FNC

TMS FNC Calendar
DEVELOPERS GUIDE

May 2018

Web: http://www.tmssoftware.com
Email: info@tmssoftware.com

Index

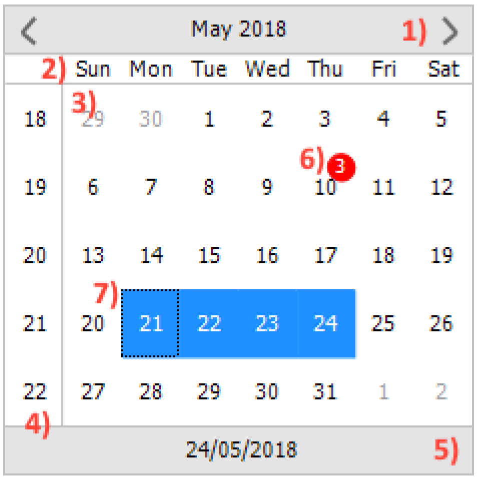
Availability 3
Hierarchy 4
Date selection 5
Navigation 6
Badges 8
TMSFNCDatePicker 9
Important methods, properties and events 10

Availability

Supported frameworks and platforms
- VCL Win32/Win64
- FMX Win32/Win64, macOS, iOS and Android
- LCL Win32/Win64, Linux and macOS

Supported IDE's
- Delphi XE7 and C++ Builder XE7 or newer releases
- Lazarus 1.4.4 with FPC 2.6.4 or newer official releases.

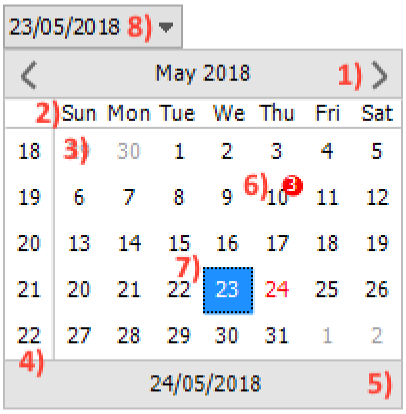
Hierarchy



May 2018

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	2

24/05/2018



23/05/2018

May 2018

	Sun	Mon	Tue	We	Thu	Fri	Sat
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	2

24/05/2018

1) Header: The header is an area that displays the currently shown month and year in text. The navigation arrows can also be found in this area. The arrow type can be changed from arrow to triangle, bitmap or custom. The navigation can also be changed to navigate by the focused date, month, year or custom.

2) Weekday names: The weekday name area shows the short names of the weekdays. The order of the days is based on what's set for the first day of the week.

3) Dates: In the date area the currently shown month's days are painted. It consists of many elements such as the optionally shown dates before and after the current month, selected dates, focused date, today's date and the badges. The focused date can be navigated with the keyboard if the TMSFNCCalendar or TMSFNCDatePicker is in focus and the keyboard support is enabled.

4) Weeknumbers: The weeknumber area is optional and it's not visible by default. The weeknumbers are shown according to the ISO-8601 standard.

5) Footer: The footer is an area to show today's date, the selected date, or no date at all. An extra caption can also be added before the date that's being displayed. If the footer is set to display today's date, then it can be used to navigate back to today's month view.

6) Selected dates: By default, only one date can be selected at a time, but multiselection can be enabled via the Interaction property. Mutliselection is supported by both mouse and keyboard.

7) Events/badges: Events/badges can be used to indicate past or upcoming events. The badges can be customized by setting their own color, font color, value, hint and the date where they should be displayed.

8) Date picker dropdown: The dropdown is displaying the currently selected date. The field can be set to editable, in which case a date can be entered and that will become the selected date. Upon entering an incorrect date format, the selected date will be today's date.

Date selection

During designtime or programmatically you can take some steps to prepare the TMSFNCCalendar for the desired use regarding to date selection. In the Interaction property you'll find options such as KeyboardSupport for keyboard usage, MultiSelect for the possibility to select multiple dates, and ReadOnlyMode to disable any date selection that can be made by the user.

Selected dates

It's also possible to access selected dates and to select dates programmatically. The only/last selected date can be accessed with a separate function but access to every selected date is also provided. For the selected dates, the following code can be used:

```
var
  d: TDate;
begin
  d := TMSFNCCalendar1.SelectedDate;

  if TMSFNCCalendar1.SelectedDates.Count > 0 then
    d := TSMFNCCalendar1.SelectedDates.Items[0];
end;
```

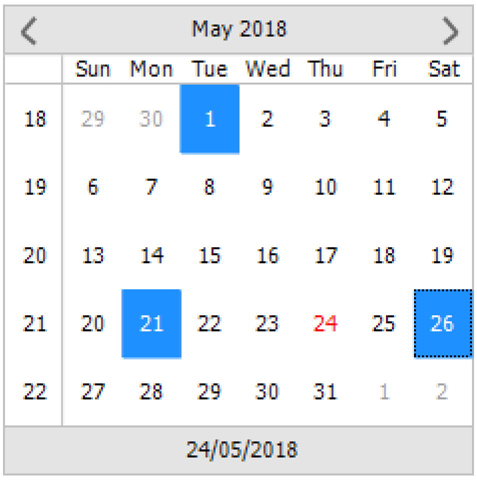
Selecting dates

Selecting a single date will unselect everything else, and it will set the focused date to the selected date. It can be done using the following code:

```
TMSFNCCalendar1.SelectADate(Now + 2);
```

There are two methods to select multiple dates. With the use of the first one, you can select dates one-by-one, which makes disjunct date selection possible. In this case, the selected dates that were selected before, will not get unselected:

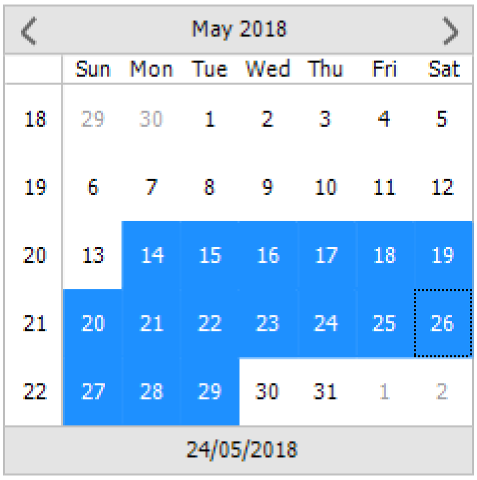
```
TMSFNCCalendar1.SelectDate(Now - 3);
TMSFNCCalendar1.SelectDate(EncodeDate(2018, 5, 1));
```



	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	2

With the second method, an interval selection can be made between two dates. Any dates that were selected prior to calling this method, will get unselected:

```
TMSFNCCalendar1.Interaction.MultiSelect := True;
TMSFNCCalendar1.SelectMultiDates(Now - 10, Now + 5);
```



May 2018

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	2

24/05/2018

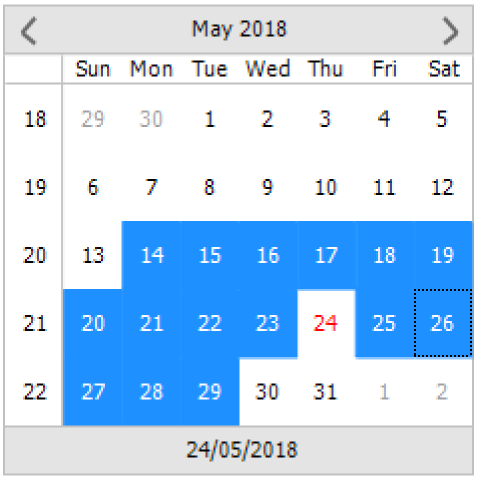
Unselecting dates

Using the TMSFNCCalendar.SelectedDate property, you can unselect dates that are selected or unselect all of the selected dates. To unselect all, use:

```
TMSFNCCalendar1.SelectedDates.UnselectAll;
```

To unselect a specific date only, the following code can be used:

```
TMSFNCCalendar1.SelectedDates.UnselectDate(Now);
```



May 2018

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	2

24/05/2018

Navigation

The default behaviour of the navigation is navigating by month. Clicking the left arrow will navigate to the previous month while clicking the right will navigate to the next month. The navigation mode can be found in Interaction.NavigationMode. You can change from nmMonth to nmFocusedDate, in which case the left arrow will move the focused date to the left and the right arrow will move it to the right by one. Similarly, changing it to nmYear will cause the navigation to navigate back and forth by one year.

The fourth option is nmCustom. Using nmCustom, you can define your own navigation method. This can be achieved by using the OnCustomNavigation event of the TMSFNCCalendar.

In this example, the custom navigation is set to navigate by 10 years back and forth:

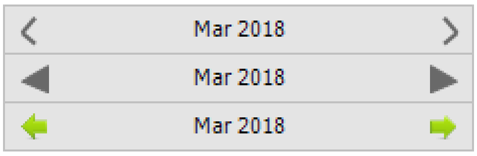
```
procedure TForm1.TMSFNCCalendar1CustomNavigation(Sender: TObject; ADate,
  AFocusedDate: TDate; ADirection: Boolean);
begin
  if ADirection then
    TMSFNCCalendar1.Year := TMSFNCCalendar1.Year - 10
  else
    TMSFNCCalendar1.Year := TMSFNCCalendar1.Year + 10;
end;
```

If the Interaction.SwipeOnHeader property is set to True, then the navigation is also working by swiping on the header.

Arrows

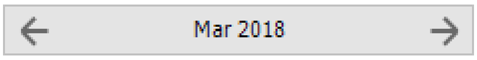
The arrows' look can be changed using the predefined two arrow types, the bitmaps and the custom option. The arrow types can be accessed through the Header.Arrow.ArrowLeftType and Header.Arrow.ArrowRightType properties. The following types are available: atArrow, atTriangle, atBitmap and atCustom.

The arrow types can be changed at designtime and programmatically. The atBitmap type will appear blank unless a bitmap is assigned to the correct properties. You can either load a bitmap to each property or use a bitmap container. To use a bitmap container, a TMSFNCBitmapContainer must be assigned to the BitmapContainer property, where the required bitmaps can be stored. Then you can add your bitmaps to the Header.Arrow.ArrowLeftBitmap and Header.Arrow.ArrowRightBitmap properties by selecting them from the bitmap container.



With the atCustom type, a custom arrow can be drawn. To do that, you will need to use the OnCustomArrow event of the TMSFNCCalendar.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  TMSFNCCalendar1.Header.Arrow.ArrowLeftType := atCustom;
  TMSFNCCalendar1.Header.Arrow.ArrowRightType := atCustom;
end;

procedure TForm1.TMSFNCCalendar1CustomArrowDraw(Sender: TObject;
  AGraphics: TTMSFNCGraphics; ARect: TRectF; ADirection: Boolean);
var
  pr, pl: TPointF;
  h, r, l: Single;
begin
  h := (ARect.Bottom - ARect.Top);
  pl := PointF(ARect.Left, ARect.Top + h / 2);
  pr := PointF(ARect.Right, ARect.Top + h / 2);
  r := ARect.Left + (ARect.Right - ARect.Left) / 2;
  l := ARect.Left + (ARect.Right - ARect.Left) / 2;
  AGraphics.Stroke.Width := 2;


  if ADirection then
  begin
    AGraphics.DrawLine(PointF(r, ARect.Top), pl);
    AGraphics.DrawLine(pl, PointF(r, ARect.Bottom));
    AGraphics.DrawLine(pl, pr);
  end
  else
  begin
    AGraphics.DrawLine(PointF(l, ARect.Top), pr);
    AGraphics.DrawLine(PointF(l, ARect.Bottom), pr);
    AGraphics.DrawLine(pr, pl);
  end;
end;
```



Badges

Badges are used for indicating events. By default, the badge is a red circle with white font color and no value. It has a Hint property which can hold texts that will be shown when the date is hovered with the mouse. Badges can be set during designtime and programmatically.

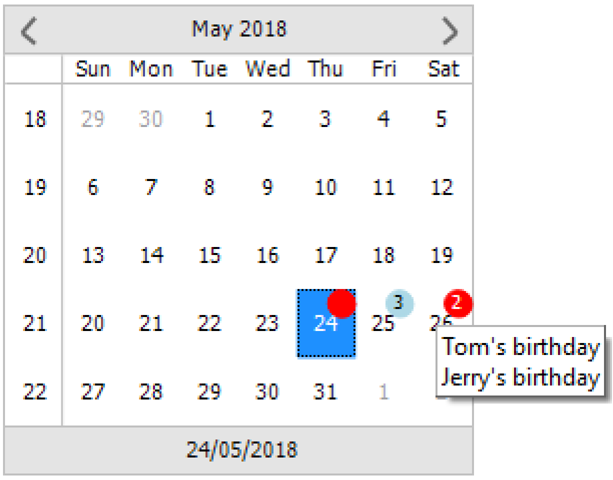
To set the badges during designtime, use the Events property of TMSFNCCalendar. With the Add New button, new badges can be added. The newly added badge will be set to today's date. Using its Date property, it can be moved to another date. A different color, font color, value and hint can be defined for each badge.

Just like at designtime, adding a new event/badge programmatically will set the event's date to today's date by default with the use of the Events.Add method. New events can be added and configured as shown in the code below:

```
procedure TForm1.FormCreate(Sender: TObject);
var
  I: Integer;
begin
  TMSFNCCalendar1.ShowHint := True;

  for I := 0 to 2 do
    TMSFNCCalendar1.Events.Add;

  //Leaving the first one as default, then:
  TMSFNCCalendar1.Events.Items[1].Date := Now + 1;
  TMSFNCCalendar1.Events.Items[1].BadgeColor := gcLightblue;
  TMSFNCCalendar1.Events.Items[1].BadgeFontColor := gcBlack;
  TMSFNCCalendar1.Events.Items[1].BadgeValue := 3;
  TMSFNCCalendar1.Events.Items[2].Date := Now + 2;
  TMSFNCCalendar1.Events.Items[2].BadgeValue := 2;
  TMSFNCCalendar1.Events.Items[2].Hint := 'Tom''s birthday' + sLineBreak + 'Jerry''s birthday';
end;
```

Alternatively the SelectEvent method can be used to add events by dates immediately:

```
procedure TForm1.FormCreate(Sender: TObject);
var
  I: Integer;
begin
  TMSFNCCalendar1.ShowHint := True;

  for I := 0 to 2 do
    TMSFNCCalendar1.SelectEvent(Now + I);

  TMSFNCCalendar1.Events.Items[1].BadgeColor := gcLightblue;
  TMSFNCCalendar1.Events.Items[1].BadgeFontColor := gcBlack;
  TMSFNCCalendar1.Events.Items[1].BadgeValue := 3;
  TMSFNCCalendar1.Events.Items[2].BadgeValue := 2;
  TMSFNCCalendar1.Events.Items[2].Hint := 'Tom''s birthday' + sLineBreak + 'Jerry''s birthday';
end;
```



May 2018

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	

Tom's birthday
Jerry's birthday

24/05/2018

TMSFNCDatePicker

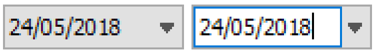
The TMSFNCDatePicker is using the TMSFNCCalendar as a selector. Therefore, most of the accessable properties and events are the same.

Date selection

With the SelectedDate property, the currently selected date can be set both at designtime and programmatically. This date is displayed in the picker itself:

```
TMSFNCDatePicker1.SelectedDate := Now;
```

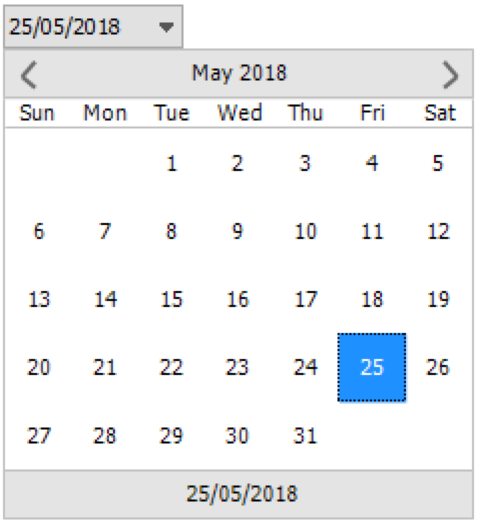
When the Editable field is enabled, text can be written inside the dropdown field. It requires the correct date format, otherwise it will reset back to today’s date when the selector popup gets opened. If the written date is correctly formatted, then upon opening the selector popup, the picker will set the selected date from the editable field.



24/05/2018 24/05/2018

The calendar selector's width and height can be set via the DropDownWidth and DropDownHeight properties.

```
TMSFNCDatePicker1.DropDownHeight := 235;
TMSFNCDatePicker1.DropDownWidth := 235;
```



Sun	Mon	Tue	Wed	Thu	Fri	Sat
		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		

Important methods, properties and events

Properties

BadgeSize	The BadgeSize property applies to all badges. It determines the diameter of the badge circle. The preset size is 14.
BitmapContainer	A container of images that is used in combination with arrows.
CurrentDate	Returns the current date.
Date	The Date property is used for navigation purposes. It's in sync with the Day, Month and Year properties.
DateAppearance	The appearance of the date area.
Day	The Day property is used for navigation purposes. It's in sync with the Date property.
DayNumberAppearance	The appearance of the weekday names area.
Events	A collection of event items. Each item's color, font color, value, hint and date can be set.
FirstDay	The FirstDay property determines the first day of the week. The values are the following: Sun = 1, Mon = 2, …, Sat = 6
FocusedDate	The FocusedDate is used to indicate the currently clicked date and to indicate movement by the keyboard.
Footer	The footer of the calendar that shows today's date by default. It can be configured to show the selected date and an additional caption can be added.
Header	The header of the calendar that shows the

	currently displayed month and year in text. The navigation arrows can be found here.
Interaction	The interaction capabilities of the calendar.
LineAppearance	The appearance of the lines that are separating the weekday names, the weeknumbers and the date area.
MaxDate	Setting the MaxDate will disable the selection of the dates that are after the MaxDate.
MinDate	Setting the MinDate will disable the selection of the dates that are before the MinDate.
Month	The Month property is used for navigation purposes. It’s in sync with the Date property.
SelectedDates	The collection of the selected dates.
WeekNumberAppearance	The appearance of the weeknumbers area.
Year	The Year property is used for navigation purposes. It’s in sync with the Date property.

Methods

ColAtX(AX: Single)	Returns the column under a specific X coordinate.
CurrentDay	Returns the current day.
CurrentMonth	Returns the current month.
CurrentYear	Returns the current year.
DateToCol(ADate: TDate)	Determines in which column the date will be, based on what is set for the first day of the week.
DateToRow(ADate: TDate)	Determines in which row the date will be, based on what is set for the first day of the week.
NextDate	Returns the next date (tomorrow).
NextMonth	Returns the next month.
NextYear	Returns the next year.
PreviousDate	Returns the previous date (yesterday).
PreviousMonth	Returns the previous month.
PreviousYear	Returns the previous year.
RowAtY(AY: Single)	Returns the row under a specific Y coordinate.
SelectADate(ADate: TDate)	Selects the date that’s given as the parameter. It will unselect all of the currently selected dates at the same time.
SelectDate(ADate: TDate)	Selects the date that’s given as the parameter. It will keep the currently selected dates at the same time.
SelectEvent(ADate: TDate)	Adds an event to the date that’s given as the parameter
SelectMultiDates(AStartDate, AEndDate: TDate)	Selects multiple dates in an interval from AStartDate to AEndDate. It will unselect all of the currently selected dates at the same time.
SelectedDate	Returns the selected date or the last selected date if there are multiple dates selected.
XYToDate(AX, AY: Single)	Determines the date at a specific X and Y coordinate.

Events

OnAfterDrawArrowLeft	Event called after the left arrow is drawn.
OnAfterDrawArrowRight	Event called after the right arrow is drawn.
OnAfterDrawBadge	Event called after a badge is drawn.
OnAfterDrawDayNames	Event called after a single weekday name is drawn.
OnAfterDrawDayNamesText	Event called after a single weekday name text is drawn.
OnAfterDrawDayNumbers	Event called after a single date is drawn.
OnAfterDrawDayNumbersText	Event called after a single date text is drawn.
OnAfterDrawFooter	Event called after the footer is drawn.
OnAfterDrawFooterText	Event called after the footer text is drawn.
OnAfterDrawHeader	Event called after the header is drawn.
OnAfterDrawHeaderText	Event called after the header text is drawn.
OnAfterDrawLines	Event called after the separation lines are drawn.
OnAfterDrawWeekNumbers	Event called after a single weeknumber is drawn.
OnAfterDrawWeekNumbersText	Event called after a single weeknumber text is drawn.
OnBeforeDrawArrowLeft	Event called before the left arrow is drawn.
OnBeforeDrawArrowRight	Event called before the right arrow is drawn.
OnBeforeDrawBadge	Event called before a badge is drawn.
OnBeforeDrawDayNames	Event called before a single weekday name is drawn.
OnBeforeDrawDayNamesText	Event called before a single weekday name text is drawn.
OnBeforeDrawDayNumbers	Event called before a single date is drawn.
OnBeforeDrawDayNumbersText	Event called before a single date text is drawn.
OnBeforeDrawFooter	Event called before the footer is drawn.
OnBeforeDrawFooterText	Event called before the footer text is drawn.
OnBeforeDrawHeader	Event called before the header is drawn.
OnBeforeDrawHeaderText	Event called before the header text is drawn.
OnBeforeDrawLines	Event called before the separation lines are drawn.
OnBeforeDrawWeekNumbers	Event called before a single weeknumber is drawn.
OnBeforeDrawWeekNumbersText	Event called before a single weeknumber text is drawn.
OnBeforeSelectDate	Event called before a date gets selected.
OnCustomArrowDraw	Event called when custom arrow is being drawn.
OnCustomNavigation	Event called when custom navigation is being called.
OnMonthChanged	Event called when the Month has changed.
OnSelectDate	Event called when a date gets selected.

OnSelectMultiDate	Event called when selecting dates using the SelectMultiDates selection.
OnYearChanged	Event called when the Year has changed.

TMSFNCDatePicker

DropDownHeight	Sets the dropdown’s height.
DropDownWidth	Sets the dropdown’s width.
Editable	Sets the picker’s date display area to an editable field.
SelectedDate	This property can be used to return and to set the selected date.
OnDateSelected	Event called when a date gets selected.