

Getting started with the TMS Planner for FireMonkey

Introduction

At TMS software, we have recently released a new scheduling control for cross-platform application development with the Delphi FireMonkey framework as part of the TMS Pack for FireMonkey (<http://www.tmssoftware.com/site/tmsfm-pack.asp>). While TMS Planner for FireMonkey design benefits from the significant experience the TMS team built-up with the award-winning VCL, IntraWeb and ASP.NET Planner components, the new component has been architected from the ground up for cross-platform, multi-device usage. This includes tuning for performance & footprint to run smoothly on Android or iOS smartphones and making the component flexible enough for complex scheduling applications in desktop or tablet applications.



TMS Planner for FireMonkey running on Windows, Mac OSX, iOS, Android

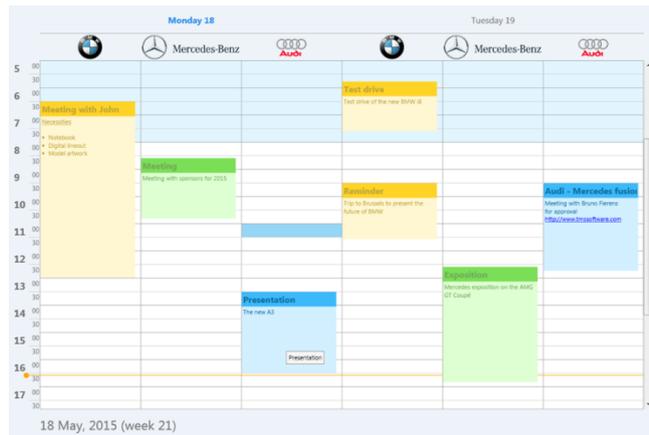
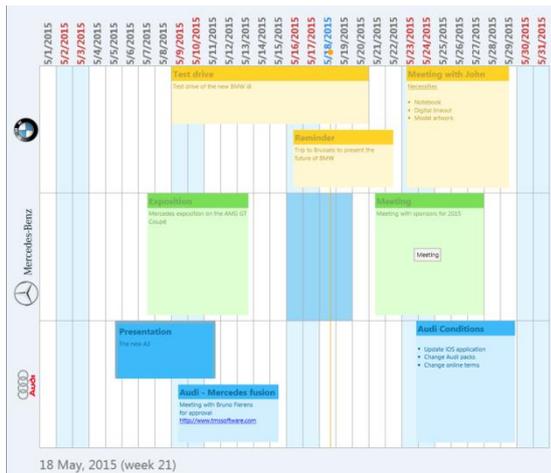
In a nutshell, the TMS Planner for FireMonkey, named TTMSFMXPlanner is a user-interface control representing a timeline and events for single or multiple resources.

Architecture

The three core parts of the TTMSFMXPlanner are: timeline, resources and items (events).

Timeline

The timeline determines the visible period of time on the TTMSFMXPlanner and is highly configurable. Default, it represents the standard 24hr timeline but this can be easily changed to month mode, a day period mode, a half-day period mode, multi-month mode or a custom timeline mode. With the custom timeline mode, the unit of time along the timeline can be fully programmatically set. In one of the included demos will be a timeline in seconds mode that represents a schedule of TV-adds. The timeline can be displayed on the left side, right side, left and right side or also at the top of the TTMSFMXPlanner in a 90° rotated view.

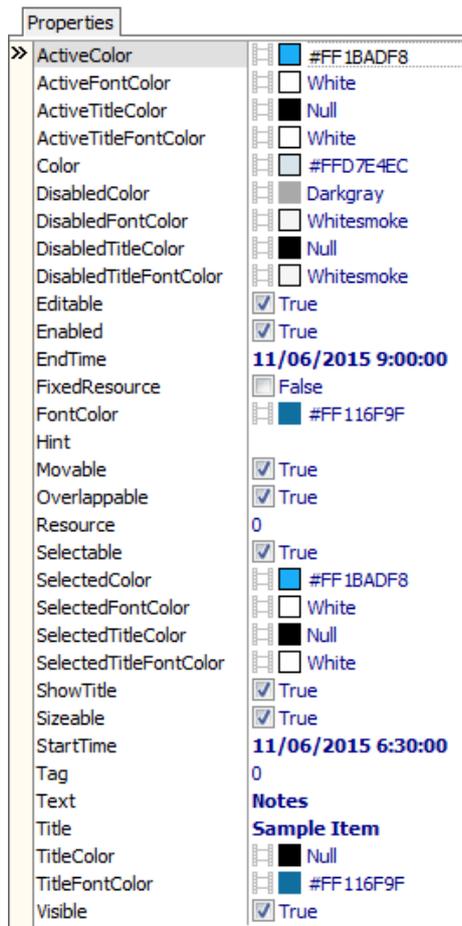


Resources

The TTMSFMXPlanner can display events for a single resource or can deal with multiple resources. But there is more. Views with multiple resources can be combined with multiple days. In this way, multiple resources per day and multiple days per resource can be shown. And in addition to this, there is also the flexible capability to create groups of a different number of resources or days, like grouping a number of days in a week or grouping a number of resources to a company etc...

Items

Finally, the TTMSFMXPlanner hosts a collection of items (events). The items have a start time, end time, title, notes, color, ... Items have settings to allow moving, sizing, for read-only, for allowing overlap or not etc... Items can be edited via inplace-editing or a built-in or custom dialog can be used for editing.



Overview of properties of an item in the TTMSFMXPlanner

Getting started

Drop a TTMSFMXPlanner instance on the form and you immediately see a view for 24hrs for 3 days. On a default TTMSFMXPlanner, you also see one item already added and a yellow line indicating the current machine time.

To programmatically create a new item, following code can be used:

```

Var
  it: TTMSFMXPlannerItem;
  pc: TTMSFMXPlannerCell;
begin
  // create the new item
  it := TMSFMXPlanner1.Items.Add;
  it.StartTime := Int(Now) + EncodeTime(14,0,0,0);
  it.EndTime := Int(Now) + EncodeTime(16,0,0,0);
  it.Title := 'New item';
  it.Text := 'Description';

  // programmatically select the item
  TMSFMXPlanner1.SelectItem(it.Index);
  // determine the timeslot where the item is placed
  pc := TMSFMXPlanner1.DateTimeToCell(it.StartTime);

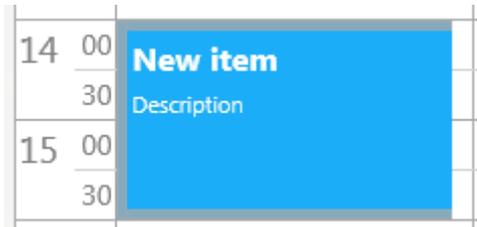
```

```

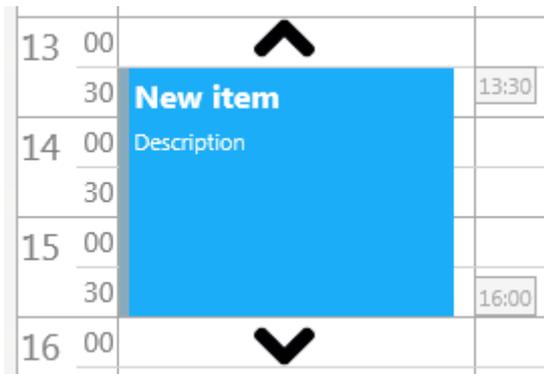
// scroll the planner to the timeslot where the item is
TMSFMXPlanner1.ViewRow := pc.Row;
end;

```

This inserts an item on today's date between 14h00 & 16h00. As an extra, the item is programmatically selected, the timeslot where the item is placed is retrieved and the TTMSFMXPlanner is vertically programmatically scrolled to the position of the item.



The user can drag the item by click & drag on the left side trackbar on a desktop system or by touch and drag on a mobile device. By grabbing the top or bottom trackbar, the item can be sized. On a mobile device, for sizing handy size-handlers are displayed for the selected item that help sizing with touch as the size-handler (arrow in this case) is a larger touch sensitive area for the finger. Notice that helper hints also appear during sizing, indicating the start and end time while sizing. The application is notified of items being moved or sized via various events like for example `OnAfterItemSize`, `OnAfterItemMove` from where the new start and end time can be retrieved.



By default, the TTMSFMXPlanner is smart enough to display the size handlers only for mobile touch based devices, but it can be enabled for desktop as well via setting `TTMSFMXPlanner.Interaction.SizeMode` to `psmMobile` instead of `psmAuto`. The size handlers itself are PNG images and can also be customized under class property `TTMSFMXPlanner.ItemAppearance` with `SizeHandlerDownBitmap` and `SizeHandlerUpBitmap`.

To allow the user to insert new items via the user interface, various options exist. First, an item can be inserted in the selected area from an external button with code:

```

var
  it: TTMSFMXPlannerItem;
begin
  it := TMSFMXPlanner1.Items.Add;

```

```

it.StartTime :=
TMSFMXPlanner1.CellToDateTime (TMSFMXPlanner1.Selection.StartCell);

it.EndTime := TMSFMXPlanner1.CellToDateTime (MakeCell (
TMSFMXPlanner1.Selection.EndCell.Col,
TMSFMXPlanner1.Selection.EndCell.Row + 1));
end;

```

Note the increment of 1 for the Selection.EndCell to retrieve the time at the end of the last timeslot in the selection.

Alternatively, with a simple property we can instruct the TTMSFMXPlanner to automatically create an item after selection with `TTMSFMXPlanner.Interaction.MouseInsertMode = pmimAfterSelection`. When an item is created this way, the `OnAfterInsertItem` is triggered when the item is created and from there the item can further customized:

```

procedure TForm1.TMSFMXPlanner1AfterInsertItem(Sender: TObject;
AStartTime,
AEndTime: TDateTime; APosition: Integer; AItem:
TTMSFMXPlannerItem);
begin
AItem.Title := 'Insert';
end;

```

Or a step further is to automatically bring up an editor dialog for the new created item and this is achieved by setting `TTMSFMXPlanner.Interaction.MouseInsertMode = pmimDialogAfterSelection`

The image shows a standard Windows-style dialog box with a light gray background and a blue border. It contains the following elements:

- Start Time:** A date dropdown menu showing '12/06/2015' and a time spinner control showing '18:30'.
- End Time:** A date dropdown menu showing '12/06/2015' and a time spinner control showing '20:30'.
- Title:** A single-line text input field.
- Text:** A large multi-line text area.
- Buttons:** 'Cancel' and 'OK' buttons located at the bottom right of the dialog.

Similarly, we can setup the TTMSFMXPlanner to automatically create and edit items on the selected timeslots when pressing the INS key by setting `TTMSFMXPlanner.Interaction.KeyboardInsertMode` to `pkimSelection` or `pkimSelectionDialog`.

In a similar way, via settings under TTMSFMXPlanner.Interaction, it can be configured how items are edited with mouse or keyboard interaction. TTMSFMXPlanner.Interaction.MouseEditMode defines whether the item goes in edit mode after a single click, double click or single click on an already selected item. When TTMSFMXPlanner.Interaction.KeyboardEdit = true, pressing F2 starts editing for the selected item. The type of editing, i.e. inplace editing or editing via a popup dialog is chosen with TTMSFMXPlanner.Interaction.UpdateMode set to pumInplace or pumDialog.

Timeline modes and resources

With this default setup, the TTMSFMXPlanner shows a multiday view with 24hrs for each day. The number of days in this mode is determined by TTMSFMXPlanner.Positions.Count. Within the 24hrs timeline, the default setting is that time between 8h00 and 20h00 is considered active time and displayed in 'Active' color whereas the inactive time is displayed in 'Inactive' color, configured respectively with TTMSFMXPlanner.Fill.Fill.Color and TTMSFMXPlanner.InActiveFill.Fill.Color. Displayed time and active time can be easily changed with TTMSFMXPlanner.TimeLine.DisplayStart, TTMSFMXPlanner.TimeLine.DisplayEnd, TTMSFMXPlanner.TimeLine.ActiveStart, TTMSFMXPlanner.TimeLine.ActiveEnd properties. Many other timeline modes are built-in also and set with TTMSFMXPlanner.Mode:

- pmDay: single day, multiple resources view
- pmMultiDay: multiple day view
- pmMultiDayRes: mixed multiple day / resources view
- pmMultiResDay: mixed multi resource / multiple day view
- pmDayPeriod: multiday per day timeline with multiresource view
- pmHalfDayPeriod: multiple day per half day timeline with multiresource view
- pmMonth: per month and per day with multiresource view
- pmMultiMonth: multiple months, single resource view
- pmCustom: custom timeline view

		BMW	Mercedes	Audi
11/06/2015	00:00 12:00			
12/06/2015	00:00 12:00			
13/06/2015	00:00 12:00			
14/06/2015	00:00 12:00			

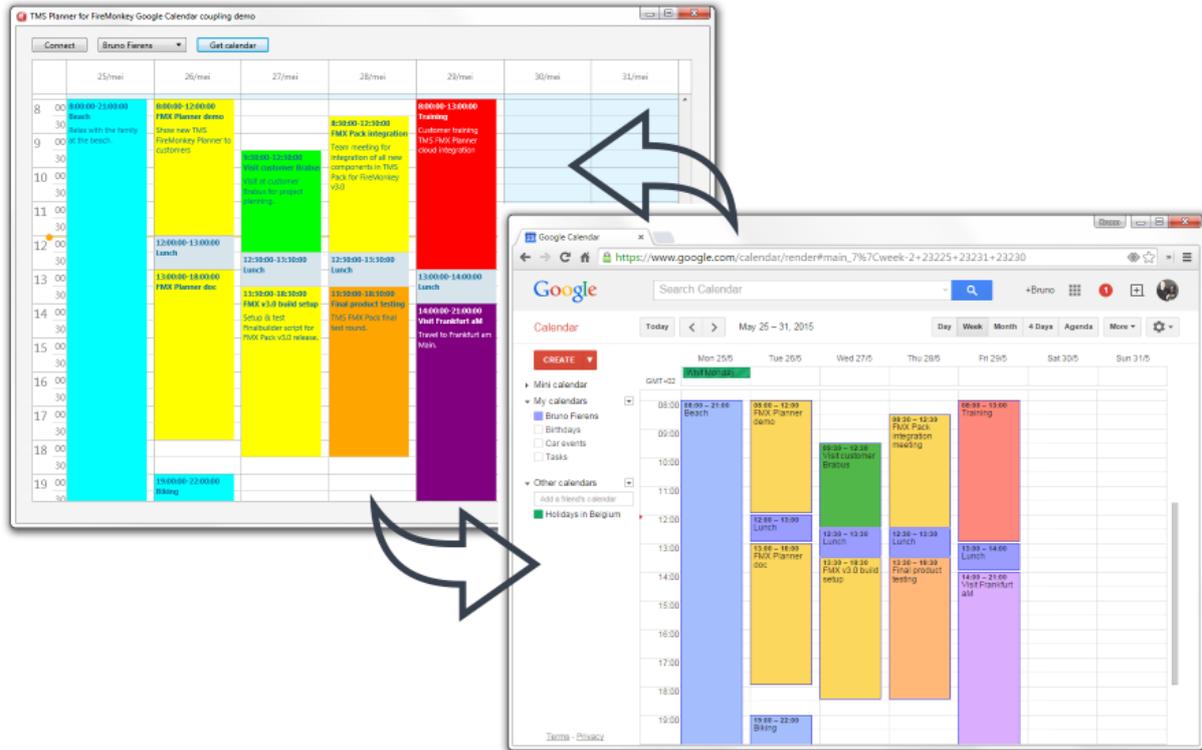
TTMSFMXPlanner in mode pmHalfDayPeriod showing 3 resources

In the TTMSFMXPlanner modes where multiple resources come into play, the resources are setup via a TTMSFMXPlannerResource collection under TTMSFMXPlanner.Resources. By default, there are already 3 resources added but this can be easily changed to the number of resources required. In a simple multiresource mode like the pmHalfDayPeriod mode in the screenshot above, when TTMSFMXPlanner.Positions is set to 3, it will list in the header for each position the resource

description found under TTMSFMXPlanner.Resources or when for a position no resource collection is defined, it will get the description "Position X". Items in the TTMSFMXPlanner are bound to a resource via the TTMSFMXPlannerItem.Resource property.

Connecting to the cloud

The TTMSFMXPlanner can, in combination with the TMS Cloud Pack for FireMonkey (<http://www.tmssoftware.com/site/tmsfmxccloudpack.asp>), be used to view or edit Microsoft cloud calendar or Google cloud calendar.



This is done via a non-visual adapter component TTMSFMXPlannerLiveAdapter or TTMSFMXPlannerGoogleAdapter. Drop this non-visual component on the form and connect it to TTMSFMXPlanner.Adapter. Depending on the cloud calendar you want to connect to, drop a TTMSFMXCloudGCalendar or TTMSFMXCloudLiveCalendar (from TMS Cloud Pack for FireMonkey) instance on the form and connect it to the LiveCalendar or GoogleCalendar property of the TTMSFMXPlannerLiveAdapter or TTMSFMXPlannerGoogleAdapter respectively. The first step is to connect to the cloud calendar and retrieve the available calendars. After selecting a calendar, getting the events from the cloud calendar in the TTMSFMXPlanner is as easy as setting the calendar ID and setting the property Active = true:

```
TMSFMXPlannerGoogleAdapter1.CalendarID :=
TMSFMXCloudGCalendar1.Calendars[idx].ID;
TMSFMXPlannerGoogleAdapter1.Active := true;
```

Default, the item properties StartTime, EndTime, Title, Text, Key are synchronized with the cloud calendar event properties. To synchronize more properties, the adapter offers the event OnGCalendarItemToItem and OnItemToGCalendarItem (in case of the Google cloud calendar). This

event handler demonstrates how the item's color can be mapped on the Google cloud calendar item color:

```
procedure
TForm1.TMSFMXPlannerGoogleAdapter1GCalendarItemToItem(Sender:
TObject; GoogleItem: TGCalendarItem; Item: TTMSFMXPlannerItem);
begin
    case GoogleItem.Color of
        icBlue: Item.Color := claBlue;
        icGreen: Item.Color := claGreen;
        icRed: Item.Color := claRed;
        icYellow: Item.Color := claYellow;
    end;
end;
```

The adapter further automatically takes care of updating, insert, deleting events on the cloud calendar in sync with manipulations on the TTMSFMXPlanner control.

Conclusion

With this brief introduction of the TMS Planner for FireMonkey, we have barely scratched the surface of the sheer number of capabilities and features. From appearance details to details of behavior, editing to configurations of timelines, current time display, overlap, ... everything is customizable to fine-tune the TTMSFMXPlanner to the particular requirements of your planning & scheduling applications. We invite you to look at demos, trial, tips and developer guide at <http://www.tmssoftware.com/site/tmsfmxpath.asp>