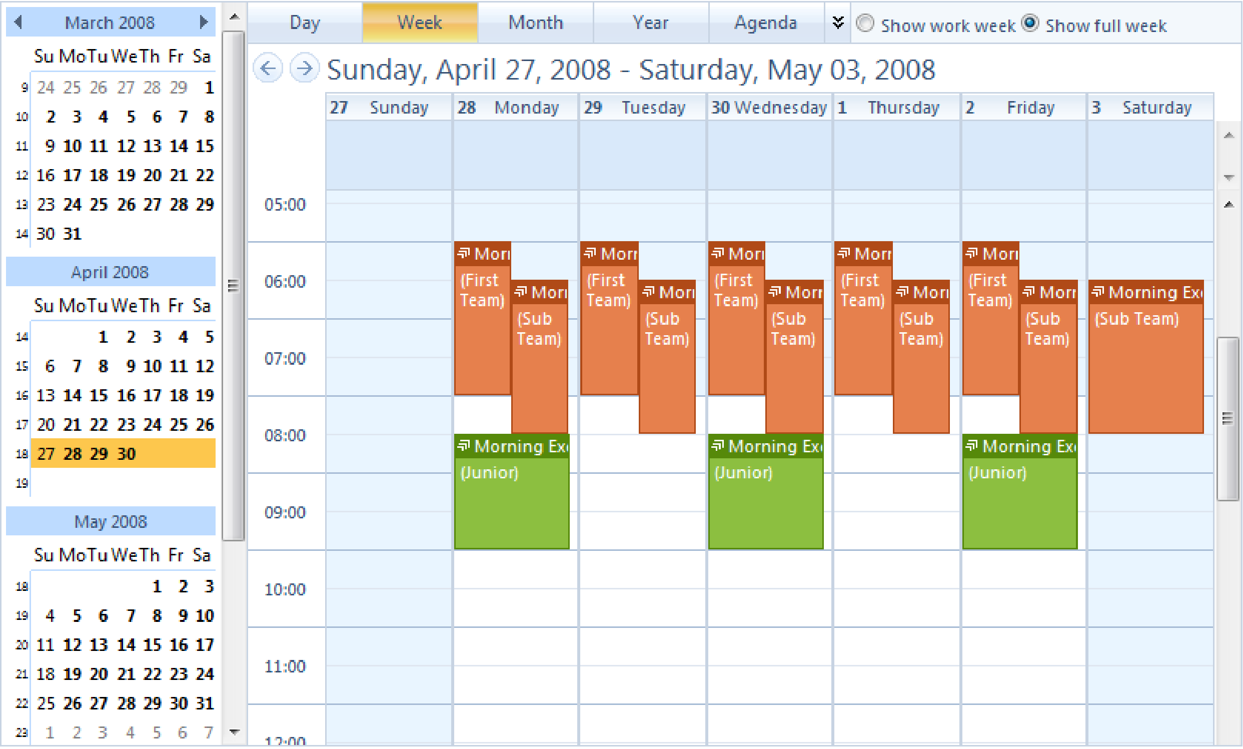
Get SQL Script button

Notice that there are 2 options; “MS SQL Compatible” and “MS Access Compatible”. User can choose the one that suit them best.

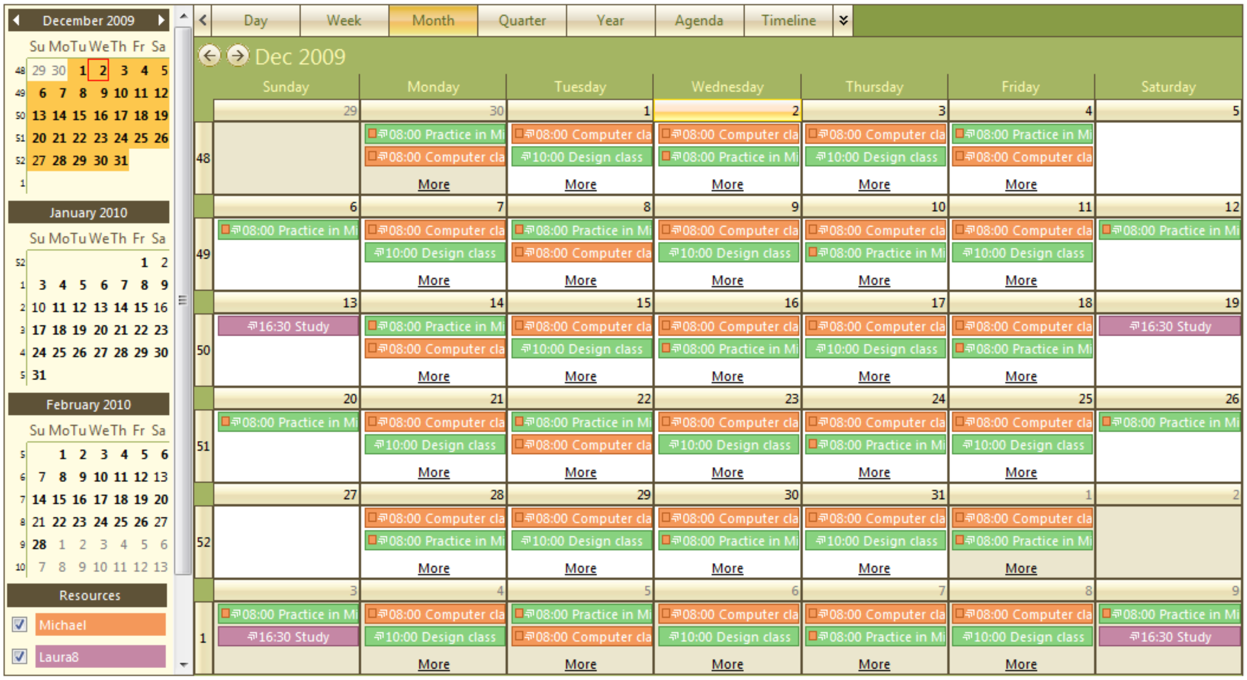
Theme and Style

New Themes

In addition to existing default style modes, new themes are added to WebScheduler, Win7 and OliveGreen. When selected, Win7 theme will be applied to all elements of WebScheduler.



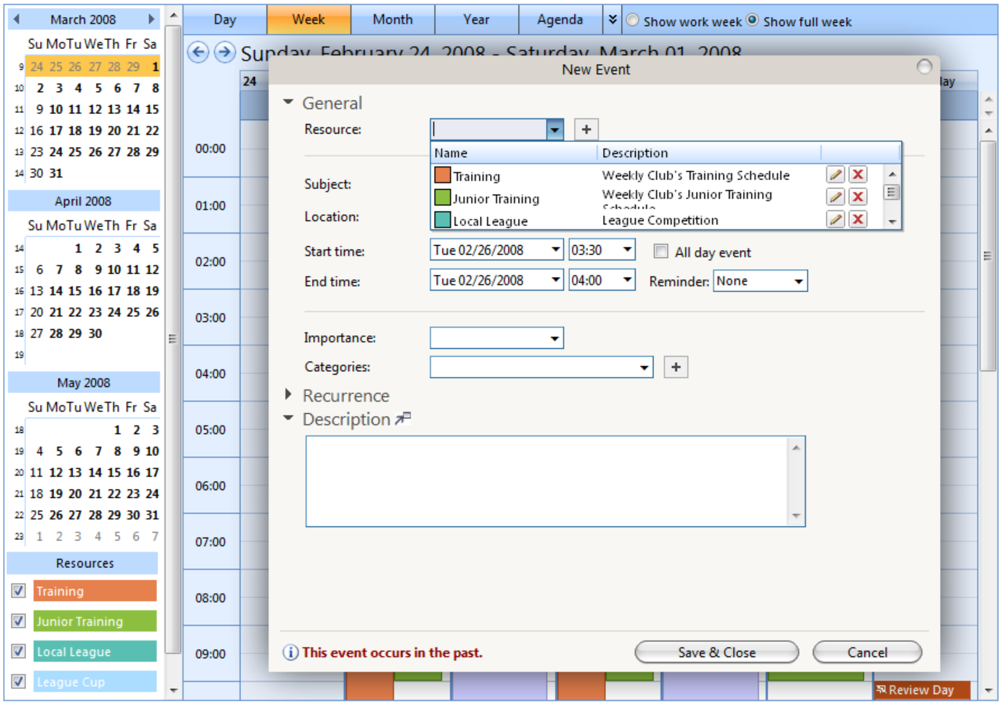
Win7 Default Style in Week view



Olive Green Default Style in Month view

Editing Form Style

When data editing feature is enabled in WebScheduler, users are able to insert, update, and delete events, using callout or built-in editing form. In this version, WebScheduler introduces a new property, ***EditingFormStyle***, which determines the styles applied to built-in editing form. Three style options for built-in editing form are: Standard style, Minimalist style, and Win7 style.



Name	Description
🟧 Training	Weekly Club's Training Schedule
🟩 Junior Training	Weekly Club's Junior Training Schedule
🟦 Local League	League Competition

Start time: Tue 02/26/2008 03:30 ☐ All day event

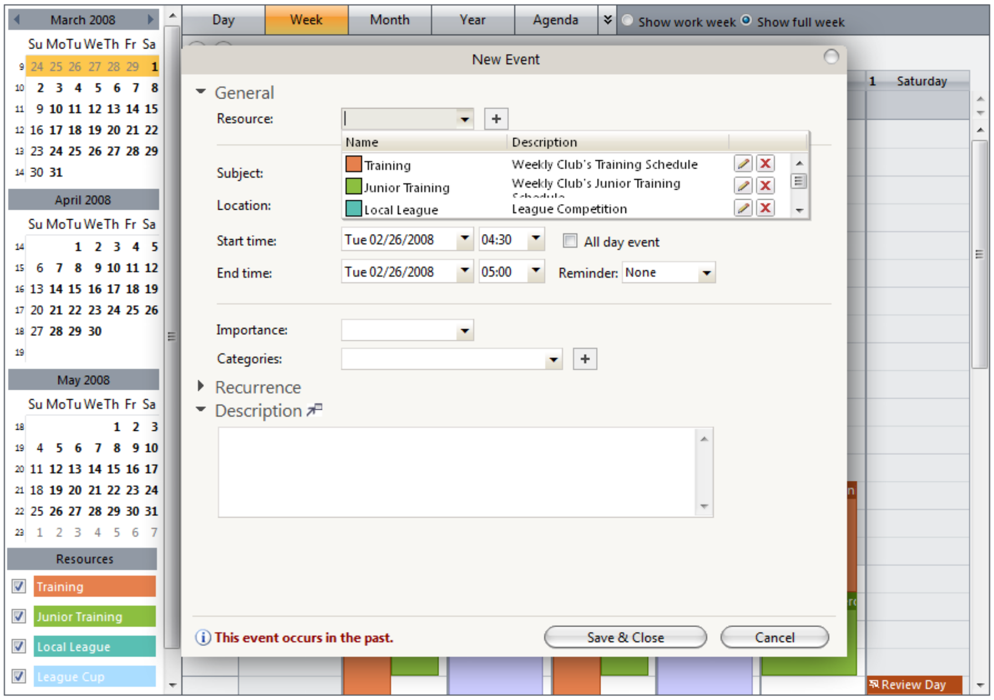
End time: Tue 02/26/2008 04:00 Reminder: None

Importance:

Categories:

This event occurs in the past.

Standard editing form style



March 2008

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

April 2008

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

May 2008

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

☑ Training
☑ Junior Training
☑ Local League
☑ League Cup

New Event

General

Resource:

Name	Description
Training	Weekly Club's Training Schedule
Junior Training	Weekly Club's Junior Training Schedule
Local League	League Competition

Subject:

Location:

Start time: Tue 02/26/2008 04:30 □ All day event

End time: Tue 02/26/2008 05:00 Reminder: None

Importance:

Categories:

Recurrence

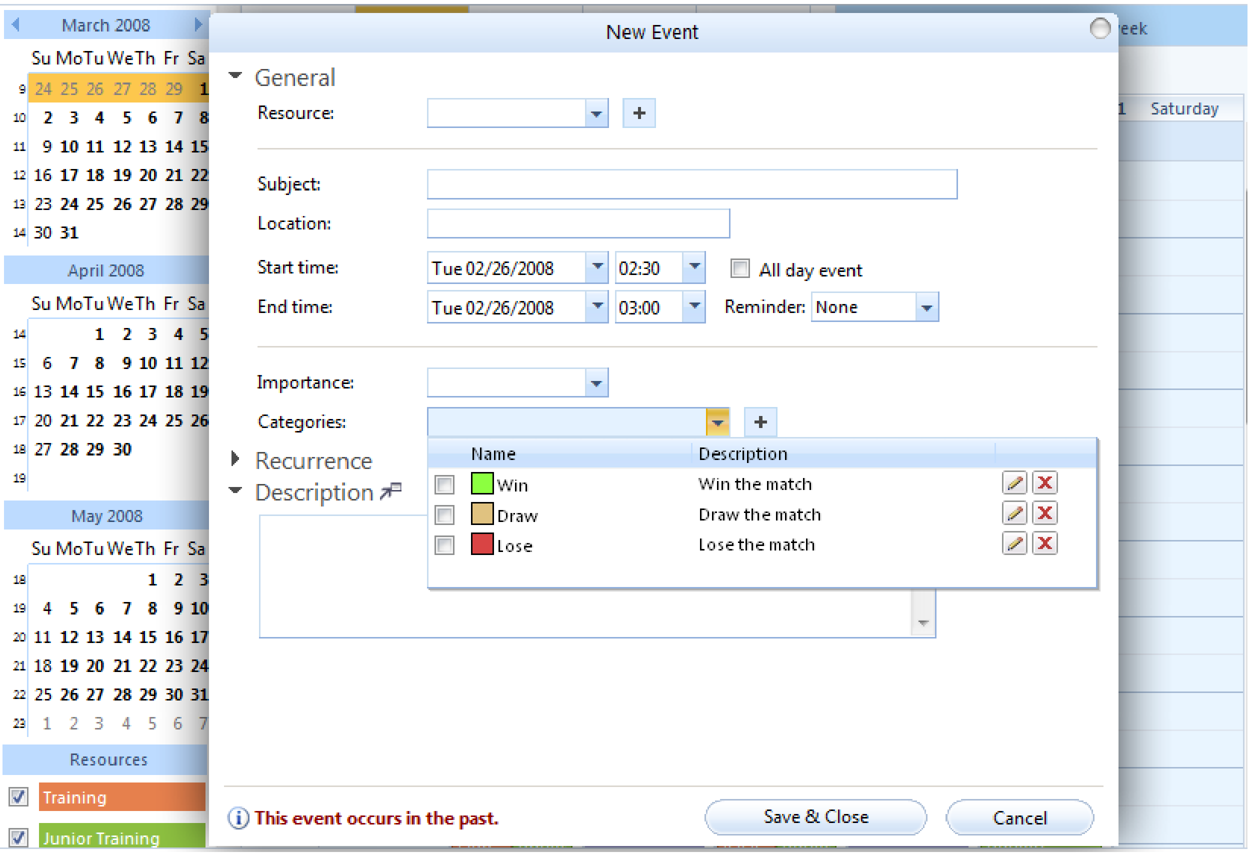
Description

This event occurs in the past.

Save & Close

Cancel

Minimalist editing form style



Resource:

Subject:

Location:

Start time: Tue 02/26/2008 02:30 □ All day event

End time: Tue 02/26/2008 03:00 Reminder: None

Importance:

Categories:

	Name	Description
□	🟩 Win	Win the match
□	🟫 Draw	Draw the match
□	🟥 Lose	Lose the match

ⓘ This event occurs in the past.

Win 7 editing form style