

Este proyecto lo había comenzado un par de años atrás luego lo abandoné, así que me decidí a retomarlo, si tendría que comenzar de cero hoy cambiaría muchas cosas ya que no son las más óptimas, pero esto no impide que el calendario funcione bien.

Ahora bien el control se trata de un calendario al cual podemos añadir eventos, esto puede servir para muchas cosas como ser mostrar vacaciones de personal, eventos de cobro, turnos, actividades de tiempo o mostrar de manera gráfica intervalos de fecha.

El control tiene soporte para hacer drag & drop de los eventos, tal como poder modificar con el mouse su rango de fecha, si presionamos la tecla Ctrl y hacemos drag & drop duplica este evento.

Luego la parte de entrada de datos queda a cargo del programador, en los ejemplos agregué unos formularios de muestra, lo que no hice fue crear una serie de eventos, pero es posible si se tiene un poco de paciencia, cualquier duda sobre esto último puedo dar una guía de cómo hacerlo.

Dentro de los ejemplos en la carpeta «Advance» encontrarán un ejemplo enlazando el calendario a una base de datos de SQLite, deberán agregar a la referencia a la dll J3cnn.dll

Los eventos del calendario no se manejan por índice sino por keys esto porque internamente éstos son ordenados por fecha y de esta forma se pierde el índice (esto no gustó mucho cómo lo encaré de un principio).

El control no tiene ninguna dependencia, inclusive los iconos son pintados con líneas.

La rueda del scrollbar funciona correctamente cuando está compilado, esto lo hice así ya que no utilicé Safe subclass, para evitarme dolores de cabeza al programar lo deje así, compilado funciona correctamente lo mismo para el evento MouseLeave. Si a alguien le es necesario que funcione en el ide puede quitar el «If App.LogMode Then».

Para ir cerrando sé que hay muchas cosas que quedaron pendientes, me gustaría que comenten si habría que implementar algo o corregir, también si ven algún error; así de esa manera me es más fácil. En los primeros días de subida seguramente lo esté actualizando frecuentemente a medida que me informen o encuentren detalles.

---

A continuación un detalle rápido de sus funciones y propiedades:

**AddEvents:** Función principal para agregar eventos al calendario, parámetros requerido; sujeto (titulo descriptivo), fecha y hora de inicio, fecha y hora de finalización y color del evento, el resto de los parámetros son opcionales, valor de retorno una key del evento.

**CenterCalenarInNow:** Mueve el scroll a la hora actual.

**Clear:** Elimina todo los eventos.

**DateValue:** Asigna o retorna la fecha actual del calendario.

**DayHaveEvents:** Retorna True/False si hay eventos en un día específico.

**DropDownColor:** Cuando hay muchos eventos en un día y en el modo de vista Mes hay más de los que se pueden mostrar, se muestra una barra desplegable la cual podemos cambiar el color con esta propiedad.

**EventsCount:** Cantidad de eventos agregados.

**EventsRoundCorner:** Propiedad booleana para mostrar o no esquinas redondeadas en

eventos y botones.

**FirstDayOfWeek:** Aquí podemos asignar que día queremos que se muestre como primer día de la semana, por defecto usa el del sistema.

**GetAllEvents:** Obtiene una colección de las keys de los eventos agregados.

**GetEventData:** Obtiene los datos de un evento, su primer parámetro es la key del evento la cual podemos obtenerla con GetAllEvents o por algún evento, el resto de los parámetros son valores de retorno.

**GetEventsFromDay:** Retorna una colección de keys de los eventos de un día en específico,

**GetSelectionRangeDate:** Función para obtener el rango de fechas seleccionada.

**HeaderColor:** Color de la cabecera y parte de la temática.

**HideEvent:** Oculta el evento, útil para filtrar.

**LinesColor:** Color de las líneas.

**Redraw:** Habilita o deshabilita el repintado del calendario, esto sirve para acelerar la carga de eventos.

**RemoveEvent:** Elimina un evento.

**Refresh:** Refresca el repintado calendario.

**SelectedEvent:** Retorna el evento seleccionado.

**SelectionColor:** Color de la selección.

**SetStrLanguage:** Aquí podemos pasar la traducción de las palabras utilizadas.

**ShowToolTipEvents:** Si se quiere mostrar o no la ventana tooltip, esta se puede remplazar por otra personalizada y con información más detallada. Véase los eventos EventMouseEnter y EventMouseLeave.

**Update:** Es más completo que refresh, este vuelve a reordenar por fecha y alfabéticamente los evento, recalcula la posición y por último repinta todo.

**UpdateEventData:** Función para actualizar los datos de un evento. Debe pasarse la key del evento que queremos modificar.

**UserCanChangeDate:** Habilita o deshabilita para que el usuario pueda cambiar la pagina actual del calendario.

**UserCanChangeEvents:** Habilita o deshabilita si el usuario puede cambiar los eventos (mediante estiramiento o arrastre).

**UserCanChangeViewMode:** Oculta todos los botones en la parte superior de la parte derecha. De esta forma el usuario no puede cambiar el modo de vista o bien el programador toma el control de que vista quiere mostrar.

**UserCanScrollMonth:** En el modo de vista mes, se puede scrollear infinitamente si necesidad de cambiar de página, esto sólo si se habilita esta opción.

**ViewMode:** Cambia por código en el modo de vista (Día, Semana, Mes, Año)

## Eventos:

**DateBackColor:** Al dispararse este evento podemos elegir según la fecha u hora el color de fondo de la celda, así como también su HatchStyle.

**DateChange:** Cuando el usuario cambia de página el calendario se dispara este evento.

**DragNewEvent:** Cuando el usuario tiene la presionada la tecla Control y hace drag & drop el evento se duplica, o sea un nuevo evento es agregado por el usuario, con este evento somos informados y así podremos guardar en la base de datos o como se lo esté manejando.

**DropDownViewMore:** Evento cuando se presiona el botón de ver más, este botón se encuentra en la vista «Meses», aparece cuando hay muchos eventos en un día y no entran en la celda, al presionar este cambiará a modo de vista = «Día», esto podemos cancelarlo y mostrar una ventana personalizada con los eventos del día, los eventos del día los podemos recuperar con GetEventsFromDay.

**EventChangeDate:** Este evento se dispara cuando el usuario hace drag & drop o cambia el tamaño del evento con el mouse. es útil para almacenar en la base de datos las nuevas fechas.

**EventMouseEnter:** Cuando el mouse entra sobre un evento.

**EventMouseLeave:** Cuando el mouse sale por encima del evento.

**EventClick:** Click en un evento.

**PreDateChange:** Antes de que el usuario pague hacia otra fecha, aquí podemos cancelarlo.

**PreEventChangeDate:** Antes que el usuario cambie los datos de un evento con del drag & drop o el mouse, este evento es muy útil para impedir arrastrar eventos en ciertas fechas u horas en los que no queremos que se agregue un evento.