

OUTLOOK STYLE APPOINTMENT CALENDAR

Version 3.0 September 2012 for Access 2003 (or later)

By Peter D Hibbs.

Introduction

This document describes a method of creating a calendar form for use in an Access database which shows appointments on a monthly, weekly and/or daily basis. The design is intended to emulate the calendar used in MS Outlook as much as possible although it is not possible to copy every facility exactly due to the limitations of the built-in Access controls.

Calendar Operation

The demo calendar works like this for each of the three modes (although you will probably want to change some parts of it to suit your project). **Monthly** mode shows a box for each day of the current month as well as a few days of the previous and next month, where applicable. **Weekly** mode shows the seven days of the current week with the 48 half-hour time slots down the left of the form and **Daily** mode is similar to **Weekly** mode except that only the current day is displayed.

Monthly Mode

The main calendar form normally opens initially in **Monthly** mode (although you can change this to **Weekly** or **Daily** mode, if required) and shows the current month with today's date highlighted in scarlet. The current month dates are shown in green and the days from the previous and next months are shown in grey (but will still show any appointments and can be selected). The white fields show the contents of the appointment's **Subject** field for each date and are colored yellow if any appointment/s have been scheduled for that day. If there are two or more appointments scheduled for the same day, they are shown on separate lines which means that (with the current field height) only about five appointments can be shown for each day.

To display the **Weekly** or **Daily** modes you can click on the appropriate button at the top right of the form or you can click on the pale yellow **Wk No** field on the left of the form for the week to show **Weekly** mode or on the green date field (above the appointment fields) to show the **Daily** mode for that date.

If you click on an appointment field (in any mode) it will be highlighted in pink to show which date is currently selected and to create a new appointment (or to delete or modify an existing appointment) you should double click on the appointment field which will display a pop-up **Appointment Schedule** form (see below for details).

To move around the calendar you can click on any appointment field, as normal, and you can also use the keyboard to move the focus to a different date. The UP and DOWN arrow keys will move up and down the fields and the LEFT and RIGHT arrow keys will move left and right (note that in **Weekly** mode you need to press the TAB key first in order to use the Left and Right arrow keys to move between dates). Also, when a date field has the focus, you can press the ENTER key to open the **Appointment Schedule** form for that date (and time).

Weekly Mode

In **Weekly** mode, the date at the top of the form shows the date of the first day of the selected week and the seven days of the week are shown along the top of the form with the time slots shown down the left side of the form which run from 00:00 to 24:00 hours at half hourly intervals

(see below if you need to change this time slot period). You can click on any of the seven date labels above the form to switch to **Daily** mode for the selected date. Also, the day and date label is highlighted in scarlet for today's date.

As in **Monthly** mode, the current time slot is colored pink, used time slots are highlighted in yellow and you should double click a time slot to add, delete or change an appointment. Also, for **Weekly** and **Daily** modes only, you can add, amend or delete appointments directly from the keyboard (see below for more details on how this works).

The time slots from 00:00 to 08:00 hours and also from 17:00 to 24:00 hours are colored light grey to indicate the normal office out-of-hours periods but these can be changed to different times, if required (see later). The **Times** column on the left shows the first 12 hour period in green and the second 12 hour period in pale green and the half hour slots are left blank (as in Outlook) but this can easily be changed, if required (see later).

If there are two or more appointments scheduled for the same time slot, the contents of the **Subject** field are shown in the cell separated by a space character. Of course, this means that it may not be possible to show the second or third appointment in the field, depending on the length of the text.

In **Weekly** (and **Daily**) mode, there are a total of 48 times slots for the 48 half hour times in the day. Whenever the user switches to one of these modes the calendar form is automatically scrolled down until the first time slot shows as 07:00 and the last as 21:30 hours so that the normal working hours are shown on the form. It is possible to change the time slot period so you may want to change the way this feature works and this is described in more detail later.

Daily Mode

The **Daily** mode is basically the same as **Weekly** mode except that only one day's appointments are shown which means that multiple appointments in the same time slot can be shown in the appointment field and since this field is a lot wider in this mode, it possible to show a lot more text and/or multiple appointments.

Also, the day - date label at the top of the form will be shown in scarlet for today's date.

Add, Amend or Delete an Appointment

To add a new appointment or to amend or delete an existing appointment you should double click on the relevant appointment date field (in whichever mode is appropriate) to open the pop-up **Appointment Schedule** form.

This form allows you to create a new appointment or modify or delete an existing one. It also shows a list of appointments for the selected date even if they start and/or finish on a different day. When the form opens, both the **Start Date** and **End Date** fields default to the currently selected date and the **Start Time** and **End Time** fields default to 08:00 – 08:30, (if the form is opened from the **Weekly** or **Daily** mode calendars the **Start Time** field is set to the time slot that is clicked on in the main calendar and the **End Time** field is set to the **Start Time** plus 30 minutes or whatever time period has been specified). The various input fields are initially disabled.

Add New Appointment

To create a new appointment click the **New Appt** button to enable the input fields and set the **Start Time** and **End Time** fields to the required times and enter some text into the **Subject** field (which is mandatory) and this text will appear on the calendar. Enter text into the **Location** and/or **Miscellaneous Notes** fields, as required. Click the **Save Appt** button to save the new appointment, the form will close and the new appointment will be displayed on the calendar. Note that you can

also change the **Start Date** and/or **End Date** fields if the appointment needs to be run over two or more days. There are two options to change the start and end dates, you can enter the date into the fields directly or you can click on the small calendar buttons to the right of the two date controls and select a date in the pop up calendar form. For Access version 2007 and later this facility is built in when you move focus to the date fields so you may want to either hide the two buttons and use the Access facility or disable the Access built –in facility and use the buttons provided instead.

Amend an Existing Appointment

To amend an existing appointment, double click the relevant date on the main calendar form and when the **Appointment Schedule** form opens, click on the required appointment shown in the list box, the input fields will be enabled and the details for the appointment copied into those fields. Make any changes and click the **Save Appt** button to update the appointment and close the form.

In this demo version it is possible to make two or more appointments for the same time slots. It is possible, however, to change this procedure so that only one appointment can exist for each time slot (see below for more details). If this option is set then you will be shown an error message if the new appointment (or an amended one) overlaps any already existing appointment.

Delete an Existing Appointment

To delete an existing appointment do the same as above but click the **Delete Appt** button to delete it (after confirming the deletion) and close the form.

Find an Appointment

To find an appointment you can click the **Find Appointment** button on the main calendar form to open the **Find an Appointment Record** form. Initially the list box on this form shows every appointment (sorted on the **Start Date** field) and you could just scroll down the list to find the one you want to show, double click on it (or single click on it and then click the **Find** button) which will close the form and move the calendar to the required record. The cursor will move to the date and/or time slot (depending on which mode is in use) and highlight the appropriate date field. When the appointment has been found the **Appointment Schedule** form will open automatically for that appointment so that you can see the appointment details. If you don't need this option, it can easily be disabled (see below for details).

Alternatively, to find a specific record, you can enter some text into the **Filter on Subject** field (or the **Filter on Location** field) which will then update the list of appointments where the **Subject** field matches the text entered. In the case of the **Location** field the list will always show appointments where this field has been left blank. Depending on your application, you may need to change the method of filtering the list in order to show your appointments.

Import from or Export to Microsoft Outlook

The **Import from Outlook** and **Export to Outlook** buttons allow you to import any appointments made in MS Outlook to your calendar table or export your appointments to MS Outlook. This is a fairly basic import/export system and is described in more detail later.

Print Monthly Calendar

The **Print Calendar** button shows the **Monthly** calendar in **Print Preview** mode which can then be printed, if required. Again, this is fairly basic and is included just to show how you could print a calendar if your application needs this facility.

Calendar Navigation

The **Today** button resets the calendar display so that it shows today's date in whatever mode is currently being used.

The **Prev Month** and **Next Month** (which show as **Prev Week/Next Week** or **Prev Day/Next Day** in other modes) buttons move the calendar display to the previous or next time period.

The small calendar button just to the left of the **Monthly** button at the top of the form opens a small pop up calendar form which allows you to select any date and display the appropriate month/week/day on the main calendar.

Calendar Design

To ensure that the tables are properly normalised, each individual appointment is stored as one record per appointment in the appointments table. The problem with this is that the appointments need to be displayed on screen in a calendar format (monthly or weekly or daily) which is not easy when there are many single records in a table.

A method often used for a monthly display like this is to have a Text box for every day of the month (that is 42 fields to cater for every month configuration) but instead, the method used in this demo uses a sub-form control and a temporary table to show the monthly data which then only requires 7 Text box controls on the sub-form (actually it uses 15 fields because the current week number and the 7 dates of the week are shown above each appointment field). When a particular month (or week or day) is displayed on screen the data from the main appointments table for the selected period is processed with some VBA code and copied into the temporary table and then the screen is updated to show the relevant appointments. The **Weekly** and **Daily** modes work in a similar way but use a different table to show the days of the week and the sub-forms show the time slots for each appointment with the sub-form scrolled up or down to show all 48 time slot rows. The methods used are described in more detail below.

To store the appointment information, each appointment is stored in one record in the table **tblAppointments**, the fields used are :-

ApptID	AutoNumber	Used to identify an appointment record.
ApptSubject	Text	Stores appointment subject (required).
ApptLocation	Text	Stores appointment location (optional).
ApptStart	Date/Time	Stores the start date and time of the appointment (required).
ApptEnd	Date/Time	Stores the end date and time of the appointment (required).
ApptNotes	Memo	Stores other miscellaneous notes (optional).

As with MS Outlook, appointments can span several days or weeks although the method of showing these on the calendar is different as Access does not have the flexibility to do this in the same way. If an appointment runs over more than one day (or more than one time slot in the case of the **Weekly** and **Daily** modes), the appointment subject is repeated for each day (or time slot).

The calendar display consists of a main form (**frmCalendarMain**) and three sub-forms (**frmCalendarMonth**, **frmCalendarWeek** and **frmCalendarDay**), these forms are placed on three pages of a Tab control (**tabCalendar**). To switch between views, the appropriate tab is selected by clicking on one of the three toggle buttons at the top of the form (or by clicking on certain fields on the calendar) which then display the required calendar sub-form. Note that the tabs on the Tab control are hidden so that the Tab control itself is not visible to the users. If you need to change the design of any of the sub-forms the easiest method is to open the relevant sub-form in Design mode from the Database Forms window (or you can select the appropriate sub-form in the **Objects** drop down box on the Access Formatting tool-bar).

Calendar Display Procedures

A brief description of the VBA code used to show the calendar data is described below, for more detailed information see the code itself which is fairly well documented. A hidden Text box control on the main form (**txtMode**) holds a value of 1, 2 or 3 for modes **Monthly**, **Weekly** or **Daily** respectively. Whenever a calendar date is changed, one of three separate procedures is called (depending on the current mode) which reads the appointment data from the appointments table for the period of time selected, it then writes that data to a temporary table (that is 'temporary' in the sense that its contents change each time the routine is called) and then refreshes the sub-form to display the new appointment data on screen. The three sub-forms are bound to the temporary tables (actually the **Weekly** and **Daily** modes share the same table) and therefore show the contents of the table in the same format as the table layout. Another hidden Text box control (**txtCurrentDate**) holds the currently selected date and is updated every time the user clicks on a date field on the calendar. This date field is used to calculate the first date of the month, first date of the week or the current day's date which is then stored in the **txtDate** control shown at the top of the calendar.

Most of the code that is used to update the temporary tables is stored in the code module **modCalendarCode**. This arrangement makes it a bit easier to find the main VBA code and make any changes to the routines, as and when required.

Monthly Mode

Whenever the **Monthly** calendar is required to be displayed, the sub-routine **ShowMonthAppts** is called and the first date of the current period (from control **txtDate**) is passed to the routine.

This routine (in code module **modCalendarCode**) does the following :-

- Calculates the date of the first Sunday to be shown on the calendar (which could be a date at the end of the previous month depending on the month and year being shown).
- Calculates the week numbers for the selected month and copies them to the **WeekNo** field in the six records in **tblMonthData**.
- Calculates the date of each day field to be shown on the calendar and copies them to the **Day1Date** to **Day7Date** fields in **tblMonthData**.
- Creates a record set of all appointment records where any date of an appointment falls within the selected month period (including dates in the previous or succeeding months where they would appear on the calendar display).
- Iterates through the record set one appointment record at a time and copies the appointment information (i.e. the **Subject** field) to a temporary array (which corresponds in size to the number of records in **tblMonthData**).
- Copies the contents of the array to the relevant fields in the **tblMonthData** table. Using a temporary array in memory speeds up the process when writing the complete set of data to the table since that then only needs to be done once.

On return from the routine, the subform (**frmCalendarMonth**) is requested to display the new contents of **tblMonthData** on screen.

This procedure runs reasonably quickly because, although the appointments table itself may have many thousands of records, only those for the current time period (i.e. 42 days in this mode) are returned by the query and copied to the temporary table.

Weekly Mode

The sub-routine used for this mode is called **ShowWeekAppts** and is very similar to the above code (except that the date passed to the routine is the first date of the selected week and the temporary table is called **tblWeekData**). The code works like this :-

- Calculates the date of the first Sunday to be shown on the calendar.
- Creates a record set of all appointment records where any date of an appointment falls within the selected week period.
- Iterates through the record set one appointment record at a time and copies the appointment information (i.e. the **Subject** field) to a temporary array (which corresponds in size to the number of records in **tblWeekData**).
- Copies the contents of the array to the relevant fields in the **tblWeekData** table.

In this mode, the temporary table, **tblWeekData**, has a record for each time slot which is 48 in this demo version (see below on how to change the time slot period). Note that the times shown in the **TimeSlots** field are not changed in this routine because once they have been entered, they do not need to be changed each time the **Weekly** (or **Daily**) calendar is displayed.

Daily Mode

The sub-routine used in this mode is called **ShowDayAppts** and is almost identical to the weekly mode routine except that only one field in the table is used, i.e. field **Day1Data**.

Installing the Calendar

To use this calendar in your database project you should do the following :-

- First compile your VBA code to ensure that there are no existing compile errors (fix them now if you can).
- Import the three tables, the query, the seven forms, the report and the three modules into your database from the demo database.
- Now compile the VBA code again to ensure that there are no duplicate definitions (you will have to change them if there are).
- In the **Database – Forms** window, double click the form **frmCalendarMain** and check that everything is working correctly. When you want to display the calendar in your project, you should open this form from your code.

If your database is split into a Front-End file and a Back-End file (as it should be) the table **tblAppointments** must be moved to the Back-End and linked to the Front-End as normal, the tables **tblMonthData** and **tblWeekData** must stay in the Front-End file. If the database is not currently split, when you do split it before rolling it out to your users, make sure you remember to move the two data tables back into the Front-End (assuming you used the Wizard to split the files which usually moves all the tables to the Back-End).

If you already have a table for appointments in your database which you need to keep, then you will either have to change the table name and field names in your existing table to be the same as the names used in the demo database table or you will have to go through all the imported forms, report and VBA code to change the table and field names to be the same as your existing table and field names (which would be a major task).

If you don't actually need the facilities to import/export MS Outlook appointments then you can delete the module **modOutlookCode** and delete the two buttons and associated code on the main form.

Also, if you don't need the report, then just delete the report and the relevant button on the main form.

Changing the Calendar Layout

Although this demo calendar does most of what the MS Outlook version does, it is quite likely that you will want some parts of it changed to suit your own requirements. This section shows some aspects of the basic design layout that can easily be changed (see also **Configuring the Calendar** section below).

Resize the Calendar Forms

The demo version was designed to fit on a screen resolution of 1024 x 768 pixels which is the most common screen size (on older PCs anyway). If you are using a bigger screen size and you want to make some of the fields bigger in order to show more text, you can change the size of the forms and fields to suit. However, this can be a bit fiddly because the three sub-forms are embedded in three pages on a Tab control which has its tabs hidden.

Probably the easiest method is, in Design mode, to first resize the main form (**frmCalendarMain**) to the size you want and then select the tab control (**tabCalendar**) in the drop down field on the Access Formatting toolbar, make that the full size of the Detail section of the form and then select each sub-form in turn (i.e. **frmCalendarMonth**, **frmCalendarWeek** and **frmCalendarDay** again in the drop down box) and resize them accordingly. When the sub-forms are the required size you should open each sub-form in Design mode in the **Database -> Forms** window and then resize the individual fields to fit on the form. Unless you really need to do this it would probably be easier to leave them as they are.

Appointments Navigation in Monthly Mode

When in **Monthly** mode you can just click on any date cell to make that date the current date and you can double click on a date cell to open the **Appointment Schedule** form for that date.

You can also use the keyboard arrow keys (in all modes) to move up and down or left and right on the grid (the TAB or SHIFT + TAB keys will also move the cell focus). If you press the ENTER key while a date cell has the focus, the **Appointment Schedule** form will open in the same way as if you had double clicked on the cell (although you can easily disable this option, if required, see below for details).

As mentioned above, if you click on any of the six yellow **Week No** fields at the left of the grid the calendar switches to **Weekly** mode for that week or if you click on any of the 42 green date fields above the appointment cells, the calendar will show the **Daily** appointments for that day. In order to implement this facility a 'transparent' button is placed over the top of the seven date controls above the date cells and some VBA code is used to determine which day of the week the user clicked on which then selects the appropriate date in **Daily** mode. A similar technique is used for the **Week No** control in **Monthly** mode and the **Date** controls in **Weekly** mode. The reason for this is that by using an invisible button the cursor shows as an arrow icon rather than an I-Beam icon if the controls themselves were used to trigger the code (which looks better in my opinion).

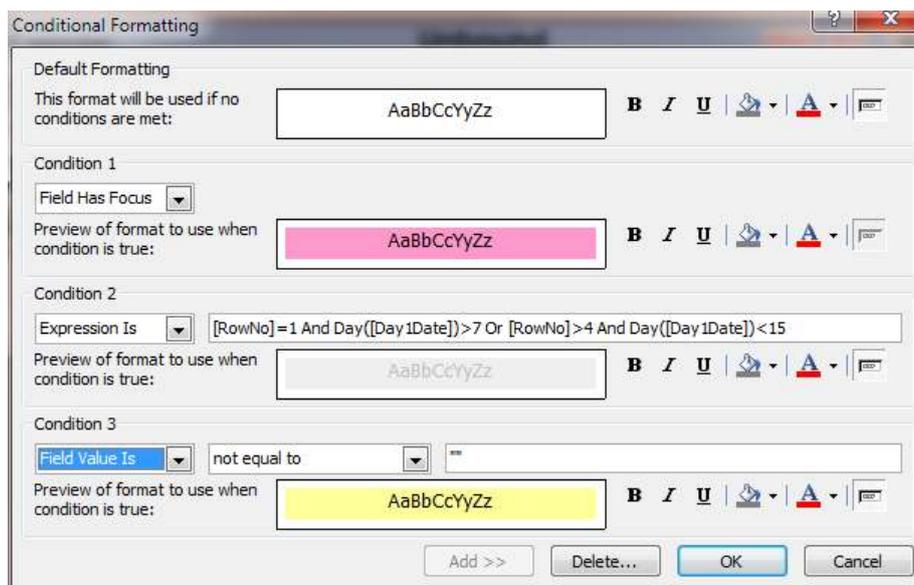
Monthly Calendar Colors

As the sub-forms use **Continuous** type forms to show the appointment data the method of showing the various fields in different colors makes extensive use of the **Conditional Formatting** facilities.

For example, the date fields for each appointment are normally colored pale green but are shown as light grey for dates that fall outside the current month and one is shown in scarlet if the date is the current date. Note that in the case of the second condition, the default grey color available on

the **Conditional Formatting** form is a bit too dark to show on a form so the color for Condition 2 is changed using some VBA code and this can be found in the **Open** event of the three sub-forms. If you should need a different color for this condition (or any of the conditions) you can change the color value, as required. The constant, **conLightGrey**, is defined in the main VBA code module – **modCalendarCode**.

Similarly, the seven appointment fields are also formatted to show similar colors except that in this case, the field is colored pink if it has the focus, light grey if the date is outside the current month and yellow if it is not blank. The screen shot image below shows the **Conditional Formatting** for field **Day1Data**.



The conditions are set as follows :-

- The Default condition shows the field in white.
- Condition 1 is used to show which field has the focus in pink and has the highest priority so that it takes preference over the other conditions.
- Condition 2 colors the field light grey if the date falls in the previous month or in the next month. This condition uses the value in the **RowNo** field and the date of the field to calculate whether it should be colored grey or not. The **RowNo** field is a hidden field on the form and holds the number of the row, i.e. 1 for the top row and 6 for the bottom row. For this condition, the field's text color is also set to grey for the dates that are not in the current month which you could also change. For example, if you want the date or the appointment text to show darker, you could change the color to black or if you do not want the text to show at all, you could change it to light grey (although in this case you would have to add a bit more code to the form to change the color). See also next section if you do not want to show appointments in cells for the previous or next month dates. The one other option on this condition is that you can disable the field altogether. For example, if you do not want the users to be able to create, amend or delete appointments for dates that are not within the current month you can disable this field when Condition 2 is active by setting a flag in the main code module, see below for details.
- Condition 3 colors the field yellow if there is any text in the field and is used to indicate which dates have scheduled appointments. This condition has the lowest priority.

Any date cell could have multiple appointments stored in them and with the current control size in **Monthly** mode, it is only possible to display about five lines of text. If you need the users to be

able to see all the appointments in a cell you have three options. You could make the controls bigger (as explained above) or you can press SHIFT + F2 when the cell has the focus to show the standard Access pop-up **Zoom** form which shows the contents of any field or you could set the **Scroll Bars** property of the seven appointment data fields to 'Vertical' so that they could scroll the field up and down when it has the focus.

If you are saving a lot of appointments for each date then showing the **Subject** for each appointment would probably not be very helpful to the end users. In this situation you could just show a count of how many appointments there are currently scheduled for the day and use the **Weekly** and/or **Daily** modes to show the individual appointments. It is possible to enable this facility by setting a flag in the configuration module, see next section for details.

Configuring the Calendar

As well as changing the calendar layout described above there are a number of small changes that you may want to make to the calendar design to suit your own project. Most of these would entail changes to the VBA code or to the properties of one or more form controls. To make this easier to implement there are 14 'flags' which can be changed to enable/disable various facilities on the calendar forms.

To change these you should open the module **modCalendarCode** and look for the 'Code Configuration Constants' section at the top of the module. Change a flag (as described in the following sections) and save the module. Note that some facilities do not take effect until the calendar form is opened so it is usually best to make any changes to the flags when the form **frmCalendarMain** is not loaded.

Hide Appointments that are not within the Current Month

On the **Monthly** calendar any dates that fall in the previous month at the top of the calendar or in the next month at the bottom of the calendar are colored grey and also show any appointments on those days. If, however, you do not want to show those appointments you can amend the code that copies the appointment data to the table to omit any that do not fall within the currently specified month.

In the 'Code Configuration Constants' section change the constant **conMonthHide** from 0 to 1. Note that this also disables the option to double click on those date fields to open the **Appointments Schedule** form.

Change the Week Start Day

The demo version uses Sunday as the first day of the week but you may need to have the week start on a Monday (or any other day). This can be done by changing the constant **conFirstDay** from vbSunday to vbMonday or vbTuesday, etc depending on which day you want to have appear first on the **Monthly** and **Weekly** calendars. Note that this also changes the report layout as well.

Change the Time Slot Period in Weekly and Daily Modes

In this demo version, in **Weekly** and **Daily** modes, the time slot period is set to 30 minutes but if you need a different period you can change this very easily by changing the constant **conPeriod** to a different value. If you want to change it to, say, 15 minutes (or whatever) just change the constant value to 15. For example :-

```
Public Const conPeriod = 15
```

Note: when setting this value it needs to be a value which divides into 60 exactly, 5, 6, 10, 15, 30 and 60 minute periods are acceptable. If you really need a different period to any of these then you

will probably have to redesign some of the code to cater for that. When you have changed the **conPeriod** value, save the module and then open the calendar, as normal.

What happens then is this – whenever the main calendar form is opened the **InitialiseTable** sub-routine (in module **modCalendarCode**) is called from the **Open** event of the main form. This routine compares the number of records in **tblWeekData** with the number of records that should be in the table based on the value of the constant **conPeriod**. If they match, as they normally would, the routine exits immediately. If, however, the values don't match because the **conPeriod** has been changed (or maybe a record has been deleted from the table) then all records in the table **tblWeekData** are deleted and the correct number of records is added to the table and various fields are re-initialised. The number of records added is based on the number of time slots required which is determined by the value of the **conPeriod** constant.

If you change the time slot period, the new time slots will be automatically displayed on the **Weekly** and **Daily** calendars the next time the main calendar form is opened. However, it is probably not a good idea to change the time slot period after you have entered a lot of appointments in the table because the calendar displays may not appear in the way that you want them to (although you would not actually lose any appointment records). You would probably also need to change some other parameters such as the out-of-hours time colors and the first time slot to be displayed on the forms (see below for details).

Change the Time Slot Color in Weekly and Daily Modes

In **Weekly** and **Daily** modes there is a row for each time slot period which is set to 30 minutes (as in Outlook). The **Times** column for the first 24 time slots (12 hours) are colored dark green and the second 24 time slots are colored pale green to make it easy to see which part of the day is being displayed. If you would prefer to have all time slots shown as dark green you can change the constant **conMidday** to 23:59 or if you prefer all in pale green, change the constant to 00:00. If you need different colors for the **Times** field you will need to change the **BackColor** property of the **Times** field on the two forms (**frmCalendarWeek** and **frmCalendarDay**) and also change the value of the constant **conPaleGreen** in the module **modCalendarCode**.

If you should change the time slot period from 30 minutes to some other value (see below for details) the VBA code will automatically adjust the settings to show the time slots from 12:00 onwards in the pale green color. If, however, you would like the color to change at a different time slot you can change the value of the constant **conMidday** to something other than 12:00 (it must be a valid time used on the calendars) and this will affect both **Weekly** and **Daily** modes.

Change the Time Format in Weekly and Daily Modes

Currently the time format in the **TimeSlots** field is set to 'Short Time' which shows a 24 hour time format. If you prefer to show a 12 hour time format then just change the **Format** property of this field to 'Medium Time' (or whatever other format you want).

Change the Time Slot Display in Weekly and Daily Modes

The **Times** column currently shows the alternate rows as blanks (as MS Outlook does) but if you want to show the time in every row you need to fill in the blank rows with the appropriate time value (which will, of course, depend on what time slot period you are using).

You can do this by changing the constant **conAltTime** from 0 to 1. In this mode all time slots will show a time, depending on the time period that you are using.

Change the Out-Of-Hours Time Slot Back Color in Weekly and Daily Modes

On the **Weekly** and **Daily** modes, the back color of the time slots for 00:00 to 08:00 and time slots 17:00 to 24:00 are colored light grey to indicate out-of-office hours and the periods within the normal office hours are colored white (unless there are any appointments scheduled). Of course, this may well be different for your situation so you may want to change those periods.

You can do this by changing the two constants **conFirstTime** and **conLastTime** to the time slots that you want to show in white. The default times are 08:00 and 16:30 which means that the time slots between these two times are colored white and the time slots up to the first and after the last ones are colored light grey. To set different times just change these two constants to the appropriate times, note that the constants must be in the 24 hour format.

You could also change the light grey color, if required, by changing the constant **conLightGrey** to a different value. Note that this color value will affect all three calendar modes.

Change the Time Slot Row Positions in Weekly and Daily Modes

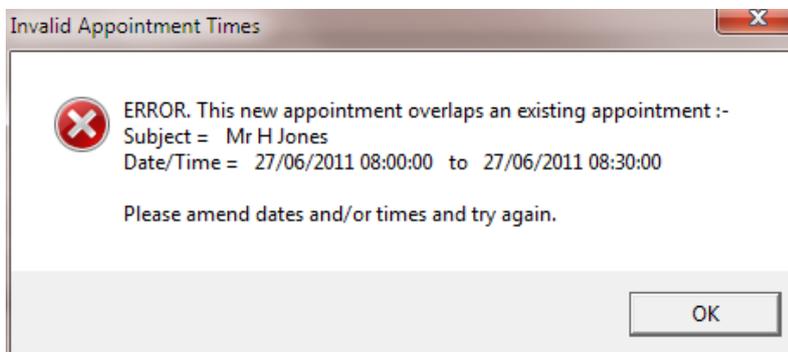
When the **Weekly** and **Daily** modes are displayed (using the current form size) the time slot for 07:00 is shown at the top of the form and time slot 21:30 is shown at the bottom. If you need a different range of time slots (and you almost certainly will if you change the time slot period) then you can do this by changing the constant **conTime** from 07:00 to the appropriate time value. The time you select will then be the first time slot to be shown at the top of the form in **Weekly** and **Daily** modes.

Multiple Appointments for the same Time Slot

Microsoft Outlook allows the user to schedule two (or more) appointments for the same time slot period, although it does warn you that there are overlapping appointments and in this demo version it is possible to do the same thing.

However, for some applications this could be a major problem. For example, if you want to create a doctor's surgery appointments or job interviews, etc you would not want the calendar user to be able to schedule two appointments for the same time. To prevent this you can change the constant **conMultiAppts** from 0 to 1 which will prevent two different appointments being scheduled for the same time slot.

Now, whenever the user attempts to create a new appointment (or amend an existing appointment) that overlaps another existing appointment an error message will be displayed with some details of the existing appointment. See example message box below.



Of course, this will not check any existing appointments so you should make sure that there are no overlapping appointments already existing in the appointments table before activating this facility.

In the code, the function **AppointmentsCheck** is called and the new start and end times are passed to the function along with the **AppointmentID** of the current appointment (this is only needed when you modify an existing appointment). The function compares the new date and times with all

existing appointments stored in the table and if the new times overlap any existing appointment record, the **AppointmentID** of that record is returned or it returns zero if no conflict is found. If a non-zero value is returned, the code then looks up the information about the existing appointment and displays it to the user (as shown above).

Edit Weekly and Daily Appointments Directly

As mentioned earlier, you can add, amend or delete appointments by double clicking on any appointment date field to open the **Appointment Schedule** form and enter the appropriate information there.

However, if you have opted to have only one appointment per time slot and you do not need an appointment to span more than one time slot and you only need to enter a subject for the appointment (i.e. Location and Notes fields are not required), then you can enter appointment information directly into the calendar for **Weekly** and **Daily** modes. This option has only be provided for those modes because it would only work for **Monthly** mode if you only ever had one appointment per day at 08:00 to 08:30 (which seems unlikely).

It works like this - in **Weekly** and/or **Daily** mode, just click in a time slot cell and enter some text, when you press the TAB key the text will be stored in the appointments table for the default time slot period of 30 minutes (or whatever you have set it up to be). The **Subject** field will hold the text that was entered and will be displayed on the calendar although the **Location** and **Miscellaneous Notes** fields will stay blank. Of course, you can still double click on an appointment to change it using the **Appointment Schedule** form, if required.

To amend an existing appointment subject, just click on the appointment and edit the text accordingly and to delete an appointment altogether, click the appointment and delete all the text in the cell and in this case, the appointment record will be deleted from the appointments table with no confirmation being requested. Note that if you press the ENTER key instead of the TAB key you will open the **Appointment Schedule** form but this can also be disabled (see next section).

If you do not want this facility to be used (because you need multiple appointments per time slot, multiple time slots per appointment or you always need to enter location or notes information) then you should disable this option by changing the constant **conDayEdit** from False to True.

Enable/Disable ENTER key on Time Slots

If you click in any date field or time slot in any of the three modes, the cell will be highlighted in pink. If you now press the ENTER key the **Appointment Schedule** form will open in the same way as if you had double clicked on the cell.

However, if you do not want this to happen you can disable the facility by changing the constant **conEnterKey** from 0 to 1.

Show Appointments Count on Monthly Calendar

If you are saving a large number of appointments in each day then the **Monthly** calendar does not really show the appointment information very well as there are too many appointments being displayed in each date cell.

An alternative method is to show the total number of appointments for each day (in this example they are split between AM and PM appointments) on the **Monthly** calendar and then use the **Weekly** or **Daily** calendars to display the details of the appointments. To do this change the constant **conMonthCount** from 0 to 1.

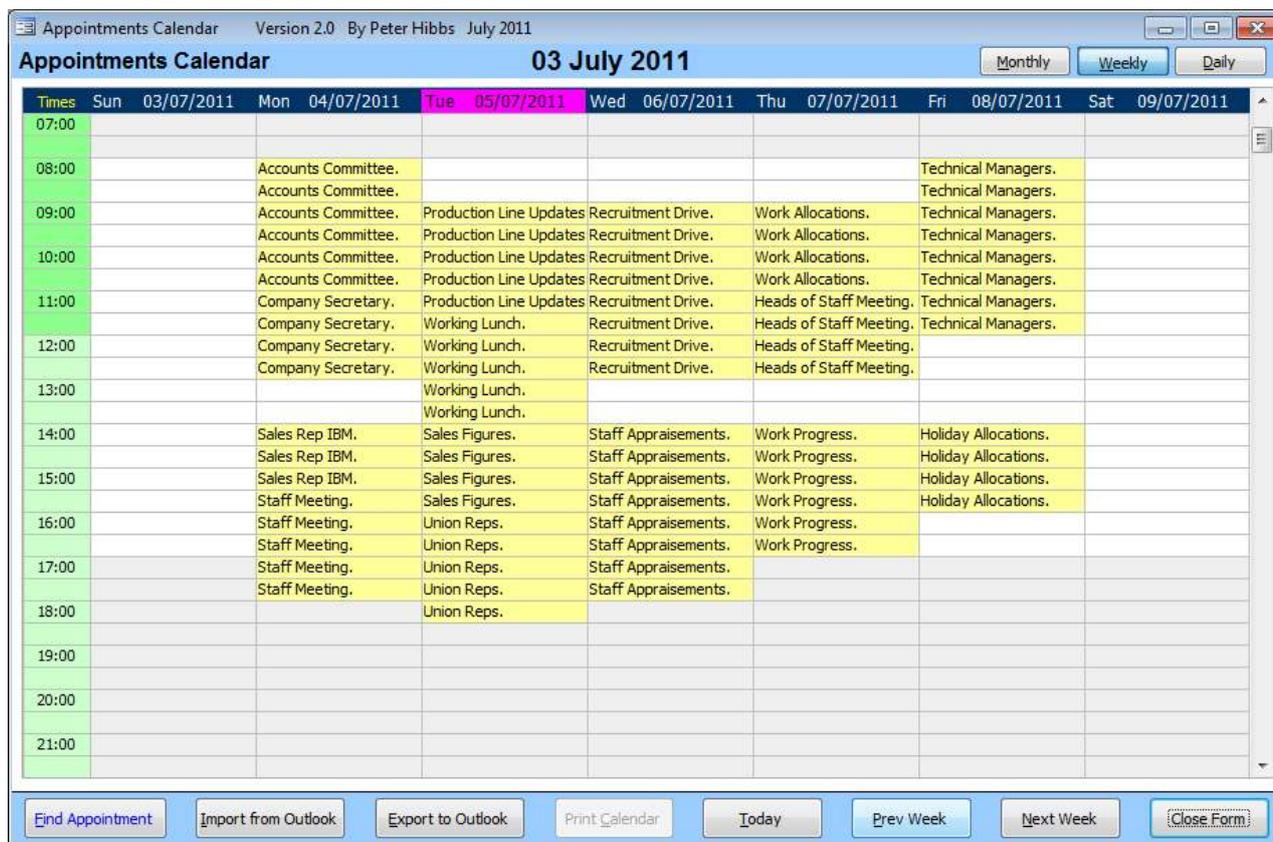
The **Monthly** calendar will now show the number of appointments in the morning and afternoon, something like this :-

AM = 3
PM = 4

This mode uses the query **qryMonthAppts** so if you want a different format, say smaller time groups or whatever, then you should change the query accordingly. Note that this query is only used for this facility so changing it should not affect any other facilities

Show Appointments that span Multiple Time Slots

In the demo database, if an appointment spans two or more time slots, the subject line is repeated for each time slot (unlike MS Outlook which draws boxes and lines over the time slots which is a bit difficult to do with Access). A typical example for the **Weekly** calendar is shown below.



However, where you might have long appointments like this, the calendar display can look a bit cluttered and a little difficult to follow. For this type of display you can set a flag to show the appointment subject in the first time slot and then show ditto marks in each subsequent time slot for the **Weekly** and **Daily** modes (it does not really work for **Monthly** mode).

To do this, change the constant **conDayDitto** from 0 to 1. Now the display should look something like below and the **Daily** calendar will look similar.

Note that if an appointment runs over more than one day, the subject line will be shown in the first time slot for each day (i.e. 00:00). The same group of appointments shown above will now look like this :-

Appointments Calendar Version 2.0 By Peter Hibbs July 2011

Appointments Calendar 03 July 2011 Monthly Weekly Daily

Times	Sun 03/07/2011	Mon 04/07/2011	Tue 05/07/2011	Wed 06/07/2011	Thu 07/07/2011	Fri 08/07/2011	Sat 09/07/2011
07:00							
08:00		Accounts Committee.				Technical Managers.	
09:00		"	Production Line Updates	Recruitment Drive.	Work Allocations.	"	
10:00		"	"	"	"	"	
11:00		Company Secretary.	"	"	Heads of Staff Meeting.	"	
12:00		"	Working Lunch.	"	"	"	
13:00		"	"	"	"		
14:00		Sales Rep IBM.	Sales Figures.	Staff Appraisements.	Work Progress.	Holiday Allocations.	
15:00		"	"	"	"	"	
16:00		Staff Meeting.	"	"	"	"	
17:00		"	Union Reps.	"	"		
18:00		"	"	"			
19:00							
20:00							
21:00							

Find Appointment Import from Outlook Export to Outlook Print Calendar Today Prev Week Next Week Close Form

This facility really only works properly where you have one appointment for each time slot. If you could have multiple appointments for the same time slot, the display would only show the ditto marks where appointments did not overlap and would probably look a bit of a mess.

As mentioned above, the code has not been included to do the same for **Monthly** mode because the appointments in this mode run horizontally on the calendar and you would almost certainly have more than one appointment in a day, so this system would not really work properly.

Open Appointment Schedule form after Find Appointment Operation

As mentioned above, you can use the **Find Appointment** button to find an appointment on the calendar. When the date and time of the appointment has been located the appropriate time slot is selected and highlighted in pink.

However, if they should want the **Appointment Schedule** form to be opened at this time so that they can display the appointment details and maybe edit the appointment you should change the constant **conOpenApptSchedule** from 0 to 1. Now, when the appointment date and time has been located, the **Appointment Schedule** form is opened immediately.

Multiple Calendars for Multiple Users

This demo calendar shows all appointments made regardless of who made them, i.e. just for the calendar user. However, in a company it may be necessary to have a separate appointment calendar for each employee, perhaps the company salesmen need their own appointment calendars or each doctor has a different calendar of appointments for their surgeries, etc.

This demo does not provide that option but it would not be too difficult to implement and could be done as follows :-

Firstly you would need to add a new Long Integer field to **tblAppointments** (call it **EmployeeID** or whatever) which would be used to identify the employee associated with the appointment record. This field would probably need to be linked to an equivalent field in a table that holds the employee information.

On the main calendar form you would add a combo or list box control which would be based on the employees table and would show a list of employees that are required to use the calendar. For a small list of employees a list box would probably be preferable because the user could then just click on an employee name in the list to show the appointments calendar for that employee. A combo box would take up less room on the form but the users would have to select an employee name from the drop down list each time they wanted to show the appointments for that employee.

In the three main sub-routines that process the appointment data (**ShowMonthAppts**, etc) you would add another in-line parameter called **vEmployeeID** (or whatever you prefer) which would hold the **ID** of the currently selected employee on the main form as specified in the combo or list box control. The sub-routine header would then look something like this -

```
Public Sub ShowMonthAppts (vMonthStart As Date, vEmployeeID As Long)
```

In the sub-routines, the Open Recordset command would then have this parameter added so that it returns only records where the employee ID matched the new **EmployeeID** field in the appointments table records, the SQL command would look like this :-

```
Set rst = CurrentDb.OpenRecordset("SELECT * FROM tblMonthData WHERE  
EmployeeID = " & vEmployeeID & " ORDER BY RowNo")
```

You would also have to modify the **Weekly** and **Daily** sub-routine calls in a similar way.

In the main form, whenever any of the three calendar routines are called, you would also add the **EmployeeID** parameter to the routine call so, for example, to update the **Monthly** calendar, the calling code would look something like this :-

```
ShowMonthAppts Me.txtDate, cboEmployees
```

Where **cboEmployees** is the name of the combo box control that holds the **EmployeeID** of the currently selected employee (or it may be a list box, of course). In the **After_Update** event of the combo or list box you would call the **CheckMode** sub-routine which would then display the calendar for the selected employee for the current calendar mode.

In addition, you would also need to pass the current **EmployeeID** number to the **Appointment Schedule** form whenever you want to create a new appointment so that the code in this form can set the **EmployeeID** field in the appointments table to the **EmployeeID** selected on the main form. And you would probably want to amend the **Find Appointment** code to allow the users to search for and display an appointment for any employee. There are so many different factors involved with something like that it is difficult to be more precise, how this sort of facility is implemented will obviously depend a great deal on what the users require.

Handling Single and/or Double Quotes in Text Fields

The contents of the various text fields used in the appointments table, i.e. **Subject**, **Location** and **Notes** are copied into the table using standard SQL code whenever an appointment is created or amended. However, if a single quotation character (') is entered into these fields by the user, the SQL code will error out.

To get around this, the method used in this demo is to declare a constant which is used in place of the single quote characters in the SQL strings. The constant is named **QUOTE** and is declared in the main code module, **modCalendarCode** like this :-

```
Public Const QUOTE = """"
```

In the SQL code, a typical command would look something like this :

```
, " & QUOTE & txtSubject & QUOTE & ", "
```

Note that there are other methods of fixing this particular problem which you may want to use instead.

Another similar problem is the use of double quotes (“”) in a text field and the method used to prevent this is to replace any double quote characters with two single quote characters before the data is written to the table. This is usually done in the **After Update** event of the text fields but may be also be done before the SQL code. To find those you can just search for the word **Replace** in all the code modules.

Color Coding Appointments

It is sometimes useful to be able to color code appointments, perhaps by category, status, etc. It is possible to do this (with some restrictions) by using the OLE type fields on a form. This demo database does not use this technique because it will vary considerably between applications so these notes will discuss some general ideas that can be used if this option is required.

A more detailed discussion of the methods used to add color coding to Continuous type forms can be found here :- <http://www.utteraccess.com/forum/Continuous-Form-Multiple-t1968354.html>

The screen shot below shows an example of how the **Monthly** calendar could look if various appointment categories are color coded. The same methods could also be used for the **Weekly** and **Daily** modes depending on the application requirements. Here is a brief description of the main changes that would need to be made to the current design to implement something like this. As there are so many different ways of adding color to a calendar, this is only a general discussion in order to show some of the techniques that can and can't be used.

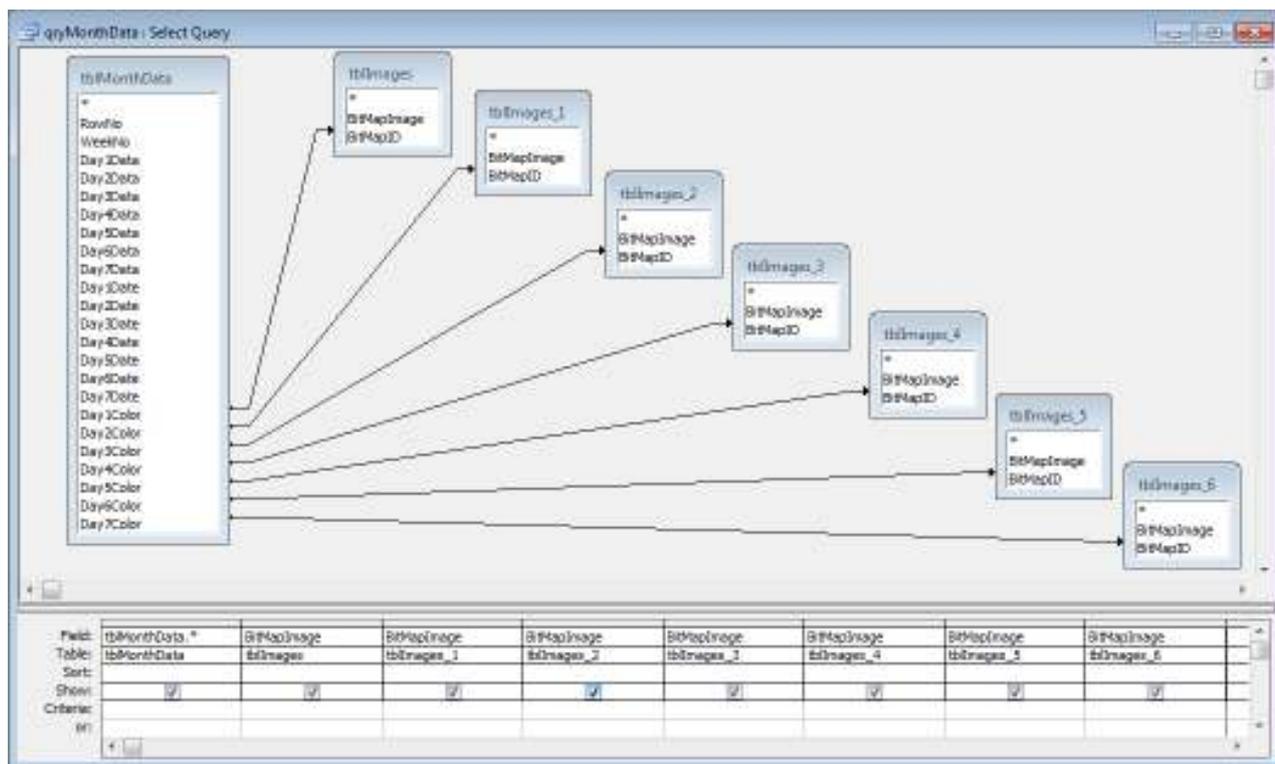


If you wanted to allocate a color category to each appointment, for example, then you would first need to add a **Category** field to table **tblAppointments** which would hold the category reference, this would normally be a Long Integer value linked to a Categories table which would hold other information about the category type (and maybe color information), etc. Obviously, whenever a new appointment record is created, the users would need to have some method to allocate a category value to the appointment, usually with a Combo box control or something similar.

In order to display a color for any day of the week, it is necessary to add seven more controls to the form **frmCalendarMonth** which will be **Bound Object Frame** types and should be placed behind the seven controls **Day1Data** to **Day7Data** which should then have their **Back Style** property changed to 'Transparent'. Also, in the table **tblMonthData**, seven new Long Integer fields (named

Day1Color to **Day7Color** in this example) should be added which will be set to the equivalent color code determined by the required category for that field.

In the demo database the form **frmCalendarMonth** is bound to the table **tblMonthData** directly because all the fields on the form are all in the table. However, in this case, the data from the table **tblMonthData** has to be combined with the bit map images from the table that holds the category colors (**tblImages** in this example) and so you will need the form bound to a query which returns the data from both tables. As there are seven separate color fields on the form (one for each day of the week) you would need the table that holds the color information repeated seven times, i.e. once for each field. The query would look something like this :-



What happens is this – the user would select a category for an appointment, let’s say there are 10 different category colors numbered 1 to 10, and the category number would be saved to the **Category** field in the appointment record when the appointment is created.

When that appointment is to be displayed on screen, the **Category** number would be copied into the temporary array in the **ShowMonthAppts** routine and then into the **Day(n)Color** field in table **tblMonthData** for whatever day the appointment is for (see code extract below for details on the extra code lines required).

Then the sub-form **frmCalendarMonth** sets the color of the appropriate day **BitMapImage** field to the required color which is based on the value of the **Day(n)Color** field which in turn is linked to the **BitMapID** field in the table **tblImages** as shown in the query above. The **BitMapID** field holds a numeric value which corresponds to the **Category** number in the appointments table and also the **Day1Color** to **Day7Color** values in the table **tblMonthData**. In this example, table **tblImages** would have 10 records with a different color image in each **BitMapImage** field and the values 1 to 10 in the **BitMapID** field for each record.

For the **Weekly** and **Daily** modes a similar system could be used to provide category colors on those sub-forms. The **Utter Access** reference mentioned above has more details on how to set up the **Bound Object Frames**, color editing and so on.

The changes below would be required to be made to the sub-routine **ShowMonthAppts** in order to add the color information to the table **tblMonthData** (code differences shown in red).

```
'Change Array size to allow for the 7 extra fields
Dim vArray(14 - 1, 5) As Variant

...

'Add the category value to the array
    vDate = vDate + 1
    vArray(vCol + 7, vRow) = Nz(rst!Category)
Loop Until vDate = vDateStop

...

'Add the 7 extra fields into the UPDATE command
For vRow = 0 To 5
    CurrentDb.Execute "UPDATE tblMonthData SET "
        & "Day1Data = " & QUOTE & vArray(0, vRow) & QUOTE & ", "
        & "Day2Data = " & QUOTE & vArray(1, vRow) & QUOTE & ", "
        & "Day3Data = " & QUOTE & vArray(2, vRow) & QUOTE & ", "
        & "Day4Data = " & QUOTE & vArray(3, vRow) & QUOTE & ", "
        & "Day5Data = " & QUOTE & vArray(4, vRow) & QUOTE & ", "
        & "Day6Data = " & QUOTE & vArray(5, vRow) & QUOTE & ", "
        & "Day7Data = " & QUOTE & vArray(6, vRow) & QUOTE & ", "
        & "Day1Color = " & Val(vArray(7, vRow)) & ", "
        & "Day2Color = " & Val(vArray(8, vRow)) & ", "
        & "Day3Color = " & Val(vArray(9, vRow)) & ", "
        & "Day4Color = " & Val(vArray(10, vRow)) & ", "
        & "Day5Color = " & Val(vArray(11, vRow)) & ", "
        & "Day6Color = " & Val(vArray(12, vRow)) & ", "
        & "Day7Color = " & Val(vArray(13, vRow)) & " "
        & "WHERE RowNo = " & vRow + 1
Next
```

There is one limitation when using this color system on a sub-form. On the three sub-forms some fields are colored light grey (using **Conditional Formatting**) to show date or time fields which are not normally used, for example – the last few dates in the previous month and the first few dates in the following month. The problem is that the data fields on the sub-forms have to have the **Back Style** property set to 'Transparent' in order for the color image behind the fields to show through. Since **Conditional Formatting** uses the **Back Color** property to show these fields in a different color, it is not possible to keep that facility when using colors to show different categories, etc.

Import Appointments from Microsoft Outlook

This demo calendar has facilities to import existing appointments from MS Outlook (and export to MS Outlook, see below) which can be used, if required. If this option is not required then just delete the buttons and code. This code is only a very basic set of routines, to use any of the more advanced features of MS Outlook you should Google the Internet for more detailed examples.

The main import/export routines for these facilities are all contained in the code module **modOutlookCode** and are called from the click events of the two buttons on the main form.

Prevent Duplicate Imports

The import facility uses sub-routine **ImportOutlookAppointments** which imports any new Outlook appointments starting from today. The import code checks the **Start Date** and **Subject** text of each imported Outlook appointment and if it matches the **Start Date** and **Subject** text of any appointment that is already in the appointments table, it assumes that the appointment has already been imported and ignores it. This procedure prevents the same appointments being added to the appointments table each time the routine is called. Of course, this method assumes that the **Start Date** and **Subject** form a unique record which may not be the case in your own application so you may need to amend this to cater for your specific situation.

Import Date Range

The code imports all appointments starting from the current day on the assumption that you would not normally want to import old appointments and it could also take a long time to import every appointment from many years ago. If, however, you need to change the import start date you can modify the code in the **ImportOutlookAppointments** routine as follows :-

In the line –

```
For Each ItemAppt In fdrCalendar.Items.Restrict("[Start] >= '" & Date & "'")
```

You can change the **Date** parameter to a specific date or if you want to import all appointments, you can remove the Restrict clause completely by removing all the code after the word **Items**.

Import All Day Events

Outlook has some facilities which have not been included in this demo database and some facilities are done in a slightly different way. For example, in Outlook there is an `AllDayEvent` flag which is not used in this version. If an appointment is imported that has this flag set, the code converts that into Start and End times which default to 08:00:00 and 17:00:00. If you are using this option you may want to amend those times to fit in with your own application.

Import Other Outlook Fields

Outlook has a number of fields which are not used in this demo database such as **Categories**, **Busy Status**, **Optional Attendees**, etc, etc. These are ignored by the import code but if you want to use some of them then you would need to add the appropriate fields to the appointments table and modify the code to copy the data to the table. Some of those fields are shown in the code but Remmed out and there may also be more fields in later versions of Outlook which are not shown.

Early or Late Binding

The code in this version is set up for **Late Binding** which means that, as there is no reference made to a specific version of the MS Outlook program, this database should work with any version of Outlook. The downside of this is that some useful facilities, such as 'Intellisense' when typing in Outlook specific code, do not work. If you are making a lot of changes to the design of these facilities it is probably advantageous to switch to **Early Binding** during the development stage and then revert to **Late Binding** when distributing the database to the user's PCs.

You can do this easily by Remming out the first four lines of code (the four Dim statements) in the routine and Un-Rem the next four lines. Also, you should set up a reference (in **Tools -> References**) to your version of the **Microsoft Outlook Object Library**. Don't forget to remove that reference when you switch back to Late Binding.

Export Appointments to Microsoft Outlook

You can also export any appointments made in the calendar to your MS Outlook calendar. The VBA code in function **ExportOutlookAppointments** is basically the same as the import procedure in reverse. There is one major difference though, in that appointments that have already been exported are flagged with a field in the appointments table.

The problem here is that if you try and use the same technique for testing for duplicate appointments in Outlook, the code would need to check every appointment in the Outlook calendar and compare the **Start Date** and **Subject** fields with each appointment record in the Access calendar to ensure that they were not the same. This would probably take a long long time to execute and so the method used here is to set the **Exported** flag in the appointments table whenever an appointment is exported. Then, when the operation is done again, any appointment records with the flag already set are ignored.

In order to give the users some flexibility with this option, the **Exported** flag is made available on the **Appointment Schedule** form so that they can reset the flag (with a Check box) if they need to export the appointment again. Of course, if the Outlook Import/Export facilities are not being used in your application it would probably be sensible to remove that Check box (or make it invisible) to avoid confusing the database users.

Set/Clear Appointment Reminder Flag

When an appointment is normally set up within Outlook, the 'Reminder' check box is automatically ticked by default and the Reminder time is set (depending on the current mode). In the demo database, this flag is reset so that the reminder timer is not activated. If you need this option to be active then you should change the **ItemAppt.ReminderSet = False** line to show **True** instead of **False** and you should un-Rem the next line and set up a value in minutes (which is currently set to 15 minutes). If you want the users to be able to change these settings, then you will need to add these fields to the appointments table and add some new controls to the **Appointment Schedule** form so that they can enter the relevant values.

The comments about **Early/Late Binding** and the other Outlook fields mentioned in the previous section also apply here.

As noted above, the Import/Export code provided here is just a basic system, if you need something more sophisticated such as providing the ability to delete or modify Outlook appointments, have the appointments reminder system active or to have the option to use any of the other available fields, etc then you should search the Internet which should provide the additional code required.

History

Version 1.0. January 2011.

Version 2.0. July 2011.

Minor bugs fixed and many other facilities added.

Import and Export appointments to MS Outlook facilities added.

Outlook style toggle buttons replaced Option group control on main form.

VBA code amended to allow the calendar to be configured to user's requirements.

This document included in demo zip file.

Version 3.0 September 2012

Bug fixed that caused code to error out when used with **SQL Server**. Problem was in two routines, (**UpdateAppointment** in **modCalendarCode** and **cmdSave_Click()** in **frmCalendarAppt**) where the code tried to write the **AppointmentID** value to the **ApptID** field which is an **AutoNumber** field. Problem fixed by removing that code.

A lot of code has been tidied up and simplified and several 'transparent' buttons used to eliminate a lot of messy code.

A number of Constants added to the main module code to allow various optional facilities to be switched on or off just by changing one constant value.

A pop up calendar added to some forms (courtesy of Allen Browne) to allow any date to be entered easily.

=== End of Notes ===