

OCal v2.0.0

by Brainy Data Limited

About OCal

Latest Changes

For a list of the latest changes please scroll to the end of this chapter or [click here](#).

Introduction

OCal is a much enhanced calendar component that replaces the calendar component provided by Raining Data. In the main it allows you to attach a list of events and view these events in various display modes, such a single day, week, two weeks, day plan and the original month view.

OCal implements drag & drop for moving events and the sizing of the events directly in the view. The appearance is somewhat based on the Mac OSX iCal software.

The following is a brief description of the various parts of this software.

- The calendar external. Place this file in your *External* folder.
- The example library calendar.lbs. The library implements a practical example relating to the use of the calendar.
- If you are trying a demo version of the software you should have requested a demo serial number from Brainy Data.

Installing the Software

There are a number of components to install and the component names vary between platforms. As a general rule, the downloaded folder containing the software will be organised so that you may follow these generic steps.

The downloaded folder contains folders for different versions of studio, i.e. *studio430*, *studio500*, *studio520*, *studio600* and *studio610*.

- Open the appropriate Omnis Studio folder. Always use the latest version that is not later in version than your version of Omnis Studio. For example, for Studio 5.0.2 you would use components from the *studio500* folder and **not** the *studio520* folder.

The Omnis Studio folder may contain four further folders called *JSCOMP*, *WEBCOMP*, *XCOMP* and *Web Client*. These may or may not be post fixed with a three letter platform identifier, i.e. *_mac*, *_win* or *_lin*.

- Copy the components from inside JSCOMP, WEBCOMP and XCOMP to your Omnis tree. You will find identical named folders inside the Omnis tree. On Mac OSX these can be found inside the *Omnis bundle->Content->MacOS*.

- To install the web client component you can load it into your Omnis download manager, this tool is available of the *Tools Menu->Web Client Tools*.
- For software that supports the Omnis Java script platform, there will be an additional folder named *html*. This folder will contain additional java scripts and modification to *jsctempl.htm* that require integration with your Omnis tree. Please refer to the accompanied *readme.txt* file

You can open the example library directly from the *OCal* folder.

Deploying your software

Please refer to the license agreement for rules on deployment.

Documentation

This documentation describes the functionality provided by the external.

Table of Contents

History

Version	Description
2.0.0	<p>New properties:</p> <ul style="list-style-type: none"> - \$viewmode replaces \$dayview. The calendar control now provides a number of new viewing modes. - \$dayviewalldayheight specifies height of all-day panel in pixels. - \$dayviewalldayheightmax specifies the maximum height for the all-day pane. - \$dayviewheadingusedaycolors if true, the day header colours are matched with the day colours. - \$templatetimelines for specifying one or more time lines. <p>Other Improvements:</p> <ul style="list-style-type: none"> - Visual indicators for events that are not in view. - The nowrap tag for display templates now causes ellipses to be displayed when the text does not fit horizontally.
1.7.0	<p>New version numbers: OCal version 1.7.0 supercedes version 10.6.x. Please read the release notes for a detailed explanation.</p>
10.6.0	<p>New properties:</p> <ul style="list-style-type: none"> \$isoweektext - If specified, OCal will display the ISO 8601 week number in both the day and month views.
10.5.5	<p>New properties:</p> <ul style="list-style-type: none"> \$repeatcontent - if enabled content is displayed repeatedly when the event wraps in day view or month views.
10.5.0	<p>New properties:</p> <ul style="list-style-type: none"> \$holidaylist - list for specifying public holidays. \$vertscroll - property for showing/hiding the vertical scroll bar (day view only). \$vscroll - specifies the current vertical scroll position (day view only). evScroll - new event message. This event is send when the control is scrolled vertically. \$dayviewtimecolumnwidth - width of the time column in pixels \$makedatareference() - static library method for creating a special list reference for use with OCal. \$disablescreenupdates() - static library method for disabling screen updates. \$enablescreenupdates() - static library method for enabling screen updates. \$dayviewnowrap - Property for turning off wrapping of text within an event box. \$template... - set of new properties for controlling event content, time column and scales. Please refer to the new chapter OCal Templates for a detailed description of this new feature.

10.4.0	<p>New properties:</p> <p>\$ddallowotherfields - If true, calendar will allow drops from other fields of any type (default is false)</p> <p>\$ddhilitellday : If true, all-day cells are enabled and highlighted during drag & drop</p> <p>\$ddhiliteworking : If true, working hours time slots are enabled and highlighted during drag & drop</p> <p>\$ddhilitenonworking : If true, non-working hours time slots are enabled and highlighted during drag & drop</p> <p>\$mousedate - returns the date and time under the mouse, adjusted according to \$dayviewsnapminutes</p> <p>\$mouseallday - returns true if the mouse is over the all-day pane</p> <p>Page 2 out of 3</p> <p>\$mouseevent - returns the line number of the event in the list currently under the mouse, zero if the mouse is not over an event</p>
10.3.0	<p>New properties:</p> <p>\$columnallday : if set calendar allows all-day events</p> <p>\$noeventborder : removes the extra border when showing event boxes</p> <p>\$dayviewdividercolor : fill colour for all-day / day-view divider</p> <p>\$dayviewtimecolumncolor : fill colour for time column in day view</p> <p>\$dayviewtimecolumntextcolor :text colour for time column in day view</p> <p>\$dayviewdeadcolumncolor : fill colour for dead area above scroll bar</p> <p>New event:</p> <p>evDeleteEvent : sent to field when user presses the delete/backspace key</p> <p>New event parameters:</p> <p>pAllDay : used with evCreateEvent, if true event was created in all-day pane</p>
10.2.0	<p>New properties \$dayviewminutescale and \$dayviewtimescale.</p> <p>New property \$weekenddays.</p> <p>New property \$dayviewvposminutes.</p>

Introduction

Overview

The OCal software consists of the following

- The OCal example library.
- The external XCOMP component libraries contain the main Omnis window and report components.
- The external WEBCOMP component libraries contain the Omnis remote form component.

Examples

The example library implements a main example window that demonstrates the use of the enhanced calendar component and its features.

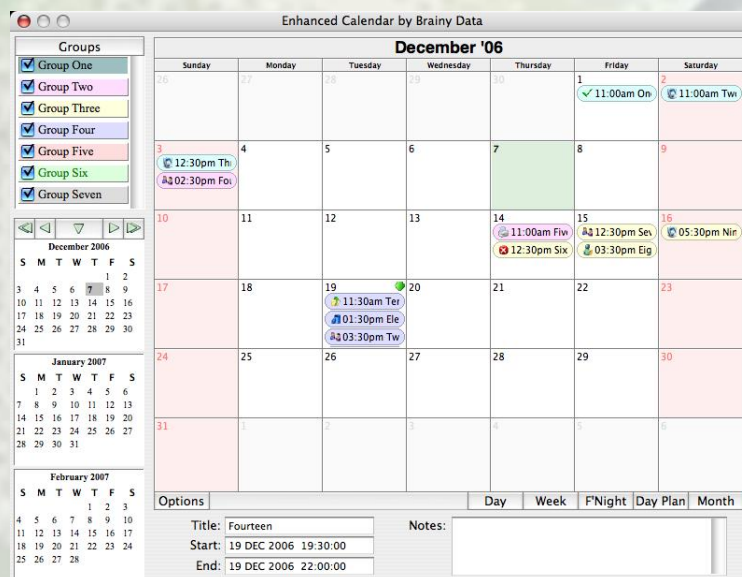
External Libraries

The external libraries provide the calendar objects for use in your Omnis windows and reports. For a description of the calendar properties and functions please read the chapter External Component Reference

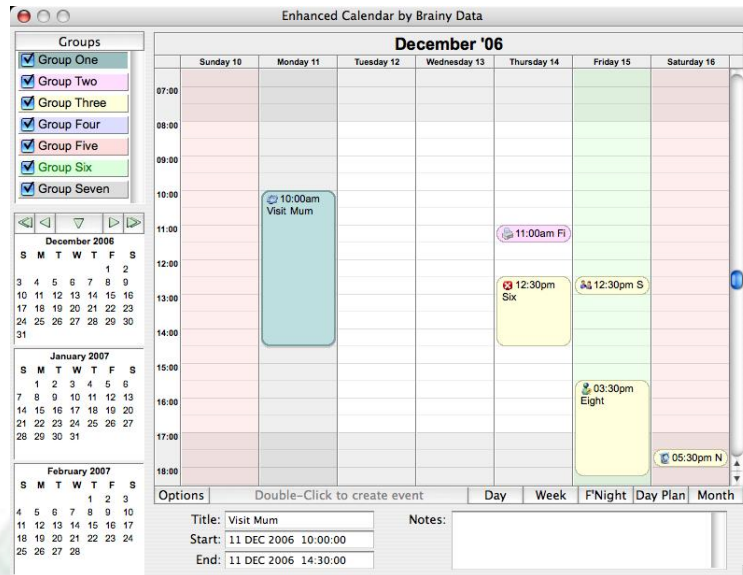
OCal main Features Summary

The enhanced calendar allows you to display and manipulate a list of events. The list can come from any source but must provide columns with start date/time and optionally an end date/time for each event. You can additionally provide columns for event colours, icons, text, layers, etc. The Chapter Designing OCal describes these features in more detail.

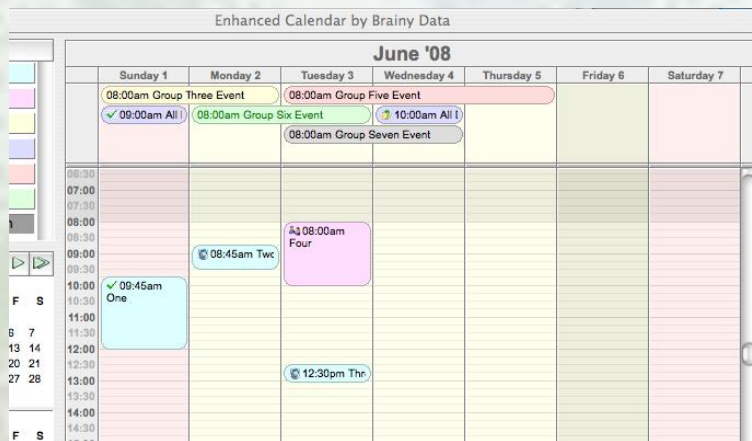
The OCal month view displaying events for an entire month.



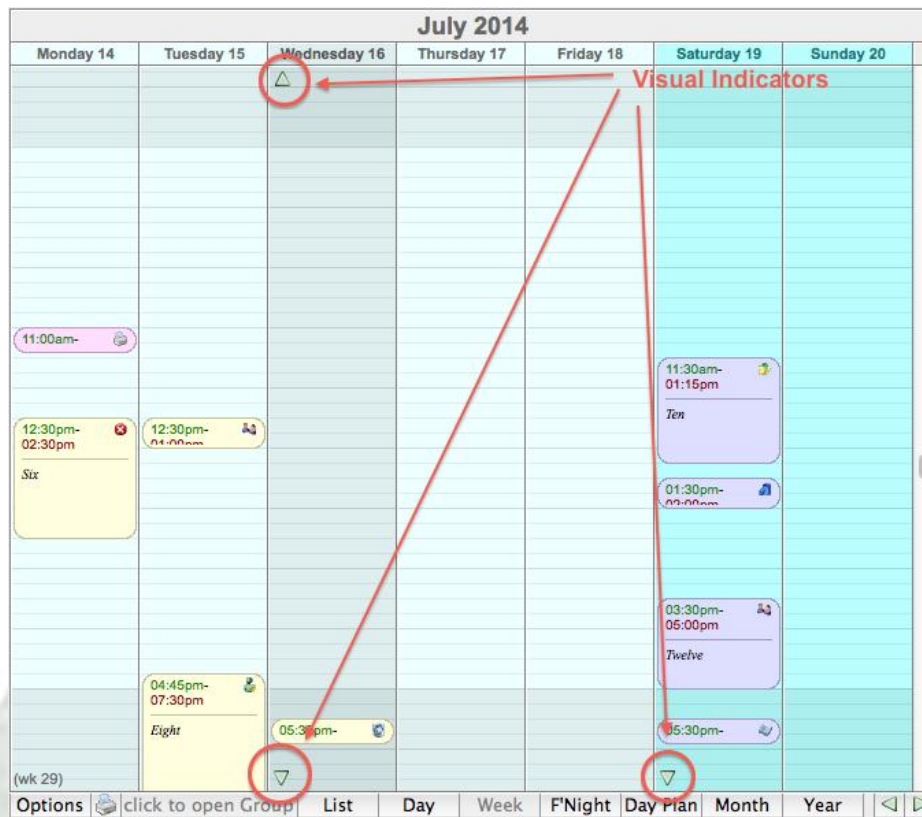
The OCal day view mode can display any number of days from 1 day to an entire year (if the screen size will allow it).



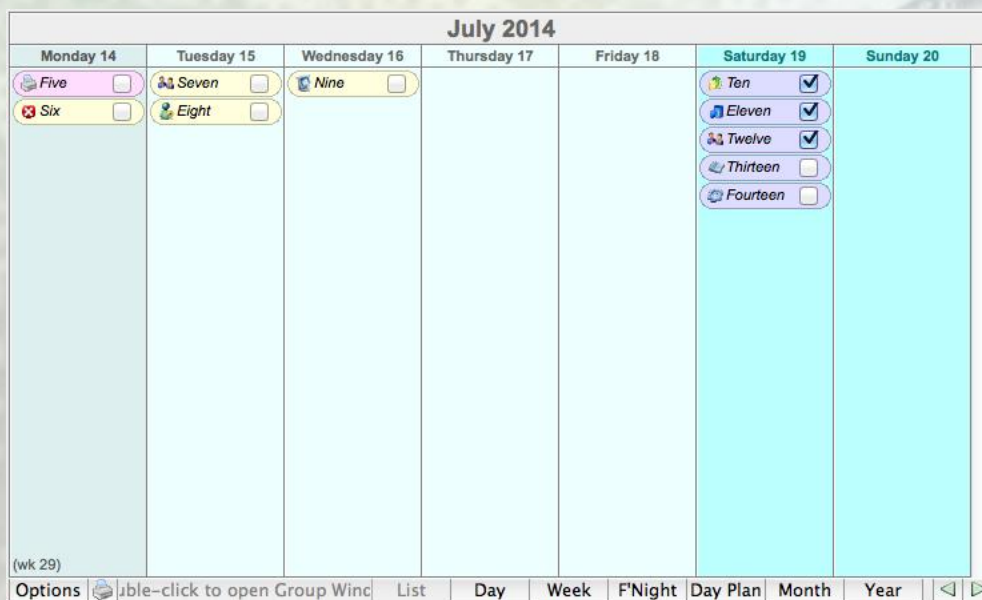
OCal displays all-day-events in a separate pane above the standard day-view.



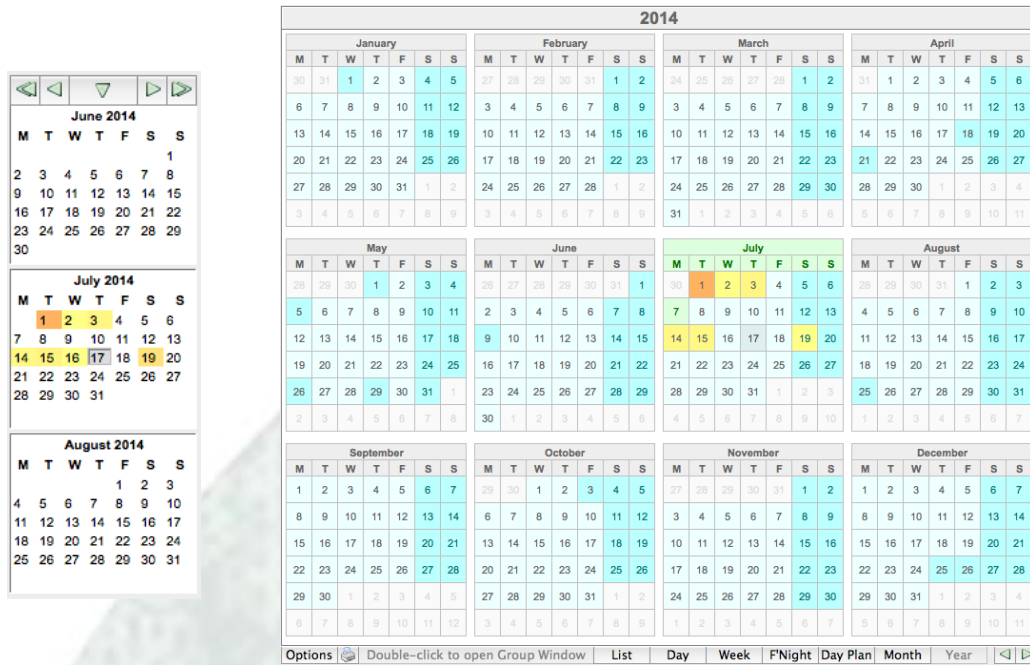
Visual indicators for events that are not in view.



The list day view displays all event s in list form and can display check boxes for simulating to do lists.



Month and Yeardigest views can outline busy days using definable colour ranges.



Designing OCal

Introduction

This chapter gives a brief description of the most important aspects of the enhanced calendar control. For a more detailed description of the external components please read the chapter External Component Reference.

Essentially, the enhanced calendar allows you to display and manipulate a list of events. The list can come from any source but must provide columns with start date/time and optionally an end date/time for each event. You can additionally provide columns for event colours, icons, text, layers, etc.

Important: Your event list must be sorted at all times in ascending order based on the start date/time column of your list.

Basic OCal interface

The ocal interface allows the display of events in the traditional month and day views as well as list and digest modes. The day view is capable of displaying any number of days from just 1 day to 7 days for a week, 31 days for a month, 90 days for a quarter of a year or as many days as is physically possible to fit on the screen.

The Event List

OCal directly interacts with the list of events that you assign to it via the `$eventlist` property. When a user clicks an event box, OCal automatically updates the selection states and current line of your list. However, it is up to you to implement code to manipulate your event list when you receive the appropriate events, i.e. when the user moved or resized an event, pressed the delete key, or dropped objects or data from outside the calendar control. Nothing will happen unless you say so.

Important: When assigning a list to `$eventlist`, the list must be an instance, task or class variable of type list and belong to the window/sub-window that owns the OCal control. You cannot specify an item reference variable because of a data handling limitation in the external interface. However, many people have expressed a need to pass references to list into a window for editing so we have implemented a solution in version 10.5.0. We have provided a new function called `$makedatareference()` available under the 'Functions' tab in the Catalog. This function can be used in the class instance that owns the master list to produce a text based reference to the list that can be used with `$eventlist`.

Event List Columns

As well as specifying the list of events, OCal also needs to know the names of the columns that contain the information that OCal requires to display events. You can tell OCal about your list columns via the set of `$column...` properties. At a minimum, you must provide a column for `$columndate` and `$column time`. If your list contains just one column with combined date and time, you may assign the same column name to both properties. If your events have a duration you must also provide a name for `$columnenddate` and `$columnendtime`.

In addition you can specify columns for the event fill color, text color, icon, etc. OCal will use this to format and display information in the event box. Version 10.5.0 introduces a new way to specify content in events using templates. Please refer to the chapter OCal Templates.

Important: At all times you must ensure that your list is sorted in ascending order by the starting date and time. If the list is not sorted correctly when you issue a redraw, events may not be displayed correctly.

Drag and Drop

Users can interact with events by dragging them to move or resize them. When this happens you will receive appropriate events in the `$event` method. OCal provides a set of properties whose names begin with `$dd...`, that give you control over what users can and cannot do. They are mostly concerned with what can be dropped where.

Layers

Ocal provides the ability to group your events in up to 255 layers. Each event in your list can belong to one or more of these layers. You will need to provide a column in your list that specifies the layers string for each event. The layers basically consists of a series of 'Y' and 'N' characters. For example, the string "YYNYY" tells OCal that this event belongs to layer 1,2,3 and 6. You can tell OCal to show just one, several or all layers at the same time by calculating the property \$layers with a similar string. This makes it very easy to show and hide individual groups of events.

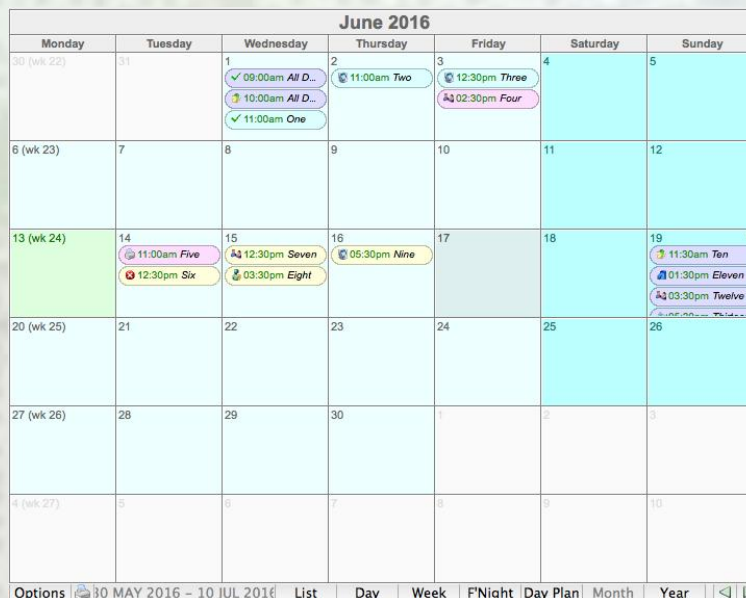
View Modes

From version 2 onwards, besides the traditional Month and Day viewing modes, the calendar can display events in a list view and show monthly or yearly digest information. The desired viewing mode is selected by assigning one of the `kCalViewMode...` constants to the property `$viewmode`.

<code>kCalViewModeMonth</code>	Traditional month view (default)
<code>kCalViewModeDayNormal</code>	Traditional day view with vertical time scales
<code>kCalViewModeDayList</code>	List based day view without vertical time scales New
<code>kCalViewModeMonthDigest</code>	Shows events in month view as shades of color ranging from idle to busy (see <code>\$digest...</code> properties) New
<code>kCalViewModeYearDigest</code>	Shows events in year view as shades of color ranging from idle to busy (see <code>\$digest...</code> properties) New

kCalViewModeMonth

The traditional month view mode displays events one month at a time showing seven days along the horizontal axes and the weeks of the months along the vertical axes. Within this mode, the event display can be customised using the `$templatemonthview` property. There are numerous other appearance properties that can alter the appearance of the month view all of which are documented in the chapter “External Component Reference”.



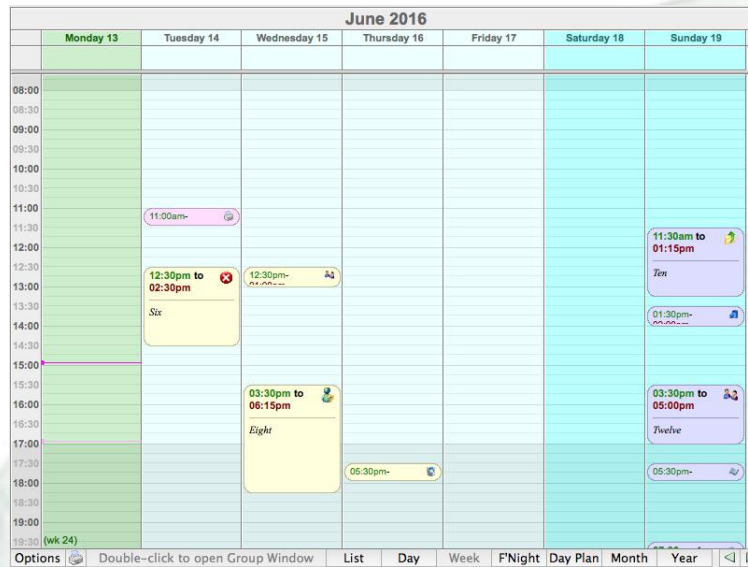
June 2016						
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
30 (wk 22)	31	1 ✓ 09:00am All D... ✓ 10:00am All D... ✓ 11:00am One	2 11:00am Two	3 12:30pm Three 02:30pm Four	4	5
6 (wk 23)	7	8	9	10	11	12
13 (wk 24)	14 11:00am Five 12:30pm Six	15 12:30pm Seven 03:30pm Eight	16 05:30pm Nine	17	18	19 11:30am Ten 01:30pm Eleven 03:30pm Twelve
20 (wk 25)	21	22	23	24	25	26
27 (wk 26)	28	29	30	1	2	3
4 (wk 27)	5	6	7	8	9	10

Options: 30 MAY 2016 – 10 JUL 2016 | List | Day | Week | F'Night | Day Plan | Month | Year

kCalViewModeDayNormal

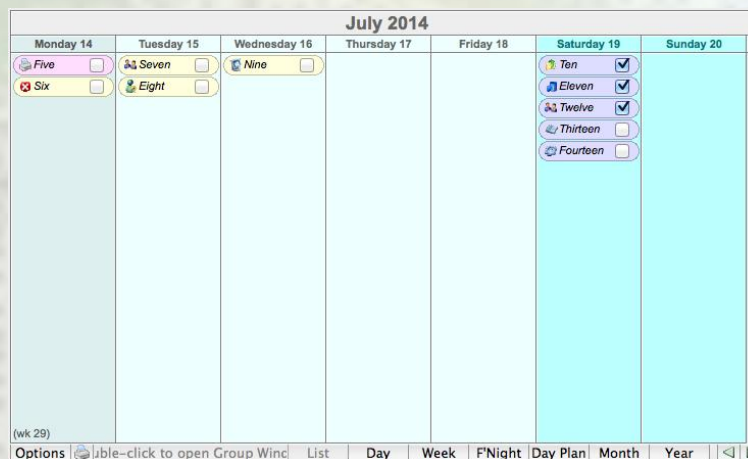
The traditional day view mode can displays events along a vertical time scale with a varied number days along the horizontal axes controlled by `$dayviewdaycount`. Within this mode, the

event display can be customised using the \$templatedayview property and the vertical time scale appearance can be controlled by the \$templatetimecolumn and \$templatetimescale properties. There are numerous other appearance properties that can alter the appearance of the day view all of which are documented in the chapter “External Component Reference”.



kCalViewModeDayList

The new list view mode is similar to the day view display mode but it does not display a vertical time scale. This view simply lists the events vertically only showing a single line of content for each event. The content can be controlled by specifying a template for the \$templatedaylistview property. A new template tag has been added so an active check-box can be displayed within the event allowing users to tick events/to-do items as they are completed.



When a user clicks a check-box, the event evCheckboxClick is sent to the calendars \$event method. The event parameter pColumnNameNumber specifies the list column that the check-box is associated with, if a column was specified in the display template.

Note: Only one check-box is supported.

kCalViewModeMonthDigest & kCalViewModeYearDigest

The new month digest view is like the standard month view except events are displayed in digest mode (the background colour of each day changes according to how busy the day is). The new year view displays an entire year by displaying 12 small month views in digest mode.

June 2014						
M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

July 2014						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

August 2014						
M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

2014

January							February							March							April						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
30	31	1	2	3	4	5	27	28	29	30	31	1	2	24	25	26	27	28	1	2	31	1	2	3	4	5	6
6	7	8	9	10	11	12	3	4	5	6	7	8	9	3	4	5	6	7	8	9	7	8	9	10	11	12	13
13	14	15	16	17	18	19	10	11	12	13	14	15	16	10	11	12	13	14	15	16	14	15	16	17	18	19	20
20	21	22	23	24	25	26	17	18	19	20	21	22	23	17	18	19	20	21	22	23	21	22	23	24	25	26	27
27	28	29	30	31	1	2	24	25	26	27	28	1	2	24	25	26	27	28	29	30	28	29	30	1	2	3	4
3	4	5	6	7	8	9	3	4	5	6	7	8	9	31	1	2	3	4	5	6	5	6	7	8	9	10	11

May							June							July							August						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
28	29	30	1	2	3	4	28	29	30	31	1	2	30	1	2	3	4	5	6	28	29	30	31	1	2	3	
5	6	7	8	9	10	11	5	6	7	8	9	10	7	8	9	10	11	12	13	4	5	6	7	8	9	10	
12	13	14	15	16	17	18	9	10	11	12	13	14	15	14	15	16	17	18	19	20	11	12	13	14	15	16	17
19	20	21	22	23	24	25	16	17	18	19	20	21	22	21	22	23	24	25	26	18	19	20	21	22	23	24	
26	27	28	29	30	31	1	23	24	25	26	27	28	29	28	29	30	31	1	2	25	26	27	28	29	30	31	
2	3	4	5	6	7	8	30	1	2	3	4	5	6	4	5	6	7	8	9	10	1	2	3	4	5	6	7

September							October							November							December						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6	7	29	30	1	2	3	4	5	27	28	29	30	31	1	2	1	2	3	4	5	6	7
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28
29	30	1	2	3	4	5	27	28	29	30	31	1	2	24	25	26	27	28	29	30	29	30	31	1	2	3	4
6	7	8	9	10	11	12	3	4	5	6	7	8	9	1	2	3	4	5	6	7	5	6	7	8	9	10	11

Options

Double-click to open Group Window

List

Day

Week

FNight

Day Plan

Month

Year

The digest colours and how they are calculated is controlled by a range of digest properties, \$digestcoloridle, \$digestcolorbusy, \$digestminutesidle, \$digestminutesbusy, \$digestoptions.

Further Reading

The Brainy Data support website lists a number of technical notes relating to OCal and other software. You should always read these notes before you begin developing our software.

The chapter Event Templates explains in detail how to compose your own templates for the visual interface of OCal.

Event Templates

OCal Version 1.5 introduces templates for greater control over the display of content in an event box, and customize the day view time column and scales. The template properties `$templateallday`, `$templatedayview`, `$templatemonthview` and `$templatedaylistview` control the appearance of event content, making it possible to display multiple icons in any order, text using multiple fonts and styles, and to have different content depending on the size of the event box. The template properties `$templatetimecolumn` and `$templatetimescale` control the day view time column and scales allowing one to place scale lines at any desired position and paint any text in the time column using horizontal or vertical text directions.

Template Syntax

The syntax of a template is in many ways very similar to simple HTML or XML in as far as that you specify tags within the ‘<’ and ‘>’ characters and actual content anywhere else. Some tags may have a start and end tag with content in between, i.e. `<column>` and `</column>`.

You may specify multiple tags within the same set of ‘<’ and ‘>’ characters.

Some tags may have parameters. Parameters are specified by the parameter name, the assignment character followed by the value in double quotes, i.e. `coltype="text"`.

The syntax of a template is very strict and errors in the syntax will result in the template not to paint correctly if at all.

Not all tags are supported in every template, although including them does not cause an error. For example, when specifying a template for custom day view scales, the only valid tags are *scaledark*, *scalelight*, *scalepos*, *scaleline* and *scalefilldark*. Each tag description will specify in which templates the tag is supported.

Decision and Data Tags

This section lists all tags related to data handling and decision making during paint.

<use>

Syntax: `<use>event_content</use>`

Example: `<use wmax=20>...</use>`
`<use hmax=15>...</use>`

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

The use tag is used to decide if the content within the start and end use tags is to be displayed based on the current width or height of the event box.

You may not specify both wmax and hmax for the same use clause. A template may contain multiple use clauses followed by default content without a use clause. Only the first use clause that fits the criteria will be used and no further clauses are processed once one is found.

Parameters	Description
wmax	content is displayed if the event box is n pixels or less in width
hmax	content is displayed if the event box is n pixels or less in height

<column>

Syntax: `<column>column_name</column>`

Example: `<column coltype="text" nowrap>EV_TEXT</nowrap /column>`
`<column coltype="icon48" align="right">EV_ICON_ID</column>`

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

The column tag allows one to specify a column from the event list for displaying content. The coltype parameter tells OCal what type of column this is, text, time or icon. For icons one can also specify icon16, icon32 or icon48, to specify the icon size to be used. In addition Icons can also be right aligned.

Parameters	Description
coltype	Specifies the column type, either "text", "time", "checkbox", "icon", "icon16", "icon32" or "icon48".
align	Specifies the horizontal alignment "left", "center" or "right".

<calculation>

Syntax: <calculation>omnis_calculation</calculation>

Example: <calculation caltype="text" nowrap>cap(iEventsList.EV_TEXT)</nowrap /calculation>
 <calculation caltype="text">jst("T: H:M", iEventsList.EV_DATE_START)</calculation>

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

The calculation tag allows you to specify any valid Omnis calculation that is within the context of the OCal window. The calculation may refer to a column in the event list by specifying the instance list name and column name, omitting the line number. OCal will automatically set the correct line number prior to evaluating the calculation.

Just as with the column tag above, the calculation tag has a parameter to specify the result type of the calculation, so OCal knows how to paint the result.

Important Note: Calculations are slower than the more direct column tags and overuse of calculations may effect performance.

Parameters	Description
caltype	Specifies the calculation result type, either "text", "time", "checkbox", "icon", "icon16", "icon32" or "icon48".

Paragraph and Style Tags and Parameters

This section lists all the tags for handling paragraphs and text styles.

<p>

Syntax: <p>content</p>

Example: <p><column>CUST_NAME</column></p>

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

The paragraph tag is optional. The only time you need to use it is if you want some extra spacing between two sets of content. OCal will automatically add a third of the current font size as a gap between two paragraphs.

Parameters	Description
<i>none</i>	

Syntax:

Example: <column>CUST_NAME</column>
<column>CUST_ADDRESS</column>

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

Insert a new line.

Parameters	Description
<i>none</i>	

,<i>,<u>

Syntax: content,<i>content</i>,<u>content</u>

Example: <i><column>CUST_NAME</column></i>

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

Paints text in between tags in bold, italic or underline.

Parameters	Description
<i>none</i>	

<small>,<big>

Syntax: <small>content</small>,<big>content</big>

Example: <small><small><column>CUST_NAME</column></small></small>

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

Paints text in between tags 1 point smaller or bigger. You may use multiple small or big tags to reduce or increase the font size by several points.

Parameters	Description
<i>none</i>	
<nowrap> Syntax: <nowrap>content</nowrap> Example: <calculation calctype="text" nowrap>cap(iEventsList.EV_TEXT)</nowrap /calculation> Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview All content in between the nowrap tag will prevent wrapping and only specific break or paragraph tags will insert line feeds. If text does not fit on the nowrap line it is displayed with ellipses.	
Parameters	Description
<i>none</i>	
<color> Syntax: <color="#RRGGBB">content</tag> Example: <color="#0000FF"><column>CUST_NAME</column><color="#000000"> Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview The color tag changes the current text color. The color change applies to the remainder of the content. The color value can be specified as a 6 digit hexadecimal value in the format RRGGBB.	
Parameters	Description
<i>self</i>	the color value in hexadecimal
<fontname>, <fontsize> Syntax: <fontname="font_name" fontsize="font_size"> Example: <fontname="Arial" fontsize="10"><column>CUST_NAME</column><fontname="TimesNew Roman" fontsize="12"> Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview The font name tag changes the current font face. The font change applies to the remainder of the content. The font name must be one of the font names that Omnis supports. For a complete list of font names use the Omnis font function FontOps.\$winlistfonts()	
Parameters	Description
<i>self</i>	the font name

<align>

Syntax: <align="alignment">

Example: <column align="left">CUST__NAME</column>

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview,
\$templatetimecolumn

The align tag changes the current text or icon alignment. When used with \$templateallday, \$templatedayview, \$templatemonthview, the only supported option is “right” when used with icons. Changing the text alignment in event boxes is not supported.

When used with \$templatetimecolumn, all options are supported. The up and down alignment will draw the text vertically, centering the text at the current scale position.

Parameters	Description
<i>self</i>	the alignment, one of "left", "center", "right", "up" or "down".

<hr>

Syntax: <hr color="#RRGGBB">

Example: Text Above<hr color="#FF0000">TextBelow

Templates: \$templateallday, \$templatedayview, \$templatemonthview, \$templatedaylistview

The horizontal ruler tag will draw a horizontal single pixel line, allowing the clear separation of content. By default the line will take on the color of the event box border. This color can be overridden by specifying a color within the ruler tag

Parameters	Description
color	the line color

Scale Tags

The following are specific tags supported when painting custom day view time column and scales.

<scalelight><scaledark>

Syntax: <scaledark>dark_scale_lines<scalelight>light_scale_lines

Example: <scaledark><scalepos="480" scaleline><scalelight><scalepos="540" scaleline>

Templates: \$templatetimecolumn, \$templatetimescale

Selects the light or dark font or pen for painting the lighter or darker scale text or lines.

Parameters	Description
<i>none</i>	

<scalepos>

Syntax: <scalepos="minutes">

Example: <scaledark><scalepos="480" scaleline><scalelight><scalepos="540" scaleline>

Templates: \$templatetimecolumn, \$templatetimescale

Sets the current scale position by specifying the time of day in minutes. For example 8am would be specified as 480 (8 x 60), and 5pm would be 1020 (17 x 60)

Parameters	Description
<i>self</i>	the vertical position in minutes

<scaleline>

Syntax: <scaleline>

Example: <scaledark><scalepos="480" scaleline><scalelight><scalepos="540" scaleline>

Templates: \$templatetimecolumn, \$templatetimescale

Instructs OCal to paint a scale line at the current position using the selected pen.

Parameters	Description
<i>none</i>	

<scalefilldark>

Syntax: <scalefilldark>

Example: <scalepos="720" scalepos="900" scalefilldark> // fill the area between 12pm and 3pm

Templates: \$templatetimecolumn, \$templatetimescale

Instructs OCal to fill the background of the time column or day view area with the non-working hour color. The fill is positioned between the last two scalepos coordinates.

Parameters	Description
<i>none</i>	

Sample Event Templates

The following are the templates used in the OCal example library. Explanations of each line is given in read between the lines of the template text but do not form a valid part of the template text it self.

Month View Template(\$templatemonthview)

The month view template specifies four “use” clauses for when the width of the event is less or equal to 60, 80, 100, or 130 pixels respectively. For the smaller use clauses less data is displayed using a smaller font size with the use of the <small> tag.

;; for month views, event content is not wrapped so we use the <nowrap> tag for all content
<nowrap>

;; when the event width is 60 pixels or less, we display content three point sizes smaller, and we display an icon,
;; start time in green and the event text in italic.

```
<use wmax=60>
<small><small><small>
<column coltype="icon">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column>
<column coltype="text" color="#000000"><i>EV_TEXT</i></column>
</use>
```

;; when the event width is 80 pixels or less, we display content two point sizes smaller, and we display an icon,
;; start time in green and the event text in italic.

```
<use wmax=80>
<small><small>
<column coltype="icon">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column>
<column coltype="text" color="#000000"><i>EV_TEXT</i></column>
</use>
```

;; when the event width is 100 pixels or less, we display content one point size smaller, and we display an icon,
;; start time in green and the event text in italic.

```
<use wmax=100>
<small>
<column coltype="icon">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column>
<column coltype="text" color="#000000"><i>EV_TEXT</i></column>
</use>
```

;; when the event width is 130 pixels or less, we display content at the default size, and we display an icon,
;; start time in green and the event text in italic.

```
<use wmax=130>
<column coltype="icon">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column>
<column coltype="text" color="#000000"><i>EV_TEXT</i></column>
</use>
```

;; when the event width is wider than 130 pixels, we display content at the default size, and we display an icon,
;; start time in green, end time in red and the event text in italic.

```
<column coltype="icon">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column><color="#000000">-
<column coltype="time" color="#880000">EV_DATE_END</column>:
<column coltype="text" color="#000000"><small><i>EV_TEXT</i></small></column>
```


All-Day View Template(\$templateallday)

The all-day view template specifies just one “use” clause for when the width of the event is less or equal to 60 pixels.

;; for all-day views, event content is not wrapped so we use the <nowrap> tag for all content
<nowrap>

;; when the event width is 60 pixels or less, we display content two point sizes smaller, and we display an icon,
;; and the event text in italic.

```
<use wmax=60>
<small><small>
<column coltype="icon">EV_ICON_ID</column>
<column coltype="text"><i>EV_TEXT</i></column>
</use>
```

;; when the event width is wider than 60 pixels, we display content at the default size, and we display an icon
;; and the event text in italic.

```
<column coltype="icon">EV_ICON_ID</column>
<column coltype="text"><i>EV_TEXT</i></column>
```

Day View Template(\$templatedayview)

The day view template specifies the “use” clauses for when the hight is 25 pixels or less and the width is 90 pixels or less, respectively.

;; when the event height is 25 pixels or less, we display content one point size smaller, and we display a right-aligned
;; icon, the start time in green, the end time in red and the event text in Italic and Times New Roman.

```
<use hmax=25><small>
<column coltype="icon" align="right">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column color="#000000">-
<column coltype="time" color="#880000">EV_DATE_END</column color="#000000"><br>
<column coltype="text" color="#000000" fontname="Times New Roman"><i>EV_TEXT</i></column>
</small></use>
```

;; when the event width is 90 pixels or less, we display content one point size smaller, and we display a right-aligned
;; icon, the start time in green, the end time in red and the event text in Italic and Times New Roman.

```
<use wmax=90><small>
<column coltype="icon" align="right">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column color="#000000">-
<column coltype="time" color="#880000">EV_DATE_END</column color="#000000"><br>
<column coltype="text" color="#000000" fontname="Times New Roman"><i>EV_TEXT</i></column>
</small></use>
```

;; for all other event sizes we use the default font size and we display a 16 pixel right-aligned icon, start time in green,
;; end time in red, followed by a new paragraph with the event text in Italic and Times New Roman

```
<p><b>
<column coltype="icon16" align="right">EV_ICON_ID</column>
<column coltype="time" color="#008800">EV_DATE_START</column color="#000000">
to <column coltype="time" color="#880000">EV_DATE_END</column color="#000000">
</b></p>
<p>
<column coltype="text" color="#000000" fontname="Times New Roman"><i>EV_TEXT</i></column>
</p>
```

List Day View Template(\$templatedaylistview)

Our example list template displays a non-wrapping line containing a checkbox, icon and text.

;; specify no wrapping, causes text to display ellipses if it does not fit

<nowrap>

;; use smaller text if we have 80 pixels or less in width

<use wmax=80>

<small><small>

<column coltype="checkbox" align="right">EV_DONE</column>

<column coltype="icon" align="left">EV_ICON_ID</column>

<column coltype="text"><i>EV_TEXT</i></column>

</use>

;; standard view when we have more than 80 pixels

<column coltype="checkbox" align="right">EV_DONE</column>

<column coltype="icon" align="left">EV_ICON_ID</column>

<column coltype="text"><i>EV_TEXT</i></column>

July 2014				
	Tuesday 15	Wednesday 16	Thursday 17	Friday 18
	 Seven <input type="checkbox"/>	Group Sev... <input type="checkbox"/>		
	 Eight <input type="checkbox"/>	 Nine <input checked="" type="checkbox"/>		
		Group Sev... <input type="checkbox"/>		

Ellipses

Sample Time Column and Scale Templates

The following sample templates demonstrate what can be achieved when applying templates to both the time column and day view scale. The Image at the end shows the visual result of the two templates.

Time Column Template(\$templatetimecolumn)

This sample displays a mixture of times painted horizontally and text denoting different parts of the day, painted vertically.

```
;; we pretend to be a restaurant and break up the day into portions relevant to the establishment.
;;
;; first we fill an area between 3pm and 6pm with the non-working hour fill, as this is the time when we are closed
<scalepos="900" scalepos="1080" scalefilldark>

;; next we specify the times that separate the parts of the day. We right justify the times within the time column and
;; paint them using the default light font
<align="right" scalelight>
<scalepos="480">08:00
<scalepos="660">11:00
<scalepos="900">15:00
<scalepos="1080">18:00
<scalepos="1380">23:00

;; next we paint portions of the day when the restaurant is closed in a subtle red color
;; for this we now use a larger Arial font and we change the painting direction to up
<fontname="Arial" fontsize="18" align="up">
<color="#886666">
<scalepos="990">Closed

;; the portions of the day when food is served are painted in a subtle green
<color="#668866">
<scalepos="570">Breakfast
<scalepos="780">Lunch
<scalepos="1230">Dinner
```

Scale Template(\$templatetimescale)

This template is used to specify the position of the horizontal scale lines in the day view. Typically it is synchronized in some form with the content of the time column.

```
;; first we fill an area between 3pm and 6pm with the non-working hour fill, as this is the time when we are closed
<scalepos="900" scalepos="1080" scalefilldark>

;; now we paint the darker scale lines at the time positions that separate the different parts of the day
;; (Breakfast, Lunch, etc)
<scaledark>
<scalepos="480" scaleline>
<scalepos="660" scaleline>
<scalepos="900" scaleline>
<scalepos="1080" scaleline>
<scalepos="1380" scaleline>

;; in between the darker scales we paint lighter scales on every hour
<scalelight>
<scalepos="540" scaleline>
<scalepos="600" scaleline>
```



```
<scalepos="720" scaleline>
<scalepos="780" scaleline>
<scalepos="840" scaleline>
<scalepos="960" scaleline>
<scalepos="1020" scaleline>
<scalepos="1140" scaleline>
<scalepos="1200" scaleline>
<scalepos="1260" scaleline>
<scalepos="1320" scaleline>
```

The final result is shown in the image below.



Timelines Template(\$templatetimelines)

The \$templatetimelines property specifies the template for time lines that are drawn across all days or specified days within the day view display mode. The following template tags are supported.

scalepos: specifies the position in minutes within the day view

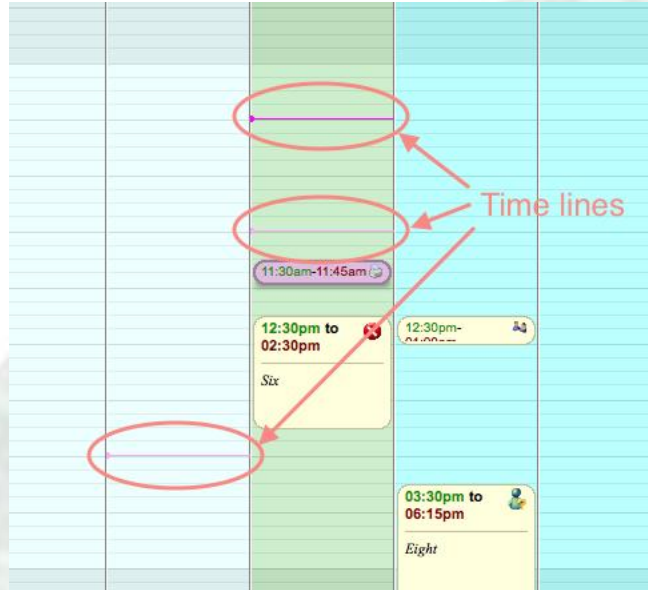
scaleposoffset: specifies the offset in minutes from scalepos. If this tag is not specified, scaleline instruction will paint a line across all displayed days. If scaleposoffset specifies zero, a scaleline instruction will draw a line within today's box only. If scaleposoffset specifies a negative or positive offset, the next scaleline instruction draws a line offset by the specified minutes within the box of the appropriate day.

scaleline: draws a time line at positions calculated from scalepos and scaleposoffset.

color: changes the color for the next scaleline instruction

Example: The following template would produce the time lines displayed in the image.

```
<color="#FF00FF" scalepos="540" scaleposoffset="0" scaleline>
<color="#FFAAFF" scaleposoffset="-1080" scaleline>
<scaleposoffset="+120" scaleline>
```



The bolder time line is displayed at 9am in the todays box. The second time line is displayed 18 hours earlier and the third time line is displayed 2 hours later.

Implementation notes: In the examples we use a timer object (oCalCurrentTime) that runs once every minute which updates \$templatetimelines with the current time.

External Component Reference

Introduction

This chapter lists all the enhanced calendar properties, methods and events. The OCal component library supplies a window control for use on your windows, a remote form control for use in remote forms for the web-client, and a report object for printing calendars.

Contents

Properties

Events

Methods

Static Methods

Properties	
Property	Description
\$allowchange	If true, the user can change the current date
\$ampmstring	String that specifies the am/pm characters separated by a '/'. Example: 'am/pm'
\$column...	\$columnname, \$columnendname, \$columnstart, \$columnend, \$columnallday, \$columnicon, \$columnlabel, \$columnlabelcolor, \$columnbackground, \$columntooltip, \$columnlayers - all these properties specify the names of the columns in your list that provide the information. This gives you great flexibility in how you organize your list/smart list.
\$currdays	The calendar's current date
\$currdayscolor	The fill color of the current day
\$currdaysicon	The icon used to represent the current day (month view only)
\$currdaysmode	The border style for the current date
\$currdayscolor	The color of the current date
\$dayscolor	The color of other then the current, weekend, and other months days
\$daysfont	The font for the days in this month
\$daysfontsize	The font size for the days in this month
\$daysmode	The border style for the days in this month
\$daytooltips	If true, a separate tool-tip is required for each day. See also \$tooltipday.
\$dayview (v2.0)	Obsolete in version 2. Please see \$viewmode.
\$dayviewalldayheight (v2.0)	User defined height for the all-day panel. If set to zero, the panel will size as required, if set to a positive value the height of the panel is fixed to that number of pixels and a scroll bar will be provided if required.
\$dayviewalldayheightmax (v2.0)	Maximum height off all day pane in percent (10% to 90%, default 40%). A scroll bar will be shown if not all all-day events can be shown in the provided space. See also \$dayviewalldayheight
\$dayviewcancreate	If true, the user can create events in day view mode by clicking and dragging across the calendar background. Feedback will be given during dragging. See evCreateEvent.

\$dayviewdate	<p>Date of first day displayed in day view mode. Prior to switching on \$dayview, calculating this property to #NULL will use \$currday and \$dayviewdaycount and \$firstday to calculate a sensible \$dayviewdate. How it is calculated depends on the setting of \$dayviewdaycount;</p> <ul style="list-style-type: none"> 1 \$dayviewdate will be set to \$currday. 7 \$dayviewdate will be set to \$firstday in the week specified by \$currday. 14 \$dayviewdate will be set to \$firstday in a two week period specified by \$currday. 21 \$dayviewdate will be set to \$firstday in a three week period 28 \$dayviewdate will be set to \$firstday in a four week period specified by \$currday 31 \$dayviewdate will be set to the first of the month specified by \$currday, and only the days of that month are shown. <p>For all other settings, \$dayviewdate is calculated so that \$currday is centered. See also \$firstday.</p>
\$dayviewdaycount	The number of days displayed in day mode.
\$dayviewdayend	Number of minutes into the day when working hours end, 1020 minutes = 1700 hours. Non-working hours are displayed darker.
\$dayviewdaystart	Number of minutes into the day when working hours begin, 540 minutes = 0900 hours. Non-working hours are displayed darker.
\$dayviewdeadcolumncolor (v1.3)	Fill color for the left and right dead columns in all day panel.
\$dayviewdividercolor (v1.3)	Fill color for the all-day / day-view divider.
\$dayviewheadingusedaycolors (v2.0)	If true, day headings are rendered using the colours of the day as used in the client area.
\$dayviewhoursvisible	Number of hours displayed in day mode.
\$dayviewmineventwidth	The minimum event width in percent of the day column width, when two or more events overlap in time.
\$dayviewminutescale (v1.2)	If non zero, additional horizontal minute scale lines are drawn at the specified interval. The valid range is 1 to 30. A value of 0 is the default and draws a scale at 30 minutes to every hour. See also \$dayviewtimescale.

\$dayviewnowrap (v1.5)	If kTrue, text in an event box is not wrapped. Only a CR character will create a new line.
\$dayviewvposminutes (v1.2)	The vertical scroll bar position in minutes from mid-night onwards. This property can be used to position the vertical offset in the report object to simulate the current view in the window calendar component.
\$dayviewsnapminutes	The number of minutes events are snapped to when dragging or sizing events. If it is set to 15, events are sized or moved in increments/decrements of 15 minutes.
\$dayviewtimecolumncolor (v1.3)	Fill color for the time column in day view.
\$dayviewtimecolumntextcolor (v1.3)	Text color for the time column in day view.
\$dayviewtimecolumnwidth (v1.5)	Width of the time column in day view. Set it to -1 to hide the time column, set it to zero to let the control size the column, set to > 0 to fix the width of the time column. When reading this property at runtime, it returns the actual width of the time column in pixels.
\$dayviewtimescale (v1.2)	If non zero, additional times are displayed in the time column at the specified minute interval. The valid range is 1 to 30. A value of zero is the default and only draws times on the hour. See also \$dayviewminutescale.
\$ddallowotherfields (v1.4)	If true, the calendar control will allow drops during drag & drop from other fields of any type (default is false). You can still implement evCanDrop in the usual way.
\$ddhilitellday (v1.4)	If true, all-day cells in the day view mode are enabled and highlighted during drag & drop. (default is true)
\$ddhiliteworking (v1.4)	If true, working hours time slots in the day view mode are enabled and highlighted during drag & drop. (default is true)
\$ddhilitenonworking (v1.4)	If true, non-working hours time slots in the day view mode are enabled and highlighted during drag & drop. (default is true)
\$digestcoloridle (v2.0)	Color representing idle range as described by \$digestminutesidle.
\$digestcolorbusy (v2.0)	Color representing idle range as described by \$digestminutesbusy.
\$digestminutesidle (v2.0)	Minimum number of minutes required (or events when \$digestoptions specifies kCalDigestOptCountEvents) before the idle colour is shown.

\$digestminutesbusy (v2.0)	Minimum number of minutes required (or events when \$digestoptions specifies kCalDigestOptCountEvents) before the busy colour is shown.
\$digestoptions (v2.0)	specifies options for how to collect digest data. One of the following kCalDigestOptNone: No digest options specified kCalDigestOptCountEvents: If specified digest view will count events rather than minutes occupied by events kCalDigestOptIgnoreWorkHours: If specified, events falling partially or completely outside working hours will be included in the digest view in their entirety
\$downarrowwid	This property allows you to specify icons for the up and down arrows to scroll through events in a single day (Month view only) if there is not enough room to display them all.
\$dropdate	The date on which the drop occurred. Set when you receive an evDrop message.
\$eventslist	This property takes a list name with the following possible columns; combined Date/Time or separate Date and Time columns, combined End Date/Time or separate End Date and End Time columns, Icon, Display Text, Display TextColor, Display Back Color, Tool-tip text, Event layers (groups this event belongs to).
\$firstday	The first day the calendar will show. See also \$dayviewdate.
\$headingbold	If true, the column headers are drawn in bold
\$headingcolor	The fill color for the column headers
\$headingfont	The font for the column headers
\$headingfontsize	The font size for the column headers
\$headingmode	The border style for the column headers
\$headingstyle	Alternative text style for the column headers
\$headingtextcolor	The text color of the calendar column headers

\$holidaylist (v1.5.0)	<p>Specifies a list of public holidays. The list must contain the following columns in the given order:</p> <p>Date: The day, month and year of the special day</p> <p>Fill Color: The background fill color for that day. If kColorDefault, the fill-color is not altered.</p> <p>Text Color: The text color for that day</p> <p>Text: The text to be displayed in the bottom left corner of that day, i.e. the name of the public holiday or any other suitable description.</p> <p>Tooltip: (optional) If specified, it provides the text for a tooltip.</p> <p>One can specify multiple lines of text by separating text with a chr(13) character.</p>
\$isoweektext (v1.6.0)	<p>If specified, OCal will display the ISO 8601 week number in both the day and month views. The week number is always displayed in the cells for each Monday. When specifying the text, including a ‘\$’ character indicates the point where OCal will insert the week number. If the ‘\$’ character is not present, the week number is appended.</p>
\$layers	<p>This property can be set to only show events belonging to the specified groups (layers). It works like a bit array. An event can belong to all your groups or just some or one. The \$layers property can be set to show all groups, just some or just one. It is specified as a string of ‘Y’ for show events and ‘N’ for hide events. If your calendar supports 7 groups your string may look like “YYNNYY” which means show groups 1,2,6 and 7.</p>
\$monthtextcolor	<p>The color of the days in this month</p>
\$mouseallday (v1.4)	<p>Returns true if the mouse is over the all-day pane in the day view.</p>
\$mousedate (v1.4)	<p>Returns the date and time under the mouse. The time is adjusted according to \$dayviewsnapminutes.</p>
\$mouseevent (v1.4)	<p>Returns the line number of the event in the list currently under the mouse. Zero is returned if the mouse is not over an event.</p>
\$noeventbackground	<p>If true and \$roundedboxes is false, events will ignore event background colors.</p>

\$noeventborder (v1.3)	If true, rounded boxes have no extra border around the content making them slightly smaller in appearance.
\$otherdaycolor	The fill color of the days in previous and next month
\$otherdaymode	The border style for the previous and next months days
\$othertextcolor	The text color of the days in previous and next month
\$repeatcontent	If true, content is repeated when the event wraps in the day view or month views.
\$roundedboxes	If true, events are drawn inside a box with rounded corners. A bit like events in iCal on Mac OSX.
\$shortname	If true, the days are drawn using a short name
\$showheading	If true, days of the week are shown
\$templateallday (v1.5)	Template for rendering content for events in the all-day pane. Please refer to the section Event Templates for a full description.
\$templatedayview (v1.5)	Template for rendering content for events in the day view. Please refer to the chapter Event Templates for a full description.
\$templatemonthview (v1.5)	Template for rendering content for events in the month view. Please refer to the chapter Event Templates for a full description.
\$templatetimecolumn (v1.5)	Template for rendering content in the time column of the day view. Please refer to the chapter Event Templates for a full description.
\$templatetimelines (v2.0)	Specifies a template for rendering time lines in the day view. Please refer to the chapter Event Templates for a full description.
\$templatetimescale (v1.5)	Template for rendering scale lines in the day view. Please refer to the chapter Event Templates for a full description.
\$titlebold	If true, the calendar title is drawn in bold
\$titlecolor	The fill color for the calendar title
\$titledateformat	The date format used when evaluating the text for the calendar title, i.e. "n y" Shows "January 2006"
\$titlefont	The font used for the calendar title
\$titlefontsize	The font size for the calendar title
\$titlemode	The border style for the calendar title.
\$titlestyle	Alternative text style for the calendar title, no need to specify font, font size, etc

\$titletext	The calendar title. The '\$' character specifies the insertion point for the calendar date with the formatting as specified by \$titledateformat
\$titletextcolor	The text color of the calendar title
\$todaybold	If true, today's date is drawn in bold
\$todaycolor	The fill color of today's date
\$todaytextcolor	The text color of today's date
\$tooltipday	The date to use for the current tool-tip; only significant when \$daytooltips is kTrue, and a tool-tip is being generated.
\$uparrowwid	see \$downarrowwid.
\$viewmode (v2.0)	<p>Specifies the current view mode. Please also see the chapter “Designing OCal”</p> <p>kCalViewModeMonth Traditional month view (default)</p> <p>kCalViewModeDayNormal Traditional day view with vertical time scales</p> <p>kCalViewModeDayList List based day view without vertical time scales New</p> <p>kCalViewModeMonthDigest Shows events in month view as shades of color ranging from idle to busy (see \$digest... properties) New</p> <p>kCalViewModeYearDigest Shows events in year view as shades of color ranging from idle to busy (see \$digest... properties) New</p>
\$vertscroll (v1.5.0)	Hides or shows the vertical scroll bar in day view.
\$vscroll (v1.5.0)	The current vertical scroll position in day view. See also evScroll.
\$weekendcolor	The background fill color for Saturdays and Sundays.
\$weekenddays (v1.2)	A 7 character mask of 1s and 0s that specify the days that make up the weekend. The first character in the mask is the Sunday, and the seventh character is the Saturday. The default mask for western countries is “1000001”.
\$weekendtextcolor	The background text color for Saturdays and Sundays.

Events

evCanDrop

Standard Omnis event. Send during dragging of events.

Standard parameters

evClick

This event is generated when the user has clicked on an event. The list lines would have been appropriately selected or de-selected. Shift clicking will select a range of events, ctrl-clicking will add the event to the selection.

This event has no parameters

evCreateEvent

This event is generated when \$dayviewcancreate is enabled, and the user clicks and drags the calendar background in day view mode to create an event.

Event Parameter	Description
pStartDate	The start date & time for the new event snapped to \$dayviewsnapminutes.
pEndDate	The end date & time for the new event snapped to \$dayviewsnapminutes.
pAllDay (v1.3)	If true, the event was created in the all-day pane of the day view.

evDateChange

Sent to the control when the current date is changed.

Event Parameter	Description
pCurrentDate	The new current date.

evDateDClick

Sent to the control when the user double clicks on a date.

Event Parameter	Description
pCurrentDate	The date & time clicked on snapped to \$dayviewsnapminutes.

evDateRClick

This event is generated when the user has right clicked on an event or the calendar background.

Event Parameter	Description
pCurrentDate	The date & time clicked on snapped to \$dayviewsnapminutes.

evDeleteEvent

Version: 1.3

This event is generated when the user presses the delete or backspace keys. The default action should be to delete all selected events in the event list.

This event has no parameters

evDoubleClick

This event is generated when the user has double-clicked on an event. The list lines would have been appropriately selected or de-selected.

This event has no parameters

evDragFinished

Send when drag & drop has finished

This event has no parameters

evDrop

Standard Omnis event. Send when an event was dropped.

Standard parameters plus \$dropdate is set to the date and time where the user let go. The time part of \$dropdate will be snapped to \$dayviewsnapminutes.

evMonthReset

Sent to the control when the month is changed.

Event Parameter	Description
pCurrentDate	The new current date.

evResizeEventStart

This event is generated when the user has changed the start date/time by dragging the top edge of an event.

This event has no parameters, but \$dropdate is set to the date and time where the user let go. The time part of \$dropdate will be snapped to \$dayviewsnapminutes.

evResizeEventEnd

This event is generated when the user has changed the end date/time by dragging the bottom edge of an event.

This event has no parameters, but \$dropdate is set to the date and time where the user let go. The time part of \$dropdate will be snapped to \$dayviewsnapminutes.

evScroll

Version:v1.5

This event is generated when the control is scrolled by the user.

This event has no parameters, but \$vscroll is set to the current vertical scroll position.

Methods

\$clearicons()

Syntax: *OCalObjectRef*.\$clearicons()

Version: 1.0

Sets the icon for the specified day.

Parameter	Description
returns	n/a

\$getdayicon()

Syntax: *OCalObjectRef*.\$getdayicon(iDay)

Version: 1.0

Gets the icon ID for the specified day.

Parameter	Description
iDay	The date of the current month.
returns	The icon ID

\$setdayicon()

Syntax: *OCalObjectRef*.\$setdayicon(iDay,iDayIcon)

Version: 1.0

Sets the icon for the specified day.

Parameter	Description
iDay	The date of the current month.
iDayIcon	The icon ID.
returns	n/a

Static Methods

\$disablescreenupdates()

Syntax: Calendar Library.\$disablescreenupdates()

Version: 1.5

This method can be used to temporarily disable all screen updates to prevent flashing and increase performance when assigning numerous properties to OCal. This call must be balanced with a call to \$enablescreenupdates().

Parameter	Description
-----------	-------------

returns 1 if successful

\$enablescreenupdates()

Syntax: Calendar Library.\$enablescreenupdates()

Version: 1.5

Call this method if you have previously disabled screen updates and you have finished modifying the control. OCal will update all affected controls.

Parameter	Description
-----------	-------------

returns 1 if successful

\$makedatareference()

Syntax: Calendar Library.\$makedatareference(cVariableName,&cOutReference)

Version: 1.5

Creates a special reference to an Omnis list containing events for display in OCal. The reference that is returned can be used with \$eventslist in any OCal window control even if the list belongs to another class. WARNING. If the data list is destroyed and a OCal control still has a reference to that list, Omnis may crash.

Parameter	Description
cVaraiableName	Name of the Omnis list.
cOutReference	A text based reference is returned in this parameter.

returns 1 if successful