

EDICION DE RECURSOS DESDE RESOURCE WORKSHOP

Que es y para que sirve Resource Workshop.

Resource Workshop es un editor de recursos, es decir, un programa que nos permite crear y gestionar recursos para después utilizarlos desde un programa escrito en diversos lenguajes de programación para Windows como por ejemplo FiveWin.

Esto no nos dirá mucho si no decimos primero que es un recurso.

En Windows estamos acostumbrados a manejar programas que están todos ellos compuestos de elementos comunes de trabajo como pueden ser las ventanas con sus menús y barras de botones, los cuadros de diálogo que son como ventanas pero que aparecen dentro de la ventana del programa principal para acciones específicas del programa como por ejemplo buscar una palabra, otros elementos que usamos son los textos que aparecen dentro de las ventanas (SAY de FiveWin), campos de edición que son cajas donde podemos escribir datos dentro de las ventanas (GET de FiveWin), botones con texto o con gráficos donde podemos pulsar para aceptar, cancelar u otras cosas, pues bien, cada uno de estos elementos ventanas, diálogos, textos, cuadros de edición, botones y muchos otros más que manejamos a diario son recursos de los programas.

Un editor de recursos será por lo tanto el programa que nos permite diseñar, modificar y mantener todas las ventanas o cuadros de diálogo del programa que estemos desarrollando de una forma gráfica, visual, muy rápida y muy sencilla.

Resource Workshop es un programa de la casa BORLAND ahora rebautizada como INPRISE y viene incorporado en los programas de desarrollo de esta casa como el Borland C++ y en Turbo Pascal para Windows, aunque también puede adquirirse de forma independiente.

Por lo tanto trabajando con Resource Workshop nuestro trabajo a la hora de crear o mantener un programa se divide en dos partes una la de diseñar y mantener las pantallas de la aplicación con Resource Workshop y la segunda la de desarrollar un programa que trabaje sobre estas ventanas y haga finalmente lo que nosotros o nuestro cliente quiera que haga con FiveWin.

El diseño visual por lo tanto es cosa de Resource Workshop y por lo tanto la mitad del trabajo de un nuevo programa para Windows es cuestión de Resource Workshop, es decir que “casi la mitad” del trabajo será simplemente dibujar unas bonitas, ordenadas y eficaces ventanas que son lo que al final el usuario va a ver olvidándose de que detrás haya un programa que nos haya costado más sudores de los que pueda imaginar.

Por que usar Resource Workshop

En FiveWin podemos desarrollar todo el programa directamente desde el código fuente diseñando todas las ventanas, botones y demás elementos directamente a mano sin usar para nada un editor de recursos como Resource Workshop, entonces por que usar este editor.

Hay varias y poderosas razones para usarlo.

En primer lugar el trabajo y el esfuerzo de imaginación necesario para diseñar un simple cuadro de diálogo o ventana sin usar un editor, con media docena de textos escritos en ella (SAY) y otros tantos cuadros de texto (GET) mas un par de botones, sin ver físicamente la ventana hasta el final, sin poder dibujarla con el ratón, o sea haciéndola completamente de memoria, es un esfuerzo inimaginable y no hablemos de un ejemplo mucho más sofisticado con Folders o Tabs por ejemplo. Y lo digo con conocimiento de causa ya que tarde más de 2 años en empezar a utilizar Resource Workshop y sude muchas ventanas que ahora diseñaría en un abrir y cerrar de ojos con este programa.

El resultado de diseñar las ventanas con Resource Workshop es que es mucho más cómodo e intuitivo dibujar la ventana con sus elementos de forma VISUAL con el ratón sobre la pantalla sin ningún esfuerzo mental de imaginación ya que vemos la ventana según la estamos haciendo tal y como quedará después al ser usada desde el programa, quedando además mucho más ordenada, con un aspecto más profesional y más elegante ya que todos los elementos pueden ajustar en anchura, altura, posición, etc...

Al diseñar las ventanas con Resource Workshop además obtenemos como resultado que el fuente que escribiremos posteriormente en nuestra aplicación para manejar la ventana será mucho más contenido que diseñando la ventana dentro del mismo fuente sin usar el editor de recursos, quedando de esta manera más ordenado, más fácil de leer revisar y modificar y consiguiendo por lo tanto un ejecutable final (.EXE) más pequeño y en teoría más rápido.

OJO solo en teoría porque si bien es cierto que el ejecutable se hace más liviano por otro lado aparece el recurso que estamos editando que es otro archivo (nombre_del_recurso.DLL) que se debe de adjuntar al ejecutable y que también Windows deberá de cargar en memoria para que todo funcione en conjunto. En definitiva que ahorrar tamaño en memoria y espacio en el disco no nos ahorramos prácticamente nada.

Otra de las ventajas es que podemos introducir en un recurso todos los elementos gráficos, e incluso sonidos de un programa de forma que queden agrupados en un único archivo DLL de forma que no queden desperdigados por el disco los iconos, los gráficos BMP, los sonidos WAV y demás de forma que cualquier usuario listillo pueda meterles mano y liarnos un buen problema, siendo además más sencillo distribuir la aplicación final, un único o varios archivos de recursos contienen todo lo necesario para la aplicación.

También es muy bueno destacar que una vez de realizada una ventana o cuadro de diálogo en Resource Workshop podemos perfectamente copiarla y pegarla total o parcialmente para crear otra ventana o diálogo dentro del mismo grupo de recursos o en otro cualquiera, es decir dentro del mismo programa o para otro cualquiera, para después modificar el recurso copiado si es necesario. Por ejemplo ¿en qué se parece una ventana de gestión de clientes de un programa de ventas a una ventana de gestión de clientes de un centro deportivo? Si no en el 100% sí en un 95%.

Proyectos en Resource Workshop, como crearlos y gestionarlos

Resource Workshop nos permite gestionar una gran cantidad de recursos o controles de Windows para construir con ellos las ventanas de nuestra aplicación.

Todos los recursos que almacenamos juntos para usarlos en una aplicación se agrupan en uno o varios proyectos de Resource Workshop.

Un proyecto será en realidad un fichero que aglutinará un conjunto de recursos de varios tipos, siendo el límite de este conjunto de 255 recursos por proyecto. En él vamos a introducir nuestras cajas de diálogo, iconos, bitmaps, etc. Tenemos varios tipos de proyectos de recursos que identificaremos por la extensión del fichero que utilicemos. Principalmente utilizaremos dos de ellos, el tipo **DLL** (Librería de Enlace Dinámico) y el segundo de tipo **RC** (Recursos Compilado).

En que se distinguen ambos y cual conviene utilizar.

DLL (Librería de Enlace Dinámico)

DLL es un fichero separado del ejecutable principal EXE que contiene módulos en formato binario y que funciona de la siguiente manera.

Cuando nuestro programa necesita uno de los recursos del archivo DLL es entonces cuando lo carga a memoria y lo muestra pero esto puede hacer que parezca lento por la continua carga de recursos, en cambio la gran ventaja de este sistema es que se puede perfectamente modificar el DLL sin modificar para nada el EXE y seguirá funcionando perfectamente.

Es por eso que es mucho más cómodo sobre todo a la hora de desarrollar el programa ya que podemos cambiar el recurso en pantalla, por ejemplo modificar los datos de una ventana, añadirle un texto, cambiar sus botones, a continuación grabar el proyecto y ejecutar de nuevo el programa .EXE viendo el funcionamiento de las modificaciones al instante y en realidad la carga de recursos según van haciendo falta no es tan lento como pueda parecer y es así como funcionan en realidad la mayor parte de los programas de Windows sino solo hace falta ver la gran cantidad de DLL y de OCX (versión más moderna de las DLL) que hay en cualquier programa comercial de Windows.

Los ficheros DLL que usará nuestro programa se especificarán al principio del código del programa mediante la fórmula de FiveWin `SET RESOURCES TO "nombre_del_archivo.DLL"`, aunque de esto hablaremos más adelante en el capítulo "Como usar los recursos desde FiveWin".

Antes de finalizar el programa tendremos que poner `SET RESOURCES TO` para liberar la DLL. Si no lo hacemos, se quedará 'pillada' y no podremos grabar las modificaciones que hagamos hasta que no se cierre Windows o se descargue la DLL de la memoria con algún programa especial que entre en memoria lo localice y lo descargue, lo cual a veces es necesario si nuestro programa no ha llegado al final por un error durante el desarrollo del mismo.

Para poder descargar estas DLL hay muchos programas de tipo freeware en Internet que muestran todo el contenido de la memoria y localizando la DLL por su nombre la descargan, en mi caso utilizo uno llamado WPS.EXE que funciona perfectamente y lo conseguí de esta manera.

RC (Recursos Compilado)

Los ficheros RC pueden parecer a ojos de un programador algo más normales que los DLL.

Si miramos uno por dentro veremos que esta escrito en lenguaje natural, es en realidad un programa y se puede leer. Resource WorkShop guarda en el RC todos los recursos por su nombre indicando las coordenadas, tamaños y otras características de cada recurso.

Posteriormente se utiliza el programa RC.EXE también distribuido junto con los productos de Borland como el C++ para transformarlo en lenguaje máquina. Con este programa introduciremos los recursos en nuestro fichero EXE sumándoselos al final de este, es decir incluimos el programa con los recursos dentro de nuestro programa EXE no siendo necesario por lo tanto distribuir el fichero RC para el funcionamiento de nuestro programa.

Se hace de la siguiente manera desde la línea de comandos:

```
RC -k nombre_del_fichero.RC nombre_del_fichero.EXE
```

El modificador -k se utiliza para que el programa RC no reorganice los segmentos del ejecutable. Se añaden todos los recursos al final del ejecutable y en la cabecera se anota los recursos incluidos. El nombre del fichero .RC ha de ser el mismo que el de nuestro ejecutable pero de extensión RC.

Pero el uso de los RC ralentiza mucho el proceso de redacción y de pruebas de un programa ya que se necesita compilar, enlazar y montar el RC dentro del EXE cada vez que queramos probarlo y el proceso es algo mas lento.

En cambio la ventaja de incorporar los recursos dentro del ejecutable es que no se podrá entrar en ellos ni siquiera con Resource Workshop con lo cual ni el cliente ni otro programador podrá en principio alterarlos y además se cargan junto con el EXE en memoria al ejecutar el programa y por lo tanto no se están cargando continuamente, aunque lógicamente consumen más recursos de la máquina.

Una formula intermedia si uno no se decide por el uso de uno de estos dos sistemas podría ser la siguiente, se pueden utilizar DLL mientras se desarrolla el programa y después una vez terminado y para entregarlo grabar la DLL como un RC desde el propio Resource Workshop y adjuntar la RC al ejecutable, prescindiendo por lo tanto de entregar la DLL al usuario.

A mi juicio y creo que en esto coinciden la mayor parte de los programadores, es más cómodo, simple y rápido usar recursos de tipo DLL, ya que es muy difícil que alguien tenga un editor de recursos y vaya a entrar en ellos, lo cual también se puede evitar comprimiendo la DLL con un programa llamado SHRINKER de la casa BlinkInc que es la que programa también el archiconocido e imprescindible enlazador BLINKER

Que haremos y qué no haremos con el Resource WorkShop.

Resource Workshop es un programa que sirve para crear muchos tipos de recursos pero todos queremos que el trabajo sea lo más fácil posible. Lo que no haremos, por supuesto, es utilizar herramientas por muy visuales que sean, para realizar un trabajo que lo haríamos mucho mejor desde el código de nuestro programa. Esto es lo que sucede con algunos recursos como los menús. Los menús son mucho más fáciles de codificar desde el programa FiveWin que de hacer desde Resource Workshop. En cambio las cajas de diálogo son mucho más fáciles de diseñar con el editor de recursos que si las codificásemos desde programa. Esto sucede por que las coordenadas de las cajas de diálogo van en función del tipo de letra que estemos utilizando en la misma. Es un jaleo el situar los controles desde código, el diseño visual es ideal en este caso.

Los bitmaps los podemos almacenar tanto en ficheros BMP independientes como dentro de nuestros proyectos de Resource Workshop. Los iconos y cursores los manejaremos desde el editor de recursos. En todo caso, cada uno realizará los trabajos de una forma u otra, pero siempre tendremos que intentar hacerlo de la forma que resulte más fácil.

¿Qué haremos desde Resource WorkShop?

Desde el editor de recursos almacenaremos dentro de un proyecto los siguientes tipos de recursos

- Cajas de diálogo o ventanas (DIALOGS)
- Imágenes fijas dentro de la ventana (BITMAPS)
- Iconos (ICONS)
- Cursores (CURSORS)
- Textos sobre los diálogos y ventanas (SAY)
- Cajas de texto para escribir en ellas desde el programa (GET)
- Listas de selección (LISTBOX)
- Listas de selección combinadas para seleccionar o teclear en ellas (COMBOBOX)
- Selectores de tipo radio, redondos y como sabemos excluyentes (RADIOBUTTON)
- Selectores de tipo chequeo, cuadrados y como sabemos no excluyentes (CHECKBUTTON)
- Botones con un texto sobre los que pulsar (BUTTON)
- Botones con un gráfico sobre los que pulsar (BTNBMP)
- Distintas pestañas dentro de una misma ventana (FOLDERS) o con páginas (PAGES)
- Distintas pestañas de selección en la parte inferior de una misma ventana (TABS)
- Browsers o tablas de datos presentadas en columnas (TCBROWSE & TWBROWSE)
- Barras de progreso (METERS)
- Selectores de una cantidad con forma de volumen de sonido (SELECTOR & SLIDER)
- Líneas separadoras decorativas (GROUP & SPLITTER)

Se pueden realizar más cosas con el editor pero realmente lo que más vamos a utilizar son los controles indicados y su uso y aplicación serán detallados un poco más tarde.

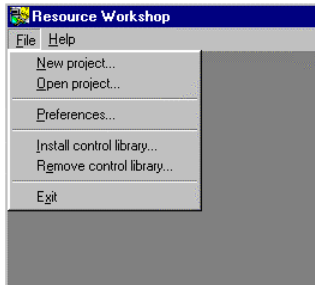
¿Qué no haremos desde Resource WorkShop?

- La ventana principal de la aplicación. Todas las demás ventanas del programa podrán ser recursos pero la ventana principal del programa ha de ser forzosamente escrita en FiveWin.
- Los menús de las ventanas y las teclas aceleradoras de atajo a las opciones de los menús que queramos.
- Las barras de botones de las ventanas.

Estos elementos resultan más fáciles y cómodos escritos en código FiveWin que diseñados como recursos de Resource Workshop.

Comenzando a trabajar con Resource Workshop.

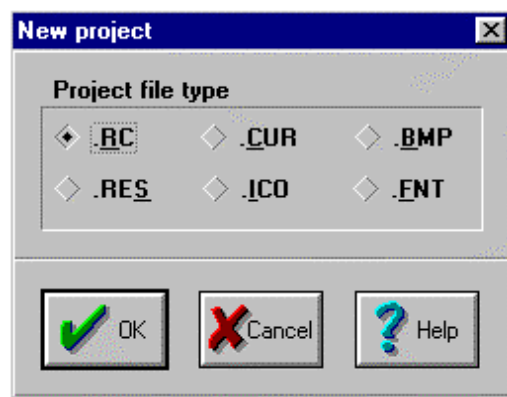
Creando un proyecto nuevo.



Para comenzar a trabajar tenemos que crear nuevo proyecto y es aquí donde no encontramos un primer problema, aunque de poca importancia.

La primera vez que se usa Resource Workshop se necesita crear un nuevo proyecto para ello se entra en el primer menú *File/New Project* o como veremos después abrir uno existente *File/Open Project*, una vez de creado o abierto un proyecto se muestra en la ventana principal que llamaremos de gestión de proyectos.

Las opciones para crear un proyecto nuevo sirven para seleccionar un tipo de archivo para crear un proyecto de entre los siguientes tipos:



.RC para grabar el proyecto en un archivo script de recursos, un fichero de texto que puede contener cualquier tipo de recurso.

.RES para grabar el proyecto en un archivo compilado de recursos, un fichero de texto que puede contener un conjunto de recursos compilados.

.CUR para crear un archivo de tipo cursor (.CUR). Un archivo cursor contiene una simple imagen de un cursor.

.ICO para crear un archivo de tipo icono (.ICO). Un archivo icono contiene una o varias simples imágenes de unos iconos.

.BMP para crear un archivo de tipo bitmap (.BMP). Un archivo bitmap contiene una simple imagen de tipo bitmap.

.FNT para crear un archivo de tipo fuente (.FNT). Un archivo de fuente contiene uno o mas tipos de letra diseñados por el usuario.

De todos estos tipos el único que puede que usemos será el RC ya que hacer un proyecto para meter un simple BMP o un ICO es una perdida de tiempo.

Pero para nuestra sorpresa y este es el problema al que hacíamos referencia, es que entre ellos no hay un proyecto del tipo DLL y entonces como haremos para poder trabajar con una DLL, pues es bien sencillo en vez de comenzar un proyecto nuevo lo que haremos es abrir una DLL mediante la opción de *File/Open Project* y trabajar sobre ella, ¿pero que DLL podemos abrir si todavía no tenemos creada ninguna?.

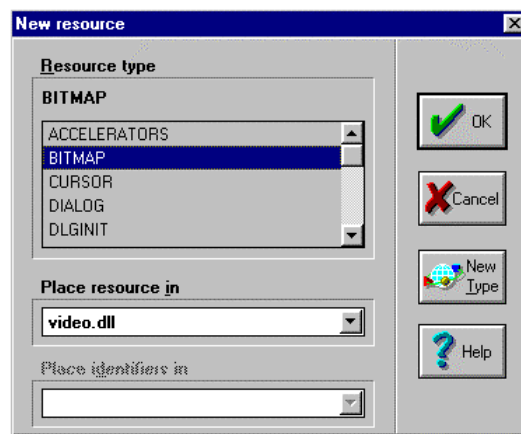
FiveWin soluciona este problema aportando una DLL vacía que se encuentra en la subcarpeta DLL de la carpeta de FiveWin que se tenga instalada en el disco y que se llama SCREENS.DLL, lo único que tenemos que hacer es copiar esta DLL u otra cualquiera que queramos a la carpeta donde vamos a desarrollar nuestro programa, renombrarla como nos parezca y después abrirla con Resource Workshop desde la opción *File/Open project* y no usar para nada la opción *New/Project*.

Llegados a este punto ya tendremos un nuevo proyecto vacío en la ventana principal de Resource Workshop que es la ventana llamada de gestión del proyecto y podremos empezar a incluir en nuestro proyecto los

recursos que necesitemos para nuestro programa, veremos cada uno de ellos por separado, para que se utilicen, como crearlos desde Resource Workshop y más tarde como utilizarlos desde FiveWin.

Incluyendo un recurso nuevo en un proyecto.

Para incluir un nuevo recurso dentro de un proyecto ya abierto hay que seleccionar la opción *Resource/New* para obtener la caja de diálogo siguiente:



Los siguientes pasos dependen del tipo de recurso que se vaya a crear:

Si queremos crear un ACCELERATOR, un MENU, un DIALOG, una STRINGTABLE, un RCDATA o un VERSIONINFO, Resource Workshop pondrá una entrada para el recurso en la ventana de proyectos y abrirá el editor apropiado para poder crearlo.

Si queremos crear un recurso de tipo imagen ICON, CURSOR, BITMAP, o FUENTE, Resource Workshop abrirá el diálogo Storage Format. Para incluir el recurso nuevo hay que hacer clic en el botón Source.

Si el tipo de recurso que se quiere crear no está en la lista hay que hacer clic en el botón New Type para crear un nuevo recurso definido por el usuario, veremos un ejemplo más adelante para incluir un recurso de tipo sonido WAV que no está entre los propios de Resource Workshop.

Añadiendo un Recurso existente a un Proyecto.

Normalmente habrá que usar dentro de nuestra aplicación Bitmaps e Iconos que podremos tomar del disco por que ya están hechos y no tendremos que crearlos como recursos nuevos.

En estos casos se puede incorporar al proyecto un recurso desde un archivo externo al proyecto abierto usando la opción *File/Add to Project* con lo cual nos mostrará el diálogo correspondiente donde seleccionando primero el tipo de recurso a añadir (BMP, ICO, CUR, etc.) podemos seleccionar el nombre del fichero que contiene el recurso o buscarlo por el disco antes de seleccionarlo.

Una vez seleccionado aparecerá junto con el resto de los recursos en la ventana principal de proyecto con un nombre genérico que se podrá cambiar seleccionándolo con el clic derecho del ratón y pulsando sobre *Rename*, para después pasar a editar este recurso pulsando doble clic sobre él si fuera necesario

Si lo que se desea es copiar un recurso existente en otro proyecto lo mejor es abrir dos instancias del programa Resource Workshop y con ambos proyectos abiertos, uno en cada instancia del programa, copiar de uno seleccionando el recurso y con la opción *Edit/Copy*, pasar desde la barra de tareas a la otra instancia

y pegar en esta con la opción *Edit/Paste* de forma que conservará su nombre y aparecerá junto con los demás recursos dentro de la ventana principal del proyecto y en su categoría correspondiente.

Abrir un proyecto existente.

Para abrir un proyecto existente lo haremos mediante la opción *File/Open project* de forma que se nos mostrará primeramente que tipo de proyecto deseamos abrir RC, RES o DLL (aquí si que aparece la opción de abrir DLL) y a continuación buscaremos por el disco la carpeta donde se encuentre el proyecto y haremos doble clic sobre su nombre de forma que aparecerá directamente sobre la ventana de proyectos. Si tuviéramos uno abierto con anterioridad habría que grabarlo y cerrarlo, lo cual en caso de olvido nos será recomendado automáticamente por el programa. La única manera de tener a la vez abiertos dos proyectos es como hemos dicho anteriormente abriendo dos veces el programa Resource Workshop y cargando un proyecto distinto en cada uno de ellos.

También es conveniente destacar que la mayor parte de las DLL de los programas comerciales que tengamos instalados en el disco y los propios de Windows pueden ser abiertos con Resource Workshop, de forma que podemos indagar en sus recursos, modificarlos (cosa poco recomendable aunque posible) e incluso copiarlos. En cualquier caso si probamos a abrir una DLL de otro programa y no es posible por cualquier motivo (normalmente por ser muy grandes y entonces hará falta otro editor de recursos como MS AppStudio o el Resource editor de Borland C++ profesional) se nos muestra un error indicando que es imposible abrirlo, pero no sucede nada, no hay peligro de dañarla.

Grabar un proyecto.

Como en todo programa de Windows es una buena idea grabar el proyecto a menudo. Resource Workshop nos propone una gran variedad de formas grabar nuestro proyecto de forma que podemos elegir exactamente que y como queremos grabar.

Entrando en el menú File con un proyecto abierto podemos encontrarnos con las siguientes opciones:

File/Save Project graba todo el proyecto abierto. Esta es la opción que habitualmente usaremos. Si lo que se graba es un proyecto recién creado que no tiene un nombre automáticamente nos llevará a la opción *File/Save As* la cual nos permitirá poner un nombre al nuevo proyecto pudiendo especificar en que carpeta del disco queremos que se grabe, lógicamente aquella en la que estemos desarrollando nuestro programa para que este pueda utilizar directamente los recursos.

Al grabar un proyecto Resource Workshop siempre se graban todos los recursos y sus referencias ID de forma que ya están directamente disponibles por el programa, cuando estamos desarrollando el programa podemos grabar el proyecto y sin tener que cerrar el Resource Workshop podemos probar directamente el resultado desde nuestra aplicación, incluso podremos modificar los recursos con nuestra aplicación en marcha pero solo podremos ver las modificaciones si grabamos de nuevo el proyecto para lo cual hemos de cerrar nuestra aplicación primero, ya que ésta abre el fichero de recursos de forma exclusiva impidiendo que se grabe mientras se estén utilizando.

Resource Workshop puede también grabar un archivo RES cada vez que se grabe el RC o DLL utilizado si se activan las correspondientes opciones de Multi-save en el diálogo de Preferencias del programa.

Si el proyecto es un archivo de tipo RC, Resource Workshop compila siempre este script y guarda una versión compilada en un archivo con el mismo nombre pero con extensión RWS, la próxima vez que se abra este proyecto Resource Workshop lo abrirá mucho más rápido utilizando la versión compilada RWS en vez del RC.

File/Save File As permite grabar un proyecto nuevo dándole un nombre, como ya hemos dicho, pero también permite duplicar el proyecto abierto dándole otro nombre o usando el mismo en una carpeta distinta.

Uso de la ventana principal de gestión de Proyectos.

La ventana de proyecto previsualiza una lista de todos los recursos incorporados al proyecto, agrupados por cada tipo de recurso (Icons, Bitmaps, Dialogs,...) de forma que sea fácil localizarlos en orden alfabético.

Puede modificarse la previsualización de los recursos de un proyecto con estas opciones.

- *View / Show Horizontal Preview*: muestra la lista de recursos del proyecto y su previsualización lado a lado en la ventana de control de proyectos. Es la opción por defecto.
- *View / Show Vertical Preview*: muestra la lista de recursos del proyecto y su previsualización arriba y abajo de la ventana de control de proyectos.
- *View / Hide Preview*: desactiva la visualización de la previsualización de los recursos o la vuelve a activar.

La línea de estado en la parte inferior de la ventana esta dividida en dos partes. A la izquierda muestra información sobre las opciones de los menús y las herramientas del programa. La parte derecha solo estará activa si se esta editando un diálogo o un bitmap y nos mostrará información sobre el elemento editado y sobre la herramienta que estemos utilizando sobre el.

Trabajando con los recursos.

Los recursos que vayamos creando o añadiendo a nuestro proyecto se irán agrupando dentro de la ventana principal del gestor de proyectos organizándose automáticamente por tipos de recurso y dentro de cada tipo en orden alfabético. Dos recursos no pueden tener el mismo nombre a no ser que se trate de recursos de distinto tipo ya que este nombre se usará desde el programa para llamar al recurso y este debe de tener un nombre único dentro de su tipo.

Haciendo doble clic sobre un recurso cualquiera, Resource Workshop arranca automáticamente el editor apropiado para poder modificarlo si es uno de los recursos propios de Resource Workshop, si es un recurso externo añadido por el usuario arrancará su editor de textos interno

Pulsando con el botón derecho del ratón sobre cualquiera de los recursos de la lista obtendremos un menú contextual que nos permite hacer acciones directas sobre ese recurso, siendo estas las siguientes:

- **Edit**: Pone en marcha el editor asociado al recurso para poder modificarlo, lo mismo que si se pulsa doble clic sobre el recurso.
- **Edit as Text**: Muestra el código fuente del recurso que es lo que en realidad se almacena en el proyecto, rara vez habrá que editar este código a mano pero veremos algún caso en el que si será necesario y como hacerlo.
- **View**: Nos visualiza el recurso al igual que Edit pero sin dejarnos modificarlo.
- **Rename**: Permite cambiar el nombre a un recurso, ojo el nombre es muy importante puesto que al programar desde FiveWin tenemos que llamar a los recursos por su nombre y no puede haber dos recursos que tengan el mismo nombre dentro del mismo tipo de recurso, sí en otro tipo distinto de recurso, por ejemplo un Bitmap y un Dialog pueden tener el mismo por que a la hora de llamarlos desde el programa se indica el tipo de recurso que vamos a usar y después el nombre.
- **Memory Options**: Las opciones de memoria indican como se carga y maneja el recurso en memoria. Estas opciones no es necesario modificarlas pues cada recurso tiene las necesarias para su correcto funcionamiento. Las opciones son las siguientes cuatro:

- Load On Call: Si está activado el recurso se carga en memoria solo cuando se necesita, de esta manera el programa arranca más rápido y ocupa menos memoria pero tiene que hacer más accesos a disco cuando se está trabajando y se llama a un recurso aunque esta es la forma normal de trabajo de casi todos los programas Windows que habitualmente manejamos. Si se desactiva el recurso se carga al llamar a la DLL al principio del programa y queda permanentemente cargado.
 - Moveable: Activado permite que el recurso una vez cargado en memoria pueda ser movido de un segmento a otro por Windows. Desactivado el recurso ocupará un lugar fijo en la memoria.
 - Discardable: Activado permite descargar el recurso de la memoria cuando no va a ser necesitado de momento como por ejemplo al cerrar una ventana y se volverá a cargar cuando haga falta. Desactivado el recurso permanecerá cargado desde la primera vez que se cargue hasta que se salga del programa o se descargue expresamente la DLL.
 - Pure: Activado previene que el segmento de memoria donde se ha alojado el recurso pueda ser modificado.
- Language: Está siempre inactivo ¿??
 - Move: Está siempre inactivo ¿??
 - Cut: Corta el recurso, es decir lo quita de la pantalla y lo lleva al portapapeles para después pegarlo mediante la opción de menú *Edit/Paste* en el proyecto en el que estamos o en otro proyecto de Resource Workshop que abramos.
 - Copy: Copia el recurso, llevando un duplicado al portapapeles para después pegarlo mediante la opción de menú *Edit/Paste* en el proyecto en el que estamos o en otro proyecto de Resource Workshop que abramos.
 - Delete: Elimina el recurso del proyecto desapareciendo totalmente, igual que si pulsamos la tecla SUPR del teclado.
 - Duplicate: Crea un duplicado del recurso dentro del mismo proyecto, igual que si lo copiamos y lo pegamos. Le da un nombre genérico que después podemos modificar.
 - Save Resource As: Graba el recurso en el disco por separado del proyecto, depende del tipo de recurso que sea se podrá grabar en distintos formatos como por ejemplo un Bitmap a archivo BMP.

Establecer preferencias de funcionamiento.

Cuando se usa por primera vez Resource Workshop se pueden establecer algunas preferencias de funcionamiento para indicar al programa como queremos que funcione, para ello elige la opción de menú *File/Preferences* y aparecerá una caja de diálogo donde se pueden establecer las siguientes opciones:

- El número de cambios que se puede deshacer (*Edit/Undo*) o rehacer (*Edit/Redo*) con un máximo de 99, el valor por defecto es de 10.
- El nombre del editor de textos que usará Resource Workshop, por defecto el NOTEPAD.EXE de Windows.
- Indicar si queremos crear ficheros de copia de seguridad (backup files) cada vez que grabemos el proyecto, que son grabados con el mismo nombre pero con extensión DL~
- Ver la versión de Windows en la que estamos trabajando, que estará establecida como Windows 3.1 y así debe de ser para trabajar con recursos para FiveWin ya que aún sigue siendo un entorno de 16 Bit.

Tipos de recursos, como crearlos y gestionarlos.

Trabajando ya dentro de un proyecto empezaremos a incorporar a este los recursos que necesite nuestra aplicación. Los recursos que permite manejar Resource Workshop son de muy varios tipos pero algunos de ellos no serán necesarios para trabajar con FiveWin en cambio si que hay otros de los que no dispone y veremos como añadir por nuestra cuenta. Los tipos de recursos que nos propone son:

- Accelerator (no usar desde FiveWin)
- Menu (no usar desde FiveWin)
- Font (no usar desde FiveWin)
- Bitmap
- Cursor
- Icon
- Stringtable
- Versioninfo
- Dialog
- Dlginit
- Rcddata

Recurso de tipo Accelerator

Un accelerator (acelerador) es una combinación de teclas que el usuario podrá pulsar desde el programa para ejecutar una tarea o una acción y en realidad son llamadas directas desde esa combinación de teclas a una opción de un menú. Para editar una tabla de aceleradores se edita también un menú desde Resource Workshop.

Los accelerator y los menús de Resource Workshop no se usan desde FiveWin por que es mucho más sencillo y cómodo editarlos desde el mismo FiveWin.

Recurso de tipo Menu

Los menús son la lista de opciones que un programa dispone situado normalmente en la parte superior de una ventana y organizados de forma lógica en grupos de elementos de parecido contenido que se despliegan al pulsar sobre el nombre de cada grupo.

Los recursos de tipo Menú de Resource Workshop no se usan desde FiveWin por que es mucho más sencillo y cómodo editarlos desde el mismo FiveWin.

Recurso de tipo Font

Un recurso de tipo Font es un fuente de caracteres o tipo de letra que se puede diseñar para luego utilizarlo desde un programa. Los Font que se diseñan desde Resource Workshop son del antiguo tipo FON de Windows 3.X y no del tipo TTF utilizado por Windows 9X mucho más eficaz, escalable a cualquier tamaño y de aspecto más elegante.

Los recursos de tipo Font de Resource Workshop no se usan desde FiveWin por que es mucho mejor utilizar los que ya están incluidos en Windows y si se quiere utilizar uno extra hay que tener en cuenta que si disponemos de el en nuestra máquina y funciona perfectamente nuestro programa habrá que instalarlo en la máquina de nuestro cliente para que funcione también el programa.

En Windows 9X esta es una tarea bastante sencilla por que basta con copiar el correspondiente fuente TTF a la carpeta de fuentes que siempre esta dentro de la carpeta en que esta instalado Windows y dentro de esta en la subcarpeta FONTS y a continuación para tener la seguridad de que funcione habría que entrar en el panel de control, carpeta fuentes, menú archivo, opción Instalar una nueva fuente para declararla y ya funcionará perfectamente.

Esta misma tarea se puede hacer más rápido desde FiveWin copiando el fichero de fuentes que incluimos con la distribución de nuestro programa

```
COPY FILE "Nombre_del_fuente.TTF" TO GETWINDIR()+"\FONTS\"+ "Nombre_del_fuente.TTF"
```

e instalando la fuente con la siguiente función:

```
ADDFONTRESOURCE(GETWINDIR()+"\FONTS\"+ "Nombre_del_fuente.TTF")
```

Lógicamente solo hay que hacerlo una vez, la primera vez que se entra en el programa que va a usar esa fuente

Ojo con los Font de los programas de Windows, no son de libre distribución y tienen copyright, pero podemos encontrar muchos otros Font en Internet que han sido cedidos al dominio público y que entonces si podemos distribuir sin problemas o diseñar un FONT propio con un programa de diseño como Coreldraw.



Recurso de tipo Bitmap

Un Bitmap es un gráfico que se puede mostrar en cualquier posición de una ventana o diálogo, delante de una opción de un menú, en un botón de la barra de herramientas, en el fondo de una ventana, etc...

Los bitmaps pueden ser dibujados desde el propio Resource Workshop pero dada la gran cantidad de imágenes BMP que nos podemos encontrar dentro de la carpeta de instalación de FiveWin en su carpeta BITMAPS, o en cualquier sitio de Internet, o en cualquier programa de diseño, es mejor tomar lo que ya esta hecho (siempre que no se vulnere algún derecho de autor) que intentar dibujar nosotros y que nos salga un bodrio de imagen (a no ser que seamos excelentes dibujantes) e incluso en esta caso para dibujar un simple BMP hay otros programas que pueden ser más completos que el simple editor de Bitmap de Resource Workshop.

Si aun así lo que se quiere es dibujar un BMP a mano basta con seleccionar simplemente la opción de menú *Resource/New* y a continuación en la caja de diálogo siguiente indicar que el recurso será de tipo Bitmap y a dibujar a ver que tal nos queda.

Para poder añadir como recurso dentro de nuestro proyecto un BMP que ya estuviera hecho y grabado en el disco es muy sencillo. Basta con indicar *File/Add to project* y de esta manera indicaremos en la caja de diálogo que aparece que el recurso es de tipo BMP y lo buscaremos por el disco para incorporarlo al proyecto. A continuación aparecerá en la ventana del gestor de proyectos con un nombre genérico de Resource Workshop que podemos cambiar pulsando sobre el con el botón derecho del ratón y seleccionando la opción Rename. Una vez incorporado el recurso Bitmap al proyecto, queda totalmente integrado dentro del proyecto de forma que el archivo BMP de donde se haya tomado el recurso ya no es necesario para trabajar ni para distribuir el programa.

Para lo que si podemos usar el programa de edición de Bitmap es para modificar un recurso de Bitmap incorporado desde el disco de forma que podremos modificar algún detalle del mismo. Normalmente lo único que aquí se cambia normalmente es el fondo de la imagen ya que en algunos casos este fondo es de algún color y nos fastidia nuestra ventana o botón ya que el resto de botones van sobre fondo gris típico de Windows y queremos que este Bitmap también se funda con el fondo. Pues bien pulsando doble clic sobre el recurso pasamos a editarlo y con la herramienta  y seleccionando el color de fondo  hacemos clic sobre el fondo del Bitmap, cerramos el Bitmap y ya esta transparente por lo cual FiveWin le aplicará de fondo el color correspondiente al fondo de donde lo hayamos puesto.

Desde la versión 2.0 de FiveWin los fondos de los botones son automáticamente calculados por el programa de forma que no seria necesario pintar el fondo salvo en casos donde FiveWin no pueda interpretar correctamente cual es el color de fondo que debe de convertir al color base de un botón o para poner el Bitmap fuera de un botón por ejemplo sobre el fondo de un diálogo.

Recurso de tipo Cursor

Un Cursor es el gráfico que vemos al mover el ratón, normalmente una flecha o un reloj de arena, pero que podemos cambiar a nuestro gusto indicando que cursor se debe de presentar al situarnos en un punto determinado de nuestro programa, como una lupa para ampliar una imagen, una mano con un dedo alzado que aparece sobre un botón para seleccionarlo, etc.

En realidad un Cursor es un Bitmap de 32 x 32 píxel de tamaño cuyo fondo es obligatoriamente transparente para que al desplazarlo por la pantalla se vea lo que hay debajo y que tiene un "Hot Point" es decir un punto caliente que es el que acciona cuando hacemos un clic con el ratón, por ejemplo en la típica flecha es la punta y en la mano con un dedo es la punta del dedo.

Exactamente igual que los Bitmap hay miles de Cursor ya hechos para que no tengamos que dibujarlos, e incluso un montón de ellos ya vienen incluidos en la instalación de FiveWin en su carpeta CURSORS o podemos encontrarlos en Internet. Su característica principal es que tienen la extensión CUR.

En el caso de querer dibujarlos procederemos igual que con un Bitmap seleccionando la opción *Resource/New* y a continuación en la caja de diálogo siguiente indicar que el recurso será de tipo Cursor y a dibujar a ver que tal nos queda.

Recurso de tipo Icon

Exactamente lo mismo pasará con los recursos de tipo Icon (Iconos). En realidad son simples Bitmap que tienen un tamaño de 16 x 16, 32 x 16, 32 x 32, o 64 x 64 píxel y que podemos usar para los mismos cometidos de un Bitmap como ponerlos en cualquier parte de una ventana, en un botón de la barra de herramientas o delante de una opción de menú (solo con Icons de 16 x 16 píxel).

Exactamente igual que los Bitmap hay miles de Icon ya hechos para que no tengamos que dibujarlos, e incluso un montón de ellos ya vienen incluidos en la instalación de FiveWin en su carpeta ICONS o podemos encontrarlos en Internet. Su característica principal es que tienen la extensión ICO.

En el caso de querer dibujarlos procederemos igual que con un Bitmap seleccionando la opción *Resource/New* y a continuación en la caja de diálogo siguiente indicar que el recurso será de tipo Icon y su tamaño en píxel y a dibujar a ver que tal nos queda.

Recurso de tipo Stringtable

Una Stringtable o tabla de textos es un conjunto de textos que podemos almacenar en un proyecto como un recurso más y se pueden utilizar para hacer programas multilingües, es decir, si queremos hacer un programa en Español, Francés e Inglés entre otras muchas cosas que tendremos que hacer, tendremos que crear una DLL por cada idioma con las pantallas correspondientes de cada idioma traducidas aun siendo en aspecto las mismas en los tres.

De esta manera al arrancar el programa y en base a alguna variable que indique en que idioma trabajar cargaremos la DLL correspondiente a un idioma y no las otras, así conseguiremos que las pantallas salgan en el idioma correspondiente sin tener que modificar prácticamente nada en el fuente de nuestro programa.

Aquí es donde cobran importancia las Stringtables ya que se pueden utilizar con tablas de mensajes en los distintos idiomas que se insertan en la DLL de forma que si en punto concreto del programa debe de aparecer un mensaje que diga en castellano "Inserte un disquete y pulsa una tecla para continuar" en Ingles deberá de decir "Insert a Disk and press any key" y claro hacer todo esto a base de IF es una locura, en cambio si creamos una tabla con Strings siendo el número 1 este mensaje en el idioma correspondiente de cada DLL, luego para que aparezca no tenemos mas que cargar desde la DLL que ya tenemos abierta el String número 1 y mostrarlo.

Las Stringtables van numeradas de 1 a n y cada mensaje tiene un número correlativo que es el que usaremos desde nuestro programa para llamar al String. Puede haber más de una Stringtable pero para que no coincidan dos números la segunda empezará con un número correlativo sobre la anterior y así sucesivamente.

Al indicar que queremos un recurso de este tipo aparece una especie de tabla tipo hoja de calculo con una primera línea donde se indica el número del String y lo único que tenemos que hacer es pulsar TAB para pasar al String o texto que queremos almacenar, lo escribimos, pulsamos INSERT y pasamos al siguiente hasta terminar.

Para usar después uno de estos Strings o cadenas de texto desde el programa habrá que utilizar la instrucción

`LOADSTRING(GETRESOURCES(), n°string)`

donde **n°string** es el numero del String dentro de la tabla del recurso que tengamos en uso.

Recurso de tipo Versioninfo

No sirven para gran cosa ya que son simple documentación interna de la DLL que podemos si queremos rellenar, después en programa no se usarán para nada.

Cuando creamos un recurso de este tipo se nos mostrará un texto donde podemos cambiar tan solo las líneas que pone VALUE e indicar en ellas el nombre de la empresa y el copyright. Estos datos solo se mostrarán si entramos desde el explorador de Windows a la DLL pulsando el botón derecho del ratón e indicando Vista Rápida.

Recurso de tipo Dialog

Las cajas de diálogo van a ser el tipo de recurso que más usaremos para nuestro trabajo. Muchos proyectos de Resource Workshop no contienen mas que un conjunto de DIALOG y nada más ya que con esto conseguimos nuestro objetivo principal, diseñar el interfaz gráfico de nuestro programa.

Los diálogos son en realidad las ventanas que iremos mostrando en nuestro programa para comunicarnos con el usuario, pedirle datos, mostrar resultados y mucho más. Sobre estos diálogos iremos situando los distintos elementos que queramos como pueden ser los Says, Gets, Listbox, Radiobotones, Checkbox, etc.

Los diálogos son en realidad ventanas pero la principal diferencia entre un diálogo y una ventana es que un diálogo es normalmente MODAL y una ventana es normalmente NO MODAL, aunque ventanas y diálogos pueden ser convertidas en modales o no modales, si fuera necesario.

Que un diálogo o ventana es MODAL significa que cuando lo mostramos en pantalla y entramos en el tendremos que terminarlo y cerrarlo para poder pasar a realizar otra tarea, ya que ese diálogo captura todo el foco de nuestro programa no dejando hacer nada más dentro del programa si no se termina primero con ese diálogo y se cierra. El ejemplo típico de una ventana modal es un mensaje de error de un programa cualquiera, si no se le hace caso y se cierra no podremos continuar.

NO MODAL, por el contrario, significa que aunque estemos en esa ventana o diálogo se pueden hacer otras cosas y abrir otras ventanas o diálogos dentro de nuestro programa, sin tener que terminar y cerrar la ventana o diálogo en el que estamos. Trabajando con ventanas no modales habrá que tener más cuidado, ya que un usuario “un poco torpe” podría abrir una ventana no modal y volver a abrir la misma 1000 veces sin cerrar la anterior, pero si se controla esto, las ventanas no modales dan muchísimo más juego a nuestro programa ya que se podrán consultar varias cosas a la vez abriendo varias ventanas para ello.

Los diálogos que creamos con Resource Workshop son en principio diálogos de tipo MODAL, aunque podemos indicar si se desea que sean de tipo no modal desde FiveWin si al activar el diálogo con `ACTIVATE DIALOG` indicamos que no se detenga y espere en él y que continúe el curso del programa con la cláusula `NOMODAL` o `NOWAIT` (son lo mismo).

Para crear un recurso nuevo de tipo DIALOG elegimos la opción de menú *Resource/New* para después elegir un tipo de DIALOG de entre un conjunto de plantillas de diálogo prefijadas, de esta manera proporciona una forma rápida de crear diálogos eligiendo un tipo de entre seis diálogos predefinidos, cada vez que se quiera crear un diálogo nuevo.

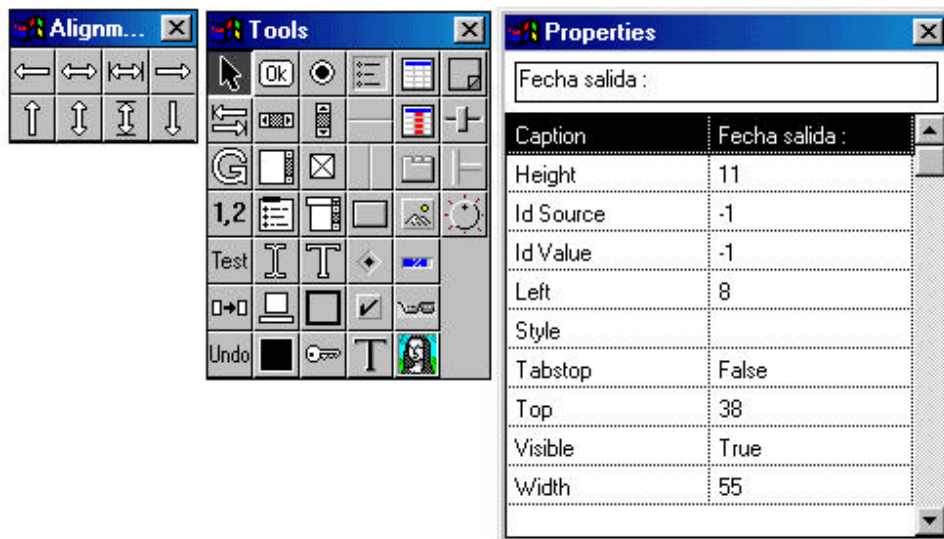
1. Windows dialog. Buttons on right: Crea un diálogo de tipo estándar con los botones OK, Cancel, y Help a la derecha de la misma.
2. Windows dialog. Buttons on bottom: Crea un diálogo de tipo estándar con los botones OK, Cancel, y Help en la parte inferior de la misma.
3. Borland dialog. Buttons on right: Crea un diálogo de tipo BORLAND con los botones OK, Cancel, y Help a la derecha de la misma.
4. Borland dialog. Buttons on bottom: Crea un diálogo de tipo BORLAND con los botones OK, Cancel, y Help a la derecha de la misma.
5. Child dialog. No buttons: Crea una ventana de tipo hija sin botones. Estas son unas ventanas sin bordes ni título ni botones y que se usarán para incluir en el interior de otras que puedan aglutinar varias de estas como son los Folder o los Tab.

6. Standard window. No buttons: Es la opción habitual, crea una ventana de tipo estándar sin botones.

Podemos elegir cualquiera de ellas y llegar al mismo resultado final pero por comodidad distingamos que las dos primeras son diálogos de tipo Windows, las dos siguientes diálogos de tipo Borland (mejor usar los demás) y las dos últimas son ventanas de Windows. En mi caso siempre uso la opción 6 Standar Windows y con ella llego siempre al diálogo que necesito.

Al elegir una de las opciones DIALOG se nos muestra una ventana con una caja de diálogo vacía o con tres botones (Ok, Cancel y Help que podemos borrar o modificar) para ir rellenándola con controles. Se llaman así a los distintos elementos que podemos poner sobre una ventana para editar su diseño, SAY, GET, BITMAPS, BOTONES y demás son los controles típicos de Windows.

En la esquina superior derecha de la ventana se muestran dos o tres barras de herramientas flotantes, la última es opcional y solo se ve si se selecciona la opción de menú *Options/Show properties*



La primera **Alignment** la utilizaremos para alinear los controles que vayamos poniendo en el diálogo para una mejor presentación del diseño, la segunda **Tools** para elegir los controles que iremos poniendo en el diálogo y la tercera **Properties**, si queremos usarla, para dar título a los controles y cambiar las sus características aunque no es necesaria puesto se puede hacer lo mismo pulsando doble clic sobre ellos.

Menú de herramientas TOOLS

Este es el menú más importante pues lo que después haga el diálogo dentro de nuestro programa dependerá de los controles que hayamos puesto sobre él.

En el menú de herramientas podemos elegir entre los siguientes tipos de controles:

Las **herramientas de diseño** no son en realidad controles puesto que sirven en realidad para trabajar sobre los demás controles ordenándolos o testándolos.



Selector: sirve para seleccionar, mover y cambiar de tamaño a los elementos. Es el botón que debe estar pulsado por defecto para el trabajo normal, es decir, si usamos cualquiera de las otras herramientas de diseño tenemos que volver a pinchar en esta para seguir trabajando.



Tab Stop: Cuando se vean los controles se verá la propiedad Tab Stop que sirve para que el en el paso de un control a otro se pueda detener el cursor en el o no. Si activamos este botón y hasta que pinchemos en el de selección, haciendo clic sobre cualquiera de los controles del diálogo activamos esta propiedad o la desactivamos si el control la tiene activada. Es decir, indica en que controles se puede entrar a trabajar o no.



Group: Cuando se vean los controles se verá la propiedad Group que sirve para indicar que controles son un grupo. Si activamos este botón y hasta que pinchemos en el de selección, haciendo clic sobre cualquiera de los controles del diálogo activamos esta propiedad o la desactivamos si el control la tiene activada.



Item Order: Indica en que orden se va a saltar de un item a otro dentro del diálogo pulsando la tecla del TABULADOR, en FiveWin si se quiere también con Intro. Si activamos este botón y hasta que pinchemos en el de selección, haciendo clic sobre cualquiera de los controles del diálogo ordenamos el control en número creciente, del primero al último. Es conveniente ponerlos todos en orden aunque no tengan Tab Stop. Veremos más adelante un truco para ordenarlos eficazmente en la sección **Ordenar el salto entre controles**.



Test: Muestra el diálogo en funcionamiento, es decir lo testea tal y como lo veremos después desde el programa en FiveWin.



Duplicate control: Duplica el control que está seleccionado en la posición que se le indique, normalmente debajo del mismo. Se puede también copiar y pegar pero puede que sea más rápido duplicarlo y luego moverlo de sitio si es necesario.



Deshacer: deshace la última acción efectuada en el diálogo y hasta las n veces que se hayan indicado en *File/Preferences*.

Los controles **genéricos de Windows** son los que nos encontramos en todo tipo de programas de Windows para las acciones normales de los programas. Serán explicados más adelante con detenimiento.



Button



Horizontal Scroll Bar



Vertical Scroll Bar



Listbox



Combobox



Static Text



Edit Text



Group



Radiobutton



Checkbox



Icon



Black Rectangle



Black Frame




Custom control

Los controles de **tipo Borland** son controles especiales de Borland que pueden sustituir a los genéricos de Windows, dando un aspecto distinto a nuestro programa usando por ejemplo botones que tengan un gráfico y un texto en vez de los habituales de Windows de solo texto o solo gráfico, pero para poder usarlos hay que tener mucho cuidado puesto que pueden dar algunas pegas. Mas adelante veremos como trabajar con ellos aunque mi recomendación suele ser simplemente que nos olvidemos de ellos y no los usemos en nuestros programas ya que lo único que cambian es el aspecto saliéndose del estándar de Windows y no ganando nada a cambio. También se verán después junto a los anteriores.



Los controles **específicos de FiveWin** son una serie de controles que se han añadido de forma separada al Resource Workshop y que sirven para diseñar ventanas súper completas con controles que después nuestro programa en FiveWin utilizará directamente.

En principio Resource Workshop no los incorpora pero es muy fácil añadirseles y mucho más cómodo usarlos en vez de tener que estar añadiendo estos controles mediante el icono de control personal del usuario  y tener que estar continuamente definiendo controles de usuario nuevos.


Para añadirlos hay que conseguir primeramente una extensión para Resource Workshop diseñada por el genial Ricardo Ramírez R. rramirez@landatech.com, que es totalmente gratuita y que se llama FWCTRLS.DLL, el autor solo requiere que nos registremos. La podemos encontrar en las nuevas distribuciones de FiveWin en el directorio Dll o en el sitio <http://www.landatech.com> y se puede añadir permanentemente a Resource Workshop grabándola en su misma carpeta y activándola desde la opción de menú *File/Install control library* y seleccionando FWCTRLS.DLL de esta carpeta. Más adelante se verá la aplicación de estos controles desde Resource Workshop.



Según vayamos añadiendo cualquiera de estos controles al diálogo se le irán asignando identificadores en principio numéricos. Los identificadores son imprescindibles después para nuestro programa y pueden ser automáticamente generados según se crean. Esta opción puede ser controlada desde la opción de menú *File/Preferences - Generate identifiers automatically*.

Menú de herramientas ALIGNMENT

Son las herramientas que utilizaremos para alinear los controles ya puestos en el diálogo y conseguir que la ventana quede con un aspecto lo más ordenado y profesional posible.

Antes de comenzar a alinear es importante seleccionar los controles que queremos alinear en un grupo de controles. Haciendo clic sobre un control este queda seleccionado, pero si queremos seleccionar más de uno para formar un grupo, en los siguientes controles a seleccionar debemos de pulsar y mantener una de las dos flechas de mayúsculas  antes de hacer clic en los demás controles que se irán uniendo al primero, así conseguimos un grupo de controles que se pueden alinear juntos, aunque también agruparlos sirve para más acciones que la de alinearlos.

Si queremos dejar de seleccionar el grupo o seleccionar otro distinto debemos simplemente de hacer clic en el fondo del diálogo o sobre un control cualquiera de este con lo que se deshace el grupo.

Una vez seleccionado un grupo de controles las opciones de alineación, que pueden ser combinadas entre si, son las siguientes:



Alinea todos los controles del grupo seleccionado a la izquierda del control situado más a la izquierda de entre todos los del grupo seleccionado.



Alinea todos los controles del grupo seleccionado a la derecha del control situado más a la derecha de entre todos los del grupo seleccionado.



Alinea todos los controles del grupo seleccionado a la parte se arriba del control situado más arriba de entre todos los del grupo seleccionado.



Alinea todos los controles del grupo seleccionado a la parte se abajo del control situado más abajo de entre todos los del grupo seleccionado.



Centra todos los controles del grupo seleccionado al centro horizontal relativo de la selección de todo el grupo, situándolos uno sobre otro.



Centra todos los controles del grupo seleccionado al centro horizontal del diálogo pero no centra los controles uno sobre otro, aunque puede combinarse con el anterior para conseguirlo.



Centra todos los controles del grupo seleccionado al centro vertical relativo de la selección de todo el grupo, situándolos en una misma línea.



Centra todos los controles del grupo seleccionado al centro vertical del diálogo pero no centra los controles en una misma línea, aunque puede combinarse con el anterior para conseguirlo.

Además de para alinearlos, agrupar controles también nos servirá para poder hacer otras acciones que pueden ser muy interesantes sobre todo grupo conjuntamente en vez de hacerlas control a control, estas acciones con las siguientes:

Mover todo el grupo junto a otra posición del diálogo pinchando sobre uno de los controles seleccionados y arrastrando el conjunto por el diálogo.

Eliminar el conjunto de controles pulsando la tecla SUPR.

Copiar los controles seleccionados para pegarlos después en el mismo diálogo o en otro cualquiera. OJO hay una opción de duplicar un control de la que hemos hablado antes *Duplicate Control* entre las herramientas de diseño, pues esta no funciona bien cuando hemos seleccionado un grupo, solo vale para duplicar un control unitario, si queremos duplicar un grupo hay que copiarlo y pegarlo y para ello podemos seleccionar los controles deseados pulsar clic derecho sobre cualquiera de ellos y elegir *Copy* y después pegarlos desde el menú *Edit/Paste* o pulsando el botón derecho sobre los mismos controles (no vale en el fondo del diálogo) elegir *Paste* y hacer clic donde se quieran pegar en el fondo del diálogo. Más adelante veremos un truco para poder *Duplicar controles rápidamente*.

Para cambiar alguna o algunas propiedades de todos ellos consecutivamente. Seleccionando los controles y pulsando INTRO o doble clic sobre uno de ellos aparecerá la ventana de diálogo de propiedades del primer control seleccionado y después de cambiar la propiedad que queramos la cerramos y aparecerá el diálogo de propiedades del siguiente seleccionado y así hasta el último seleccionado en el orden en que los seleccionamos.

Ajustar la anchura y altura: para ajustar la anchura y altura de un control podemos seleccionarlo y estirar de sus bordes pero si queremos ser un poco más exactos podemos pulsar clic derecho sobre el control y seleccionar la opción *Size* con la cual podemos indicar el tamaño exacto que queremos para el control en las casillas CX (anchura en píxel) y CY (altura en píxel). Pero esto no sirve para los grupos, en caso de seleccionar un grupo podremos dar a todos los controles un tamaño homogéneo mediante unas opciones concretas, esto lo vemos en el siguiente tema.

Dar una acabado profesional a nuestro diseño.

Pulsado clic derecho sobre un control o más bien sobre un grupo de controles podemos obtener también un conjunto de opciones muy interesante para dar un acabado rápido y profesional a nuestro diálogo sin tener que sudar demasiado, este conjunto de opciones están bajo los nombres de **Size y Align**.

Size controls: Para conseguir que un grupo de controles tengan una misma altura o anchura tenemos que seleccionar el grupo de controles y pulsar clic derecho sobre cualquiera de ellos seleccionando *Size* obteniendo el siguiente diálogo de Resource Workshop

Veremos en él que tenemos opciones para el ajuste del tamaño horizontal y vertical por separado pero que se repiten tanto para el uno como para el otro.



No change: dejar los controles sin cambios, o sea nada.

Shrink to smallest: encoge los controles hasta el tamaño horizontal o vertical del más pequeño de ellos, quedando todos del mismo tamaño horizontal o vertical.

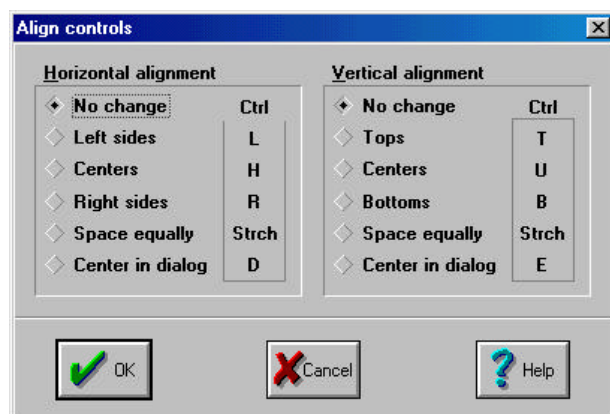
Grow to largest: Aumenta los controles hasta el tamaño horizontal o vertical del más ancho o alto de ellos, quedando todos del mismo tamaño horizontal o vertical.

Width of size box: iguala las anchuras de las cajas que contienen los controles hasta el tamaño horizontal o vertical de la caja más amplia de ellos. No muy útil.


Width of dialog: iguala las anchuras de las cajas que contienen los controles hasta el tamaño horizontal o vertical de todo el diálogo. No muy útil.


Align controls: Para conseguir que un grupo de controles tengan un aspecto alineado y ordenado tenemos que seleccionar el grupo de controles y pulsar clic derecho sobre cualquiera de ellos seleccionando *Align* obteniendo el siguiente diálogo de Resource Workshop

Veremos en el que tenemos opciones para la alineación horizontal y vertical por separado pero que son similares tanto para el uno como para el otro. En realidad son prácticamente las mismas opciones que ya hemos visto en el menú de herramientas Alignment pero hay algún otro elemento nuevo y muy interesante.




No change: dejar los controles sin cambios, o sea nada.



Left sides: es lo mismo que  Alinea todos los controles del grupo seleccionado a la izquierda del control situado más a la izquierda de entre todos los del grupo seleccionado.



Right sides: es lo mismo que  Alinea todos los controles del grupo seleccionado a la derecha del control situado más a la derecha de entre todos los del grupo seleccionado.

Tops: es lo mismo que  Alinea todos los

controles del grupo seleccionado a la parte se arriba del control situado más arriba de entre todos los del grupo seleccionado.

Bottoms: es lo mismo que  Alinea todos los controles del grupo seleccionado a la parte se abajo del control situado más abajo de entre todos los del grupo seleccionado.

Centers: es lo mismo que  o  Centra todos los controles del grupo seleccionado al centro horizontal o vertical relativo de la selección de todo el grupo, situándolos uno sobre otro.

Center in dialog: es lo mismo que  o  Centra todos los controles del grupo seleccionado al centro horizontal o vertical del diálogo.

Hasta aquí nada nuevo pero tenemos además dos opciones nuevas y muy interesantes:



Space equally: toma el conjunto de controles seleccionado y tomando como referencia el más izquierdo y el más derecho en caso de horizontal, o el superior y el inferior de ellos en caso de vertical, los distribuye uniformemente repartiendo el espacio entre control y control exactamente igual moviendo todos ellos salvo los dos extremos de referencia.

Con esto último y con los Size igualados en altura y en anchura, si procede, podemos dar un toque final muy profesional a nuestro diálogo, digno del programa de gestión más elaborado que podamos conocer.

Duplicar controles rápidamente.

Para no andar igualando los sizes podemos copiar y pegar un o varios controles ya hechos anteriormente de forma que los nuevos sean idénticos y la forma más rápida de hacerlo es seleccionando el control o los controles a duplicar (por ejemplo un Static Text o SAY y un Edit Text o GET) y después con la tecla **CONTROL** pulsada movemos los controles a su nueva posición y así son duplicados al momento variando por supuesto el ID automáticamente. Así no cabrá duda en que ambos son idénticos aunque luego puedan ser modificados por separado.

Ordenar el salto entre controles.

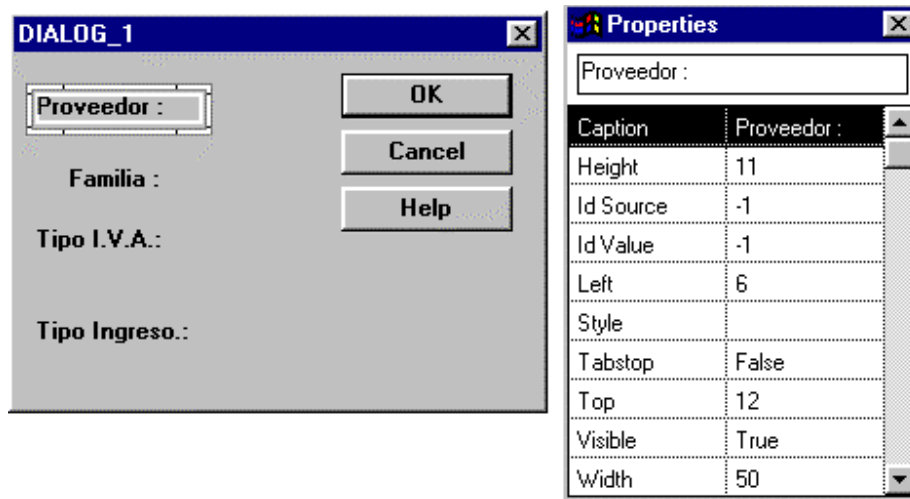
Hemos comentado ya que podemos y debemos ordenar el salto entre controles mediante  Item Order. Para ello basta con pulsar sobre esta herramienta e ir haciendo clic sobre cada uno de los controles (incluso los que no tengan TabStop como los Static Text o SAY) de forma que iremos ordenándolos uno a uno hasta el último, momento en el que finaliza automáticamente la opción y el programa vuelve al estado de selección activando la herramienta  Selector. También podemos interrumpir el orden pulsando nosotros sobre el Selector.

Ordenar los controles es muy fácil de hacer, pero a veces es engorroso ya que al insertar un control nuevo dentro de un diálogo ya hecho con un buen número de controles, éste nuevo control pasa a ser el último en el orden de salto entre controles y entonces hay que volver a numerarlos todos desde el primero hasta el último.

Pues también para esto hay un truco que consiste en lo siguiente: Insertamos el nuevo control donde queramos y a continuación seleccionamos este control y con la mayúscula pulsada todos los que vengan a continuación de él (incluso abriendo un área de selección con la mayúscula pulsada sin tener que hacer un clic en cada uno por separado). Una vez seleccionados pasamos a pulsar la herramienta de Orden y con esto pasará a dejarnos ordenar solo los controles seleccionados. Hacemos clic sobre el nuevo control insertado que pasará a ser el primero de todos ellos y salimos del orden pulsando sobre la herramienta Selector. Con esto ya están todos los controles correctamente ordenados, sin volvernos locos para ello.

Menú de propiedades PROPERTIES

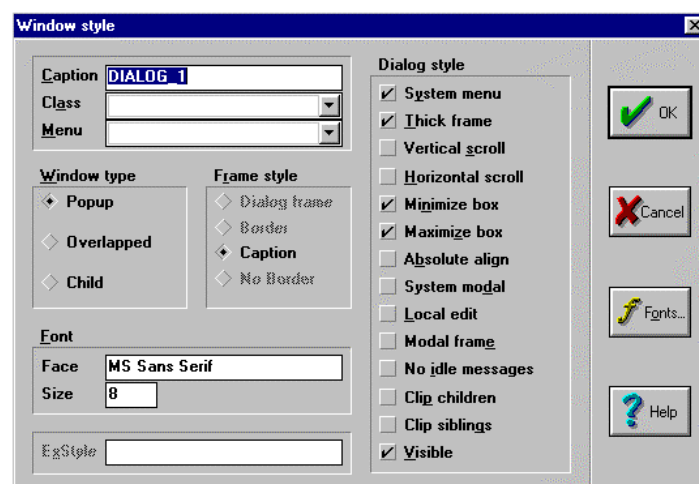
Son las propiedades del control que tengamos seleccionado, no se verá por lo tanto nada si no seleccionamos primero un control del diálogo y lógicamente no se verá ni siquiera la pantalla de propiedades si previamente no la activamos tal y como hemos indicado antes desde el menú *Options/Show properties*



Tampoco es imprescindible tener activada esta opción ya que en caso de no ver las propiedades aquí se ven directamente pulsando doble clic sobre los propios controles tal y como veremos a continuación cuando veamos como situar los controles en el diálogo. Pero si la tenemos activada podremos ver y cambiar más ágilmente las propiedades de los controles que entrando en ellos con doble clic.

Propiedades de un DIALOG

También un diálogo tendrá sus propiedades pero no se muestran de la forma descrita en el punto anterior. Para poder ver las propiedades de un diálogo tenemos que tener primeramente el diálogo en pantalla, si no lo tenemos habrá que crear uno con ResourceNew, opción Dialog y un tipo cualquiera de diálogo y a continuación podemos acceder a sus Propiedades haciendo doble clic sobre él y obtendremos la siguiente pantalla donde podemos variar sus características de funcionamiento:



CAPTION Nos permite establecer un título a la ventana

CLASS Nos permite seleccionar uno de los brush estándar para un dialog. Si dejamos este campo en blanco, posteriormente podremos definir un brush mediante la cláusula correspondiente del objeto oDlg.

Si indicamos aquí un brush, éste prevalece sobre el que posteriormente podamos definir.

Los brush disponibles son: Bordlg: Estándar Borland y Bordlg_gray: Gris

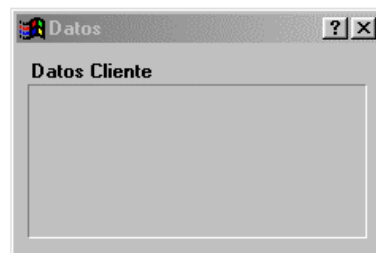
MENU No se usa en FiveWin

WINDOW TYPE Si seleccionamos los botones Popup u Overlapped nuestro diálogo será independiente de cualquier otro elemento, mientras que si seleccionamos Child, éste será "anclado" a la ventana o control al que pertenece.

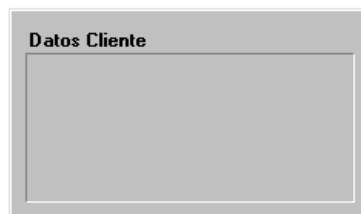
Si definimos un dialog con caption o barra de título y de tipo Popup u Overlapped asociado a una ventana podremos desplazarlo libremente por todo el escritorio de Windows, mientras que si el mismo dialog lo definimos de tipo Child, sólo podremos desplazarlo dentro de los límites de la ventana que lo contiene.

FRAME STYLE Nos permite indicar si nuestro dialog dispondrá de barra de título, en cuyo caso también podremos desplazarlo de posición, o carece de ella y se quedará fijo sin posibilidad de desplazarlo. En este último caso podremos seleccionar el borde deseado para el dialog siempre y cuando no sea dimensionable (no tenga la propiedad Thick Frame), ya que si podemos dimensionar el dialog, siempre toma el estilo Dialog Frame. Para cambiar de STYLE tendremos primero que borrar el título de la ventana.

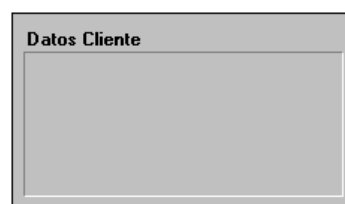
Frame Style: Caption



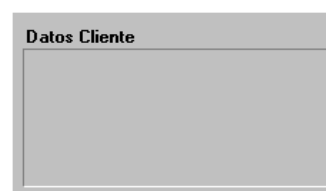
Frame Style: Dialog Frame



Frame Style: Border



Frame Style: No Border



FONT Establece el tipo de letra por defecto para el dialog. A través del botón Fonts podemos seleccionarlo de entre las fuentes que disponga nuestro ordenador pero como siempre que utilicemos un tipo de letra, hay que tener presente que no todos los equipos tendrán instalados las mismas fuentes que nosotros, por lo cual procuraremos emplear siempre fuentes estándar de Windows (Arial, Times new roman) y lo mejor es que no usemos esta opción y dejemos a la ventana que tome el estándar de Windows 9X de la siguiente manera. Una vez creado el diálogo salimos de el con Ctrl+F4 o con doble clic en el icono a la izquierda del menú *File*. Pulsamos clic derecho sobre su nombre y aparecerá un menú de opciones donde primero podríamos cambiar su nombre con la opción *Rename* y a continuación con la opción *Edit as text* nos mostrará su estilo en la línea Style como por ejemplo:

STYLE WS_POPUP | WS_VISIBLE | WS_CAPTION | WS_SYSMENU | WS_THICKFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX

Al final de esta línea tenemos que añadir `| 0x4L` y de esta manera tomará el fuente por defecto de todas las ventanas de Windows 95 o 98 y si el usuario la altera desde el panel de control el programa lo asumirá.

DIALOG STYLE El grupo de opciones Dialog Style nos permite seleccionar toda una serie de propiedades adicionales a la caja de diálogo. Muchas de las opciones propuestas por Workshop no tienen efecto en programas diseñados con Fivewin, por lo cual sólo veremos las que podremos utilizar dentro de nuestros programas.

- System Menu: Muestra o no el botón cerrar situado en la barra de título
- Thick Frame: Permite o no redimensionar el diálogo por el usuario
- Vertical Scroll: Incorpora una barra de desplazamiento vertical al diálogo
- Horizontal Scroll: Incorpora una barra de desplazamiento horizontal al diálogo
- Minimize Box: Muestra el botón minimizar
- Maximize Box: Muestra el botón maximizar
- System Modal: La caja de diálogo es modal, pero de todo el sistema, esto es, no podremos acceder a ninguna otra ventana o proceso de Windows (incluidos los que no pertenecen a nuestra aplicación) hasta que no cerremos este diálogo.
- Absolute Aling: Si el diálogo está asociado a una ventana fuerza a que las coordenadas de posición iniciales de dicho diálogo estén asociadas al escritorio de Windows y no a la ventana.

Establecer preferencias de funcionamiento dentro de la edición de un diálogo

Dentro de un diálogo podemos variar ciertas preferencias de funcionamiento, siendo alguna de ellas muy recomendable variar respecto a la configuración inicial, lo podremos hacer solo cuando nos encontremos dentro de un diálogo y desde la opción de menú *Options/Preferences*:

- Status line units: Indica si las coordenadas que se nos muestran en la línea de estado son respecto a la pantalla o respecto al diálogo. Poco interesante.
- Selection Border: Al seleccionar un conjunto de controles muestra un borde fino (Thick Frame) o un borde con los puntos cuadrados de estirar los controles (Handles). Poco interesante puesto que no se pueden estirar todos ellos para cambiarlos de tamaño juntos.
- Drawing Type: Muestra el diálogo dibujado de varias maneras:
 - Draft muestra el diálogo con los ID de los controles sobre estos en modo borrador, útil si se quieren tener a la vista todos los ID cuando estamos escribiendo el código fuente del programa y necesitamos usarlos.
 - Normal muestra el dibujo en modo borrador pero sin los ID, lo cual no es realmente útil.
 - Wysiwyg es la forma estándar de trabajo que muestra el diálogo tal y como lo verá el usuario desde el programa.
- Selection Options: Con estas dos opciones indicamos como queremos seleccionar los controles del diálogo:
 - Selection near border: Si activamos esta opción solo deja seleccionar un control pinchando en su borde, si la dejamos desactivada podemos seleccionar el control pulsando en cualquier sitio lo cual en principio es más cómodo.
 - Selection rectangle surrounds: Esta opción es muy interesante, si se activa indica que al seleccionar un conjunto de controles, abriendo un área con el ratón, solo se seleccionaran aquellos que estén totalmente incluidos en el área de selección y de esta forma se evitará el problema que tenemos habitualmente al seleccionar por proximidad (con esta opción desactivada) cuando hay controles que están muy juntos o hay entre ellos controles de tipo combobox que ocupan un área tan grande que cubre a otros controles del diálogo y que por más que queramos siempre son seleccionados solo con incluir un punto del área donde se despliegan en la selección o cuando tenemos un grupo y al seleccionar un control de su interior queda siempre seleccionado.
- Generate CONTROL Statements Only: Genera propiedades de control en lugar de controles en el diálogo. Esto no afecta para nada a los controles que usemos desde FiveWin pero no alcanzo a comprender si tiene alguna contraindicación.
- Draw Custom Controls As Frames: Solo funciona si el Drawing Type es normal y si se activa dibuja rectángulos vacíos para los controles, si se desactiva dibujará rectángulos grises. No es interesante.
- Use CTL3DV2.DLL: necesario para que los efectos 3D de los controles se muestren como tales. Es muy recomendable activarlo para que no tengamos problemas de visualización.

Todos estos controles una vez de modificados quedan permanentemente guardados por el programa hasta que volvamos a cambiarlos.

Veamos ahora como insertar controles dentro de un diálogo los distintos tipos de controles y como editar sus propiedades.

Control BUTTON (Botón de texto)



Control tipo Borland.

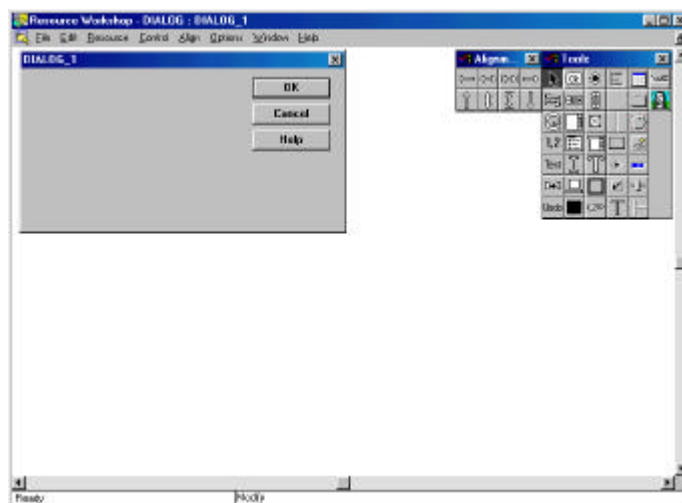
Los botones son esos rectángulos grises con un borde que los hace parece elevados y que aparecen en todas las cajas de diálogo. Su cometido es que al apretarlos con la flecha del ratón, se hundan y desencadenan una acción que tienen programada.

Los hay de dos tipos, genérico de Windows con un texto en su interior y de tipo Borland con una imagen y un texto en su interior, si se quieren botones con un gráfico es mejor utilizar los propios BTNBMP de FiveWin que veremos más adelante ya que como hemos comentado los controles Borland dan más de un dolor de cabeza.


Una vez dibujado el control y puesto el texto correspondiente ya es plenamente operativo y lo que nos queda es decirle después desde nuestro programa que acción queremos que haga al pulsar sobre él.

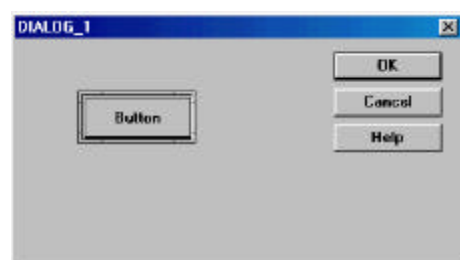
Para añadir un botón a un diálogo tenemos que tener primeramente el diálogo en pantalla, si no lo tenemos habrá que crear uno con *Resource/New*, opción *Dialog* y un tipo cualquiera de diálogo, el primero por ejemplo.

La pantalla será parecida a la siguiente:



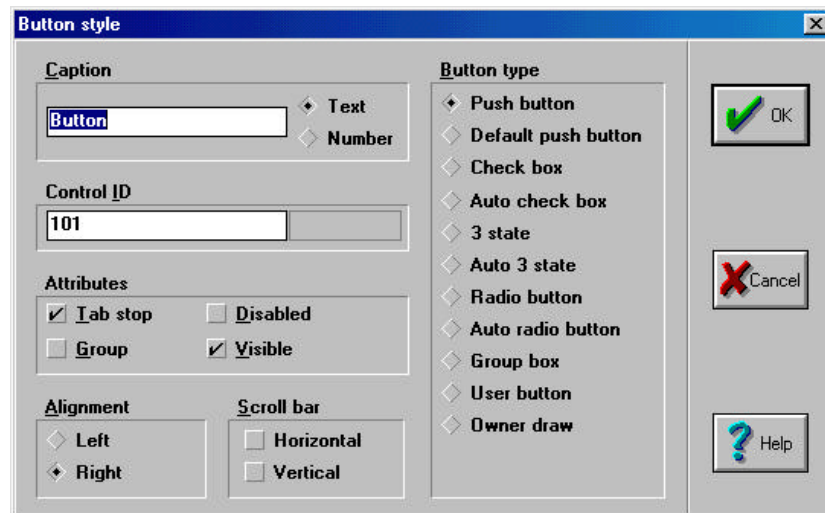
Luego elegiremos en la barra TOOLS el control de Windows  o el de Borland 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un botón siendo su borde superior izquierdo el punto donde hemos hecho clic.



El botón, al igual que el resto de los controles, se puede hacer más grande o más pequeño estirando de sus bordes y mover de sitio estirando desde el centro del mismo.

Una vez que tenemos el botón en el diálogo podemos acceder a sus **Propiedades** haciendo doble clic sobre él y obtendremos la siguiente pantalla donde podemos variar sus características de funcionamiento:



Veamos las propiedades de un botón, algunas que no comentaremos se encuentran en este diálogo para otros controles y no para un botón normal como pueden ser Check Box o Radio Button que en caso de activarlos convierten el Button en otro control distinto.

Caption es el texto aparecerá en el botón. Si al texto le añadimos un símbolo & a la izquierda de una letra, por ejemplo &Aceptar aparecerá sobre el botón como Aceptar con la letra subrayada, lo cual indica que podremos usar un atajo para pulsar sobre esta tecla sin usar el ratón, que será en este caso ALT+A (la tecla subrayada), cuidado de no poner a varios botones la misma letra subrayada.

Control ID el número de identificación del control, aparecerá automáticamente un número de serie del control aunque es posible modificarlo pero con mucho cuidado de no tener dos con el mismo número ya que este número es el que luego usaremos para relacionar el control desde nuestro programa.

Un detalle muy importante: los botones más comunes de todos los diálogos como el botón de ACEPTAR, el de CANCELAR o el de AYUDA (si lo hay) deben de tener obligatoriamente los ID 1, 2 y 3 respectivamente pongamos el Caption que pongamos en ellos y ningún otro control incluso no estando estos puede tomar estos números de ID por que no funcionaria apropiadamente ya que el control número 1 es el que responde a la pulsación de la tecla INTRO, el 2 a la tecla ESC y el 3 a la tecla F1.

Sección Attributes

- **Tab stop** indica si el control tomará el foco cuando se pulse la tecla Tabulador y se llegue a el.
- **Group** indica si nuestro control formará parte de un grupo de controles. No es usual para un botón.
- **Disabled** indica si el botón se mostrara inicialmente desactivado, es decir, difuso y sin posibilidad de funcionamiento aunque luego podremos habilitarlo desde nuestro programa mediante oNombreboton:Enable() o volver a desactivarlo con oNombreboton:Disable()
- **Visible** indica si el botón está inicialmente visible o si lo desactivamos quedará invisible. Si no es visible se podrá visualizar mediante oNombreboton:Show() y volver a ocultar con oNombreboton:Hide().

Sección Justification

No tiene sentido en los botones, su texto siempre está centrado.

Sección Scroll Bar

No tiene sentido en los botones, un botón no suele tener barra horizontal ni vertical de desplazamiento.

Sección Button type

Las únicas opciones útiles para un botón son Push Button que indica que es un botón y que es la opción por defecto y Default Push Button que indica que es el botón que se acciona por defecto cuando se pulsa INTRO dentro del diálogo y por lo tanto solo puede haber uno y normalmente será el botón de aceptar. El problema que podemos tener en un programa FiveWin es que al poder elegir que se pueda saltar de un control a otro con INTRO el Default Push Button dejaría de tener sentido que solo lo tendrá en el caso de usar la tecla del tabulador para pasar de un control a otro que es lo estándar.



Como hemos dicho también tenemos botones de tipo Borland que incluyen un dibujo. La forma de utilizarlos es la siguiente:

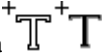
Después de hacer clic en el diálogo para poner el botón hay que utilizar los identificadores ID definidos por Borland que van desde el número 1 correspondiente al OK hasta el 999 . Estos botones predefinidos están almacenados y pueden verse abriendo el archivo BWCC.DLL con el propio Resource Workshop.

También se pueden añadir botones propios como botones tipo Borland en cuyo caso tendremos que dibujar íntegramente tres bitmap para los tres estados posibles del botón: reposo, con el foco y pulsado. Luego almacenaremos los tres dibujos con los ID 1000+númeroid, 3000+númeroid y 5000+númeroid y usaremos como ID en las propiedades del botón el númeroid que hemos utilizado que no se puede repetir para dos botones distintos. Más adelante veremos como usar botones de Borland puede darnos algún problema no ganando a cambio mas que un aspecto distinto para los diálogos ya que su funcionamiento es similar que si usamos botones Windows estándar.

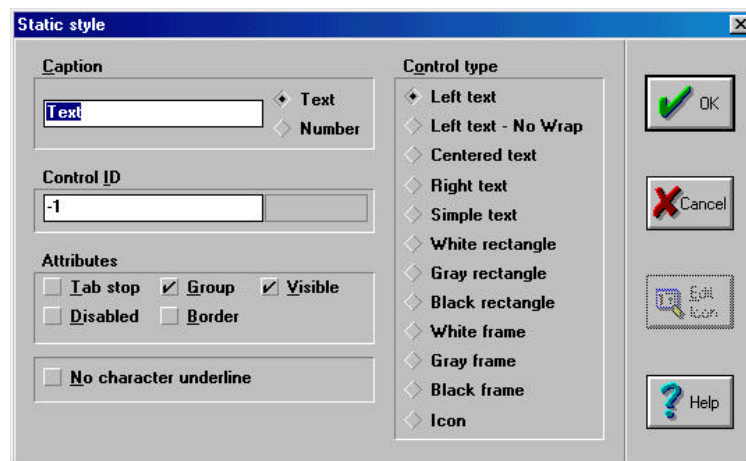
Control STATIC TEXT (Texto estático SAY)

Mediante este control podemos situar en el diálogo un texto estático, esto es, un texto que luego desde nuestro programa el usuario no va a poder editar o teclear sobre él, es decir lo que desde un programa haríamos con una instrucción SAY “Esto es un texto”. En cambio nosotros desde programa si podremos variar su contenido cuando queramos, por lo tanto no es tan estático. Un texto puede tener varias líneas (basta con hacer el control más alto) y desde el programa se le puede cambiar de fuente y color para realzarlo.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control de Windows  o el de Borland 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón y si queremos tiramos de él hasta abrir un cuadro de suficiente tamaño para escribir en él lo que queramos. Aparecerá en la caja de diálogo un cuadro con un texto “Text”, siendo su borde superior izquierdo el punto donde hemos hecho clic.

Pulsando después doble clic sobre el cuadro del texto entraremos en sus propiedades apareciendo el siguiente diálogo donde veremos sus propiedades:



Las propiedades ya vistas para el control Button y que no varían como por ejemplo Tab Stop, funcionan de la misma manera y no hace falta volver a explicarlas, veamos cuales cambian y algunas nuevas.

Caption que podemos cambiar para que poner el texto que queremos mostrar

ID automáticamente asignado como -1 para todos los textos ya que normalmente no se varían desde el programa y por ello no hace falta que cada uno tenga un ID propio, -1 es como no ponerle nada. Ahora si queremos poder variar alguno de ellos por cualquier motivo, hay que asignarle obligatoriamente otro ID distinto y propio no repetido.

Sección Attributes

- **Tab stop**
- **Group** Un texto se puede utilizar como cabecera de grupo seleccionando esta opción.
- **Disabled**
- **Visible**
- **Border** Para que el texto aparezca enmarcado con un borde.

No Character underline en caso de marcarlo indica que podemos incluir en el texto caracteres '&' sin que nos produzca el subrayado característico del carácter siguiente.

Left text para ajustar el texto a la izquierda del área del cuadro.


Left text no Warp para ajustar el texto a la izquierda del área del cuadro como el anterior, pero si el texto fuera largo y tuviera varias líneas el anterior las reparte al hacer el control más alto y este no las reparte.


Centered text para ajustar el texto en el centro del área del cuadro.

Right text para ajustar el texto a la derecha del área del cuadro.

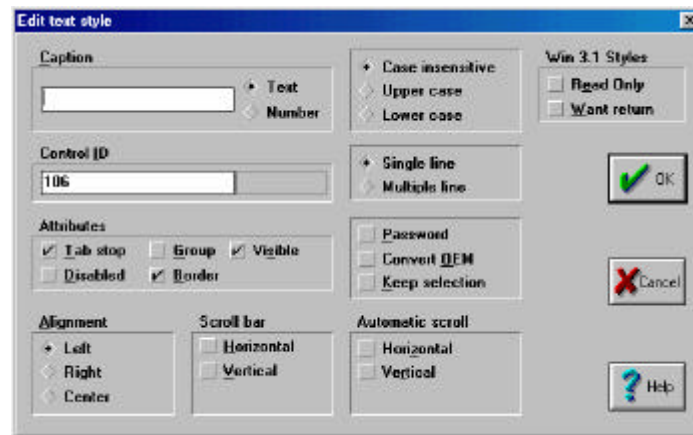
Control EDIT TEXT (Texto editable GET)

Un campo de este tipo sirve para que el usuario del programa pueda introducir información dentro de él, es lo que se conoce como un cuadro de edición (GET) y puede ser de una sola línea para introducir campos simples como el nombre o el teléfono de un cliente, como para introducir campos multilínea para editar Memos como las observaciones de un cliente.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control de Windows  no habiendo un equivalente de tipo Borland.

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón y si queremos tiramos de él hasta abrir un cuadro de suficiente tamaño para escribir en él lo que queramos. Aparecerá en la caja de diálogo un cuadro vacío, siendo su borde superior izquierdo el punto donde hemos hecho clic.

Pulsando después doble clic sobre el cuadro del texto entraremos en sus propiedades apareciendo el siguiente diálogo donde veremos sus propiedades:



Caption podemos incluir un texto inicial a mostrar dentro del control o lo más normal dejarlo vacío para que sea nuestro programa el que ponga los datos en él.

ID generado automáticamente.

Sección Attributes nada nuevo

Alignment Left ajuste izquierdo, Right ajuste derecho y Center ajuste izquierdo de lo que se escriba dentro desde nuestro programa. Hay un truco muy importante que se debe de hacer notar, si se quiere un cuadro de edición para un número que se escriba de derecha a izquierda no basta con pedir que su justificación sea Right ya que en algunas versiones de Windows 95 no funcionará, además hay que indicar que sea Multiline aunque en realidad no lo sea ya que solo le daremos la altura de una línea, de lo contrario la justificación derecha no funcionará. Windows 98 no tiene este problema pero es mejor ponerlo por si nuestro programa acaba funcionando en una máquina que tenga 95.

Sección Scroll Bar

Se debe de habilitar el desplazamiento Vertical al menos y Horizontal si se quiere, para los cuadros de edición multilínea. En los de una sola línea no tiene sentido pero hay otro truco interesante que podemos hacer añadiendo un desplazamiento vertical a un cuadro de edición de una sola línea, conseguir un contador

o SPINNER, de forma que aparezca un GET con una flecha arriba y abajo que sirvan por ejemplo para incrementar o disminuir el número de copias de una factura.



Case insensitive se podrá escribir dentro tanto en mayúsculas como en minúsculas desde nuestro programa.

Upper case convierte en mayúsculas todo lo introducido en el campo, muy conveniente para rellenar campos de bases de datos para que guarden uniformidad.

Lower case convierte en minúsculas todo lo introducido en el campo.

Single line o **Multiple line** indican si el control es de una línea o multilínea para editar un campo memo.

Password seleccionando esta opción todo lo que se teclee en el cuadro de edición aparecerá escrito con asteriscos *****, aunque el valor que devolverá será el que realmente se haya tecleado.

Convert OEM Convierte a código OEM (Windows) lo que vayamos tecleando. No es necesario FiveWin ya lo toma como OEM.

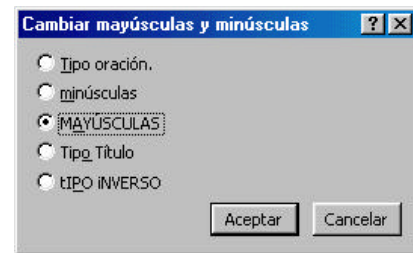
Keep selection permite efectuar selección de texto para copiar, cortar o pegar con el botón derecho del ratón. No es necesario ya que FiveWin lo activa por defecto.



Read Only Indican si el contenido del control será de solo lectura (Read Only) con lo cual no se podrá entrar a escribir en él y se comportará como un SAY enmarcado. NO funciona si se marca aquí ya que para conseguir esto hay que indicar desde dentro de nuestro fuente FiveWin que el GET será READONLY.


Want return, o por el contrario si su contenido será editable y será devuelto con su valor (Want Return). No es necesario por que FiveWin siempre nos devolverá el valor de cada GET del diálogo.

Control RADIOBUTTON (Botón de radio)

Los radio botones son un conjunto de opciones mostrado mediante círculos de los que solo se puede seleccionar uno ya que son opciones mutuamente excluyentes, pero una de ellas tiene que mostrarse seleccionada por defecto y al salir una tiene que estar obligatoriamente seleccionada.

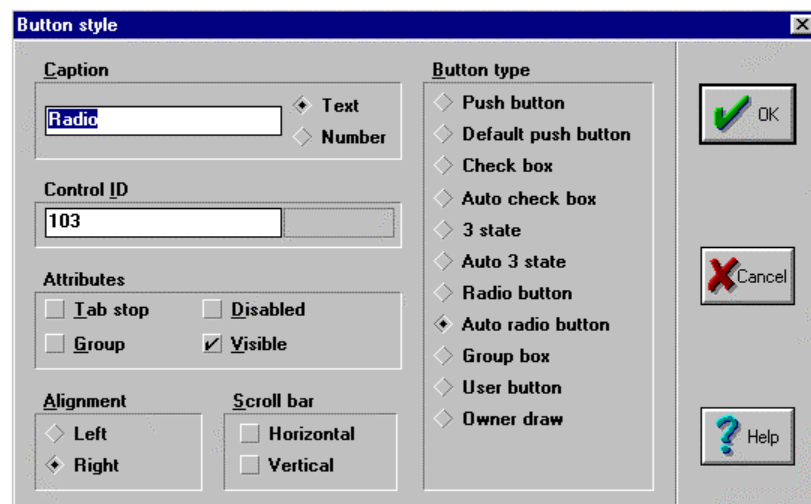


Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control de Windows  o el de Borland .

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un círculo acompañado de un texto, siendo su borde superior izquierdo el punto donde hemos hecho clic.

Pero solo aparecerá un círculo y lógicamente un conjunto de opciones ha de tener más de uno, por lo tanto habrá que repetir la operación tantas veces como opciones queramos mostrar. Los Radiobutton son agrupadas automáticamente como un conjunto de opciones único si se sitúan uno encima de otro o de izquierda a derecha en la misma línea. Ojo si se quieren hacer dos grupos distintos hay que ponerlos un poco apartados para que no se agrupen sin querer, o hacerlos muy separados y luego aproximarlos.

Pulsando después doble clic sobre cada uno de ellos entraremos en sus propiedades apareciendo una ventana similar a la descrita en casos anteriores donde veremos el Caption que podemos cambiar para que aparezca otro texto en el botón, el ID automáticamente asignado.



Caption Texto que aparecerá junto al Radiobotón.



ID automáticamente asignado pero como siempre modificable.



Justification para situar el texto a la Izquierda (Left) o a la derecha (Right opción por defecto) del círculo.

Radio Button y **Auto Radio Button** La primera se utiliza para que solo se pueda modificar el radio botón desde programa. La segunda para que el usuario lo pueda modificar siendo esta la opción por defecto.

Control CHECK BOX (Caja de selección)

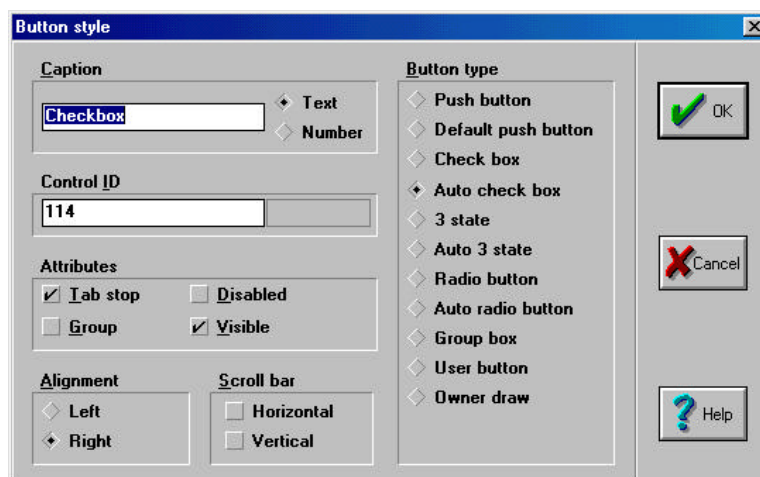
Los check box son cuadros donde podemos activar o desactivar una opción o un conjunto de opciones mostrado mediante varios cuadros de los que al contrario que en los Radio Botones se puede seleccionar solo uno, varios o incluso ninguno.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control de Windows  o el de Borland .

Ahora el cursor cambia de ser una flecha a otro de forma   lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadrado con una marca acompañado de un texto, siendo su borde superior izquierdo el punto donde hemos hecho clic.

Pero solo aparecerá un cuadrado y si se quiere hacer un conjunto de ellos habrá que repetir la operación tantas veces como opciones queramos mostrar.

Pulsando después doble clic sobre cada uno de ellos entraremos en sus propiedades apareciendo una ventana similar a la descrita en casos anteriores donde veremos el Caption que podemos cambiar para que aparezca otro texto en el botón, el ID automáticamente asignado.



Caption Texto que aparecerá junto al Check Box.

ID automáticamente asignado pero como siempre modificable.

Justification para situar el texto a la Izquierda (Left) o a la derecha (Right opción por defecto) del cuadro.


Check box y **Auto Check box** La primera se utiliza para que solo se pueda modificar el Check Box desde programa. La segunda para que el usuario lo pueda modificar siendo esta la opción por defecto.

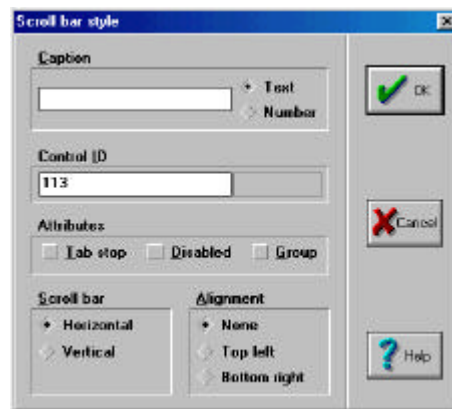
3 State y **Auto 3 State**. Habilita una tercera opción distinta a las dos lógicas de seleccionado o no seleccionado, la tercera opción es no seleccionable y se presenta como un cuadro de fondo gris pero FiveWin solo puede controlar las dos lógicas activado .T. y desactivado .F. y por lo tanto no nos sirve para nada.

Control SCROLL BAR (Barras de Scroll o de Desplazamiento)

Las barras de scroll o de desplazamiento son un control que permite desplazar el contenido de otro control o de una ventana para poder ver el resto del contenido cuando no cabe dentro de dicho control o ventana.

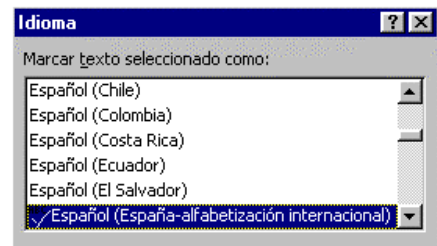
Hay dos tipos de barras  Horizontal Scroll Bar  Vertical Scroll Bar y no las hay de estilo Borland.


Tenemos que elegir uno de los iconos de la barra de herramientas, nuestro cursor cambia de forma  y como siempre tendremos que llevarlo sobre la caja de diálogo. Hacemos doble clic y aparece nuestra barra de scroll.




Control LISTBOX (Caja de lista de elementos)

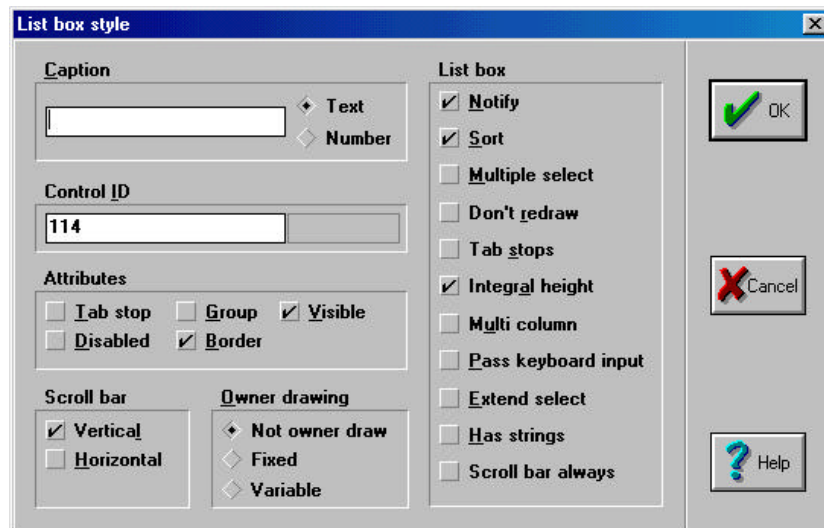
Un Listbox es un rectángulo con una barra de scroll vertical que se utiliza para mostrar un conjunto de elementos y nos da la posibilidad de seleccionar uno de ellos que queda marcado en azul. En FiveWin además se pueden mostrar listas de opciones con gráficos BMP.



Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control .

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadrado, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:



Caption en este caso no sirve de nada puesto que el texto que pongamos no aparecerá en el control.

ID automáticamente asignado pero como siempre modificable.

Scroll Bar el scroll vertical viene ya puesto como es lógico y aparecerá si la lista es más larga que el listbox pero podemos activar también que el horizontal aparezca si el contenido es más ancho que el listbox.

Owner Drawing sirve para especificar ciertas opciones de dibujo **Not owner draw** es la opción por defecto e indica que el listbox se encarga de dibujar la lista. **Fixed** lo utilizaremos cuando vayamos a incluir bitmaps junto a las opciones del Listbox. **Variable** (no sirve desde FiveWin) permite que el tamaño vertical de los elementos sea variable.

Notify activado por defecto sirve para enviar un mensaje al programa cuando se elija una opción del listbox que FiveWin puede controlar desde el ON CHANGE del listbox.

Sort muestra el contenido del listbox ordenado alfabéticamente por defecto, desactivar si se desea que aparezcan en el orden que nosotros le demos desde el programa.

Multiple Select Podremos elegir más de una opción simultáneamente, aunque en FiveWin no se usa y si queremos que sea así hay que indicarlo desde el código fuente como MULTISEL.

Integral Height La última opción de la lista no se pintará cortada si no cabe.

Multi Column Mostrar listbox de varias columnas de datos. No funciona desde FiveWin.

Pass Keyboard Input No funciona la selección de la lista por teclado.

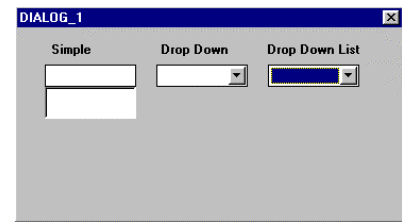
Extend Select Para hacer selección múltiple mediante clic y arrastrando el ratón.


Has strings Para hacer listbox de Radiobuttons. No funciona desde FiveWin.


Scrollbar always Muestra la barra de scroll siempre aunque no haga falta.

Control COMBOBOX (Caja de lista de elementos combinada con un cuadro de edición)

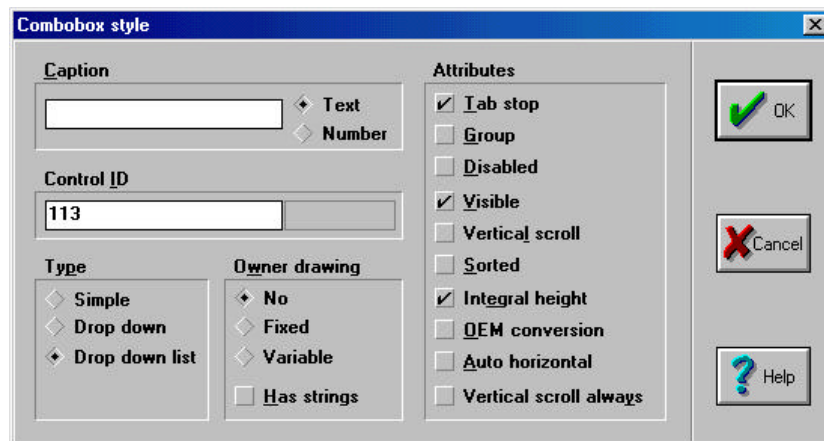
Un Combobox es un control que combina un Listbox y un get, es una lista de opciones entre las que podemos seleccionar una y a la vez es un cuadro de edición donde podemos escribir texto.



Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control .

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadrado, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:



Caption en este caso no sirve de nada puesto que el texto que pongamos no aparecerá en el control.

ID automáticamente asignado pero como siempre modificable.

Sección Type (tipo de Combobox)

- **Simple** estará siempre abierto mostrando la lista de opciones como un Listbox pero permite escribir sobre el cuadro de edición superior. El usuario podrá entrar en la lista y editarla.
- **Drop Down** se abrirá como en todos los programas de Windows pulsando sobre el botón del triángulo o con F4 desde el teclado mostrando la lista de opciones y permite escribir sobre el cuadro de edición superior para buscar el elemento a seleccionar por aproximación. El usuario podrá entrar en la lista y editarla.
- **Drop Down List** es la opción por defecto y la más lógica. Funciona igual que la anterior, permite buscar y seleccionar pero no se podrá editar la lista de opciones y así el usuario no podrá más que seleccionar una de ellas.

Auto horizontal Busca automáticamente en la lista al teclear un texto en el cuadro de edición.


Una opción interesante de los combobox es que además podemos usarlo para mostrar un determinado campo de una base de datos a través de DBCOMBO que nos permite por ejemplo al dar de alta un cliente mostrar un combobox con las formas de pago del fichero de formas de pago y permitirnos tomar una de ellas para almacenar este campo u otro de las formas de pago en la ficha del cliente.


Para lograr esto simplemente hacemos un combobox normal y lo redefinimos como un DBCOMBO desde el código fuente de nuestro programa en FiveWin añadiéndole entonces los datos necesarios para su funcionamiento, de que fichero DBF debe de tomar los datos cual es el campo a mostrar, etc.

Control TWBROWSE (Control de FiveWin)

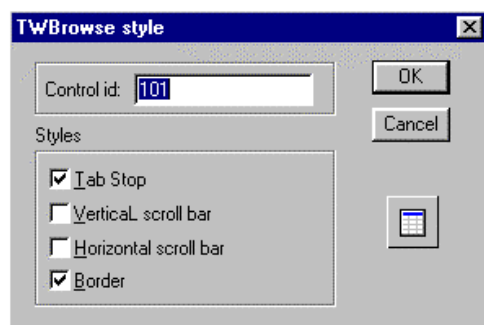
Vamos a tratar ahora sobre el conjunto de controles propios de FiveWin y que han sido añadidos a Resource Workshop de la forma ya tratada en páginas anteriores.

El primero de ellos TWBROWSE nos será de gran utilidad ya que es la manera más sencilla de representar datos de archivos DBF en pantalla en forma de lista encolumnada, es decir un BROWSE o GRID (rejilla).

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro con celdas, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:



Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, poner una barra de Scroll Vertical u Horizontal muy convenientes si los datos a presentar son muchos y no caben en el control y añadir o quitar un borde al control.


Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.


Control TCBROWSE (Control de FiveWin)

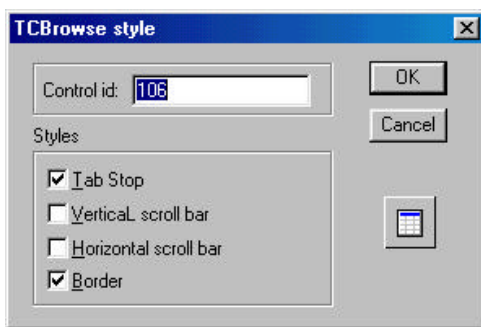
El control llamado TCBROWSE tiene en principio la misma finalidad que el anterior TWBROWSE es otra manera de representar datos de archivos DBF en pantalla en forma de lista encolumnada, es decir un BROWSE o GRID.

La diferencia ente ambas es que la primera TWBrowse es sensiblemente más rápida ya que cuando se crea evalua un bloque de código por cada línea y la segunda evalua un bloque de código por cada línea y otro por cada columna de cada línea. Además la primera tiene su código de dibujo en pantalla PAINT() en puro C++ lo que la hace super veloz y la segunda no es así, resultando más lenta.

Por lo tanto TWBrowse se usa como una tabla simple y rápida pero TCBrowse permite en principio un mejor control del aspecto visual y del contenido de las celdas y el movimiento entre ellas.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro con celdas, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.



Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo en un diálogo muy parecido al anterior:


Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, poner una barra de Scroll Vertical u Horizontal muy convenientes si los datos a presentar son muchos y no caben en el control y añadir o quitar un borde al control.


Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

Control TBITMAP (Control de FiveWin)

Se utiliza para poner un gráfico dentro del diálogo pudiendo ser de casi cualquiera de los tipos gráficos habituales, BMP, JPG, GIF, etc.

Incluso se puede presentar en el un Bitmap animado compuesto de una secuencia de varios gráficos en base a un TIMER de FiveWin.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo una imagen gráfica, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:





Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.

Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

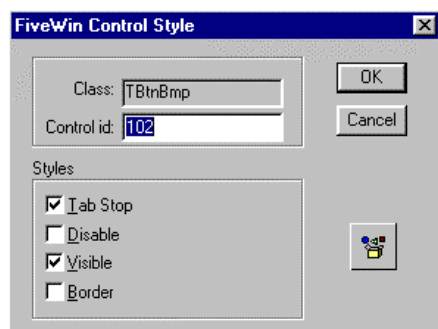
Control TBTNBMP (Control de FiveWin)

Se utiliza para poner botones sobre los que pulsar pero con un gráfico BMP en su interior en vez de con un texto.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro vacío pero con el aspecto sobreelevado de un botón, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:





Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.

Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

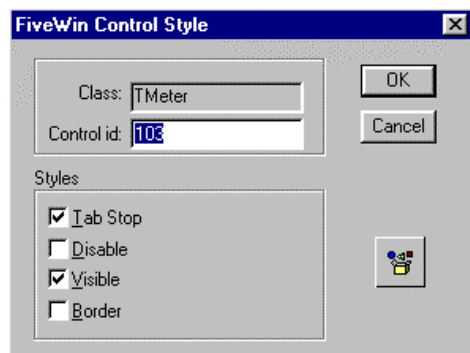
Control TMETER (Control de FiveWin)

Se utiliza para presentar un progreso dentro de una tarea que va a llevar un tiempo largo de proceso e indica el porcentaje ya efectuado de la tarea mediante una barra de progreso y un X %, pudiendo añadir un texto al porcentaje si se desea.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un rectángulo con un porcentaje 0%, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

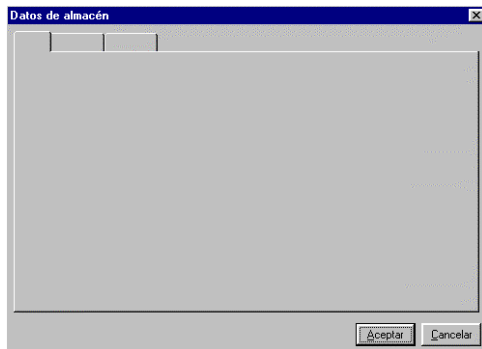
Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:



Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.

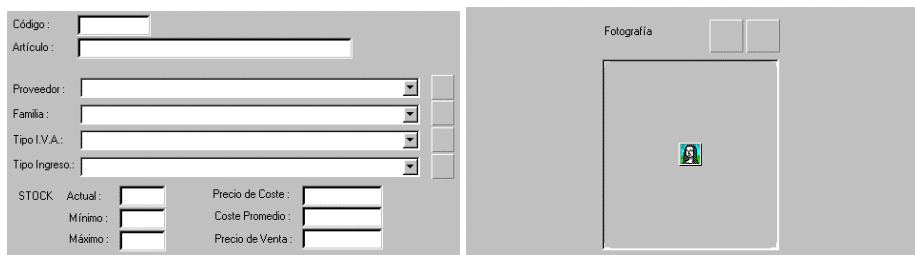
Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.


Control TFOLDER (Control de FiveWin)




Se usa para construir un diálogo contenedor de separadores FOLDER que en realidad es un diálogo que muestra unas pestañas en su parte superior y que pulsando en cada una de ellas nos mostrará un distinto contenido.

Cuando se usan FOLDER el diálogo que contiene a estos no será más que un contenedor vacío, es decir, los contenidos de cada FOLDER no se encuentran dentro de este diálogo sino que hay que diseñarlos aparte en otros diálogos distintos, uno por fólder y estos serán de tipo Child y estilo No border como en los ejemplos siguientes, para que al aparecer después encima de los FOLDER se fundan dentro de ellos



Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro con los FOLDER en su parte superior, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

En principio se muestra como si tuviera tres FOLDER posibles pero podemos poner tantos como queramos desde dos hasta los que nos quepan en el diálogo.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:

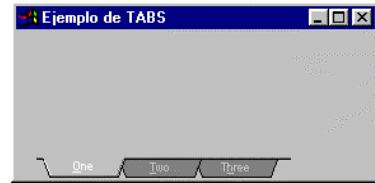



Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.


Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

Control T TABS (Control de FiveWin)

Es un control parecido al anterior, muestra una serie de pestañas pero en este caso en la parte inferior del diálogo y se usan para desencadenar una acción en el caso de pulsar sobre un determinado TAB. Por ejemplo si nos diseñamos un diálogo con un listín telefónico y ponemos un tab para cada letra inicial A B C D E F ... podríamos hacer que al pulsar sobre una letra solo se mostraran los registros que comenzaran por dicha letra filtrando la base de datos.

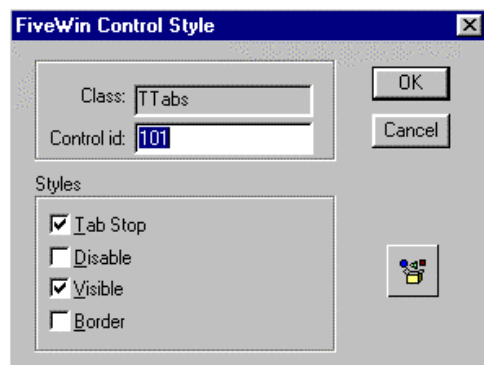


Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro con unas pestañas inferiores, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

En principio se muestra como si tuviera tres TAB posibles pero como en el caso del FOLDER podemos poner tantos como queramos desde dos hasta los que nos quepan en el diálogo.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:




Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.

Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

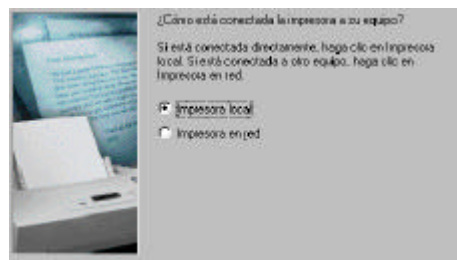
Control TPAGES (Control de FiveWin)



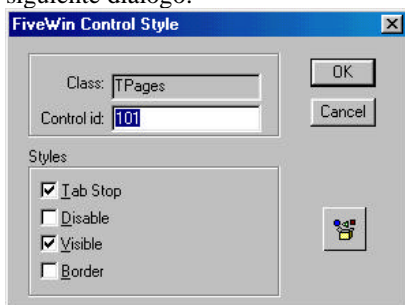
Es un control parecido a los dos anteriores en realidad sirve para montar más de un panel sobre el mismo diálogo como los FÓLDER para que cada uno tenga un distinto contenido, pero en este caso no se muestran una serie de pestañas en la parte superior o inferior del diálogo, sino que el movimiento de una a otra se hace mediante botones de adelante y atrás. Un ejemplo típico de PAGES son los programas de instalación de cualquier programa de Windows o los asistentes del mismo Windows.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control  y haciendo clic sobre el diálogo arrastramos el cursor hasta abrir un espacio suficiente para contener después las PAGES.

Después tendremos que construir las PAGES por separado igual que los Fólder que se montan después sobre este diálogo en tiempo de ejecución desde nuestro programa. Hay que diseñarlos aparte en otros diálogos distintos, uno por Page y estos serán de tipo Child y estilo No border como en los ejemplos siguientes, para que al aparecer después encima de los PAGES se fundan dentro de ellos.



Pulsando después doble clic sobre el control TPAGES entraremos en sus propiedades apareciendo el siguiente diálogo:



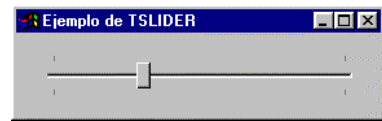
Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.


Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.


Por supuesto al diálogo que vaya a tener PAGES habrá que añadir obligatoriamente los botones de Atrás y Adelante como mínimo ya que sin ellos no se podrá avanzar o retroceder entre las distintas PAGES.

Control TSLIDER (Control de FiveWin)

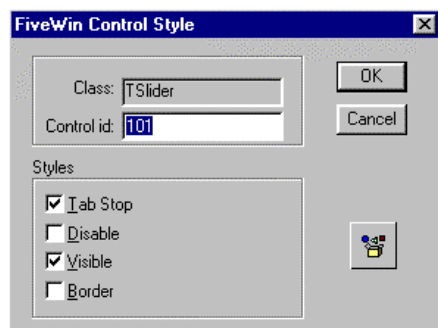
Los TSLIDER son un control del genial Ramón Avendaño y son una barra de volumen verticales u horizontales que puede mover el usuario para manipular una cantidad numérica que podemos capturar desde el programa.



Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro con un volumen horizontal, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:





Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.

Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estamos desarrollando.

Control TSPLITTER (Control de FiveWin)

También gracias a la imaginación y trabajo de Ramón Avendaño nos encontramos con esta clase que simplemente nos permite pintar rayas verticales u horizontales para mejorar el aspecto de nuestros diálogos separando zonas de datos de los mismos.

Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo una línea horizontal, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Si queremos una línea vertical bastará con tirar del borde de una horizontal, haciéndola más alta y más estrecha.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:




Poco podremos cambiar en este tipo de control, ni siquiera dispone de TabStop ya que este control es decorativo y no hay que parar en él para nada, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.


Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

Control TSELECTOR (Control de FiveWin)

Se deben también a Ramón Avendaño y son muy parecidos a los anteriores ya que tienen el mismo cometido. Son controles tipo volumen de sonido que el usuario podrá mover devolviendo al programa una cantidad numérica.



Para añadir al diálogo un control de este tipo elegiremos en la barra TOOLS el control 

Ahora el cursor cambia de ser una flecha a otro de forma  lo situaremos sobre la caja de diálogo que estamos diseñando y haremos clic izquierdo del ratón. Aparecerá en la caja de diálogo un cuadro que contiene un círculo, siendo su borde superior izquierdo el punto donde hemos hecho clic y a continuación podemos ajustar su tamaño y moverlo con el ratón.

Pulsando después doble clic sobre el control entraremos en sus propiedades apareciendo el siguiente diálogo:



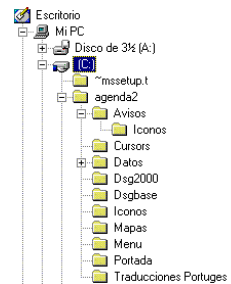
Poco podremos cambiar en este tipo de control, indicar si queremos TabStop, Disable si queremos desactivarlo o no, Visible si se quiere que aparezca mostrado u oculto y añadir o quitar un borde al control.

Será más que suficiente ya que el resto de las particularidades del control las definiremos desde el programa que estemos desarrollando.

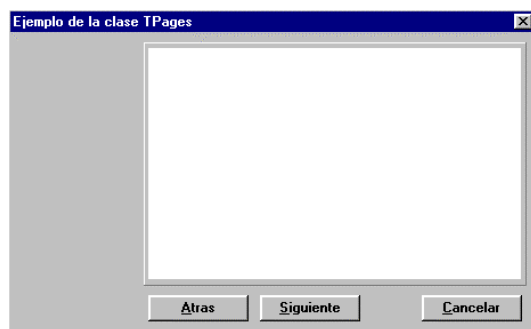
Controles definidos por el usuario.

No todos los controles posibles están contemplados por Resource Workshop ni aún añadiendo la librería FWCTRLS.DLL con los controles propios de FiveWin. Cada vez hay más controles que podremos añadir a nuestros programas para dotarlos de un aspecto y manejo más lejos de los controles habituales que hemos visto y hay una gran cantidad de ellos que se pueden comprar por separado como los magníficos controles de Francisco García 5win que harán alucinar a los más caprichosos. Muchos otros son de libre distribución y se encuentran por Internet o en el foro de FiveWin y alguno más de los hasta ahora vistos viene incluso incluido en FiveWin.

Por ejemplo un control vistoso y que puede ser útil en determinadas aplicaciones es el TTREEVIEW que nos permite presentar estructuras de tipo árbol para presentar datos de forma jerárquica tal y como lo hace por ejemplo el explorador de Windows. Se indican las líneas a mostrar por el programa, se le asigna a cada una de ellas dos bitmap a mostrar a su izquierda uno para cuando está cerrado y otro para cuando está abierto y ya se puede mostrar el árbol cliqueando para abrir y cerrar sus ramas.



Otro ejemplo de control que está incorporado en FiveWin y que no podemos definir directamente desde Resource Workshop es TPAGES que se utiliza para crear los asistentes de los programas, esos programas que nos llevan paso a paso pulsando sobre el botón Siguiente hasta lograr un objetivo y donde podemos ir atrás al paso anterior.




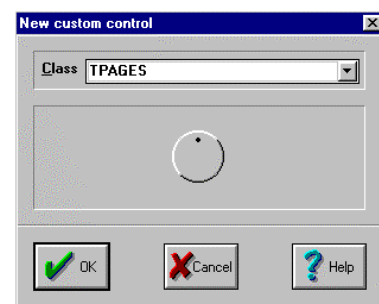
El control funcionará después de una forma similar a TFOLDER ya que el control TPAGES será un simple contenedor y cada relleno será un diálogo sin bordes que se montará sobre el control TPAGES.

Pero como podemos desde Resource Workshop añadir un control de este tipo u otro que no exista dentro de los TOOLS a un diálogo. Pues nos lo inventamos.

En el fondo todos los controles son iguales, ocupan un área, tienen unas propiedades casi similares y lo que hagan no depende de Resource Workshop sino de nuestro programa, por lo que podemos dibujar el control que nos de la gana y luego desde programa dotarlo de las características que en definitiva van a determinar que control es realmente. Es decir si dibujamos un control TBUTTON y desde el programa lo redefinimos como un BITMAP también “funciona”.

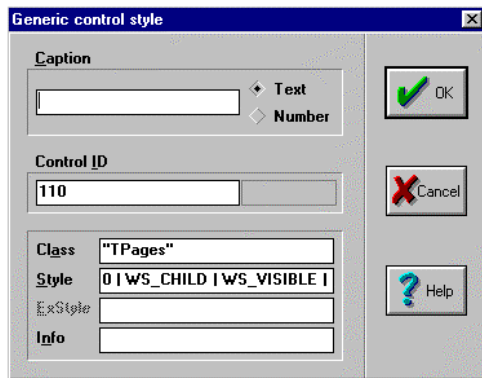
Vamos a ver por lo tanto como jugársela a Resource Workshop

definiendo un control nuevo. Elegimos el icono  y aparece una caja de diálogo con un combobox para elegir un control de entre los controles que ya hemos visto hasta ahora, pero en lugar de elegir uno de ellos lo que hacemos es teclear el nombre de la clase que usaremos sobre el control, en este caso la clase necesaria para este último control es la TPAGES que será lo que teclearemos en el combobox.



Pulsando INTRO aparece un cursor de cruz que nos permite abrir un cuadro que servirá para indicar el área que ocupa el nuevo control.

Cuando redefinamos el control desde el código de nuestro programa, FiveWin se encargará de sustituir este hueco que hemos dejado en el diálogo con el aspecto y las características de la clase requerida.



A continuación tenemos que hacer doble clic sobre el control que acabamos de pintar y veremos que es un control de tipo Generic control style de clase Tpages y lo único que tenemos que hacer es editar el estilo correcto para cada control que en el caso de este será

`0 | WS_CHILD | WS_VISIBLE | WS_TABSTOP`

que como puede verse indica que tiene TabStop que es Visible y que es Hija del diálogo. Estas características son propias de cada control y habrá que consultar los ejemplos o la documentación de FiveWin o el código fuente de la clase para poder poner la correcta. El control ya está listo para ser utilizado desde nuestro programa.

Otra forma que tenemos de incluir un control nuevo es partiendo de un control existente de formato parecido. Por ejemplo para situar un TTREEVIEW podemos dibujar un control de tipo LISTBOX y después entrar a editar como texto el diálogo desde *Resource/Edit as text* y quitando el estilo estándar del listbox `LBS_STANDARD` poner el siguiente estilo `LBS_NOTIFY | WS_BORDER | LBS_NOREDRAW | LBS_OWNERDRAWVARIABLE | LBS_NOINTEGRALHEIGHT | WS_BORDER | WS_VSCROLL | WS_HSCROLL | WS_TABSTOP` (el TabStop es optativo).

El control ya está listo para ser utilizado desde nuestro programa como un TTREEVIEW.

Recursos definidos por el usuario

Hemos visto como crear nuevos controles que situar sobre un recurso de tipo diálogo, pero también desde Resource Workshop podemos crear nuevos tipos de recursos para usar directamente desde nuestro programa.

Pero si hemos visto que los menús, aceleradores, ventana principal u otras ventanas se crean desde FiveWin sin usar recursos y los demás recursos como diálogos, iconos, cursores, bitmaps, etc se crean desde Resource Workshop y ya están definidos, que otros recursos extra podrán manejar nuestros programas que podamos almacenar en un proyecto de Resource Workshop.

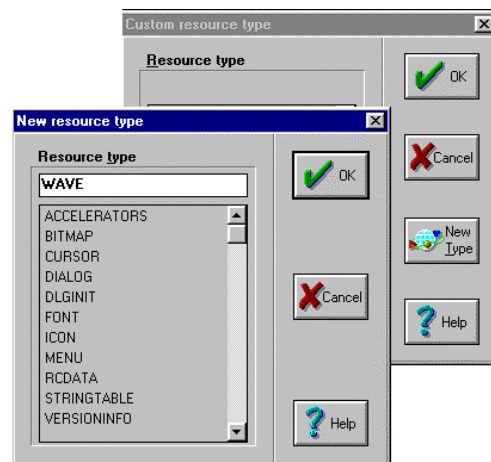
En principio el único tipo de recurso extra que se me ocurre es el de tipo WAV que son los archivos de sonido que nuestro programa hará sonar por los altavoces y la opción de almacenarlos en el proyecto como un recurso más puede ser interesante para que los archivos WAV no queden desperdigados por la carpeta del programa y sean accesibles para cualquier usuario listillo.

Pues dado que este tipo de recurso no existe en Resource Workshop habrá que inventarlo y para ello tendremos que situarnos en la pantalla principal del gestor del proyecto e indicar como hasta ahora que queremos un añadir un recurso nuevo al proyecto *File/Add to project* de forma que se nos mostrará una lista *File type* con los tipos de recursos hasta ahora vistos donde evidentemente no encontraremos uno de tipo WAV. Por lo tanto habrá que indicar el tipo *User Resource Data*, a continuación en la lista de ficheros buscar el WAV que queramos incorporar al proyecto seleccionarlo y pulsar sobre OK.

Con esto aparecerá la siguiente ventana donde al elegir la opción *New Type* nos mostrará una lista con los tipos de recursos habituales donde no encontraremos el tipo WAV y por lo tanto deberemos de introducir a mano el tipo pero indicando WAVE para que funcione correctamente. Pulsamos sobre OK y a continuación ya hay en la lista un nuevo tipo y dentro del proyecto el recurso que almacenará el sonido que hemos tomado desde el archivo WAV del disco.

Después desde el programa FiveWin y con la DLL cargada se podrá ejecutar este recurso y por lo tanto oír su sonido por los altavoces con la instrucción

SNDPLAYRES(Nombre_Recurso_Wave)



Como usar los recursos desde FiveWin

Como activar la DLL para su uso.

Usar una DLL desde FiveWin es muy sencillo, básicamente habrá que incorporar al programa un par de líneas una al principio donde indicar que haga uso de la DLL y otra al final para decirle que deje de utilizar. Es parecido a usar un archivo de datos DBF se abre cuando hace falta y se cierra al salir.

Primeramente hay que decir que hay tres tipos distintos de DLL, el primero es aquel que contiene únicamente recursos, como el que podemos diseñar nosotros desde Resource Workshop con los diálogos y demás recursos de nuestro programa, el segundo es aquel que contiene únicamente código ejecutable y no componentes visuales del programa, nosotros no las podemos hacer desde Resource Workshop para FiveWin pero hay muchas que se pueden encontrar en el disco duro y que si se intentan abrir con Resource Workshop no nos enseñarán ningún recurso pero en realidad tienen funciones ejecutables de los programas, y un tercer tipo sería una DLL mixta que combinaría componentes visuales y código ejecutable. Nosotros solo diseñaremos del primer tipo de DLL pero puede que nos haga falta utilizar alguna DLL de los otros dos tipos.

Esto es importante ya que las primeras se activan desde FiveWin con una instrucción y las otras con otra instrucción distinta.

Las primeras, es decir las diseñadas por nuestra mano con diálogos, bitmaps, etc. se activan desde FiveWin con la instrucción:

```
SET RESOURCES TO nombre_del_proyecto.DLL
```

Y para dejar de utilizarlas con

```
SET RESOURCES TO
```

Para los dos últimos tipos con código ejecutable y/o recursos gráficos, pero también si se quiere para el anterior diseñado por nosotros solo con recursos gráficos, se utiliza la siguiente instrucción:

```
hDLL:=LOADLIBRARY("nombre_del_proyecto.DLL")
```

Donde hDLL es un número de controlador devuelto por la función, y se dejan de usar con

```
FreeLibrary(hDLL)
```

Es muy importante y se debe de insistir en que al igual que se abre la DLL normalmente al inicio del programa, hay que cerrar la DLL antes de terminar por que si no se queda "pillada en memoria" y además de quitar recursos a Windows, si estamos programando y queremos hacer alguna modificación a la DLL no podremos grabarla ya que según Windows esta en uso ya que no la hemos liberado.

Otro detalle muy importante tiene mucho que ver con los controles de tipo Borland. Si usamos cualquiera de estos controles habrá que incluir obligatoriamente una DLL propia de Borland llamada BWCC.DLL que viene incluida con Resource Workshop y que se deberá de distribuir junto con nuestro programa. Es una DLL del tercer tipo ya que incluye recursos y código ejecutable. Pero ya hemos comentado como podríamos tener un problema con estos controles ya que al incluir esta DLL podría haber otra BWCC.DLL dentro de la misma máquina introducida por otro programa distinto, esto no causaría mayor problema si la DLL casualmente fuera de la misma versión, pero en caso de no serlo podríamos tener graves problemas de

funcionamiento tanto dentro de nuestro programa como quizás en otros que la usen. Para evitarlo existen dos opciones, la primera consistirá en llamar a la DLL con otro nombre (por ejemplo MIBWCC.DLL) y usar este nombre en el programa y la segunda opción consistirá simplemente en no usar los controles de Borland.

Uso de varias DLL simultáneas.

Hasta aquí lo expuesto para usar una DLL es muy simple, pero existe un problema al querer usar varias DLL simultáneas ya que al usar SET RESOURCES TO PROYECTO.DLL activamos para su uso la DLL llamada PROYECTO y podemos usar sus recursos gráficos, pero si más tarde queremos usar otra DLL y para ello usamos SET RESOURCES TO OTRA.DLL y activamos una DLL llamada OTRA podremos usar sus recursos sin problemas pero entonces ya queda olvidada la primera DLL llamada PROYECTO y sus recursos por lo que tendremos que volver a usar SET RESOURCES TO PROYECTO.DLL y si olvidamos hacerlo nos dará un fallo importante deteniendo la ejecución de nuestro programa.

Si tuviéramos cuidado en activar cada DLL cada vez que haga falta esto no sería un problema pero al más mínimo descuido se nos iría el programa al traste.

Las posibles soluciones son tres, la primera sería introducir todos los recursos en una sola DLL y problema terminado, pero si aun así queremos usar más de una DLL y a veces podemos vernos forzados a ello por querer usar conjuntamente una DLL nuestra y otra de terceros como por ejemplo NVIEWLIB.DLL para trabajar con gráficos, tendremos que trabajar de la siguiente manera.

```
static hDLL, hOTRADLL //Definir los manejadores donde se van a cargar las DLL

function main() //Al principio del programa principal

hDLL := LoadLibrary( "PROYECTO.DLL" ) // Cargar la primera DLL
hOTRADLL := LoadLibrary( "OTRA.DLL" ) // Cargar la segunda DLL

DEFINE WINDOW oWnd
.....
.....
.....
.....

ACTIVATE WINDOW oWnd;
VALID ( FreeLibrary( hDLL ) , FreeLibrary( hOTRADLL ) , .T. )

//Al cerrar la ventana principal que libere las 2 DLL

return nil

function GethDll()
return ( hDLL )

function GethOTRADLL()
return ( hOTRADLL )
```


Dentro del programa y en otro sitio cualquiera

```
Function ejemplo()  
  
LOCAL oDlg // definimos una caja de dialogo que se encuentra en la primera DLL  
  
DEFINE DIALOG oDlg RESOURCE "DLG01" TITLE "Diálogo de DLL 1" DLL GethDLL()  
.....  
.....  
.....  
.....  
  
ACTIVATE DIALOG oDlg  
  
return nil
```

Dentro del programa y en otro sitio cualquiera

```
Function Ejemplo2()  
  
LOCAL oDlg // definimos una caja de dialogo que se encuentra en la segunda DLL  
  
DEFINE DIALOG oDlg RESOURCE "DLG02" TITLE "Diálogo de DLL 2" DLL GethOTRADLL()  
  
ACTIVATE DIALOG oDlg  
  
return nil
```

Se puede usar esta fórmula para dos o más DLL e incluso a la vez usar otras DLL con el comando SET RESOURCES TO sin perder para nada las que tengamos abiertas de esta manera.

Una tercera forma sería al tener que usar una segunda DLL capturar la que este en uso hasta el momento

```
hResOld := GetResources()
```

usar la segunda DLL con SET RESOURCES TO y recuperar de nuevo la DLL anterior con

```
SetResources( hResOld )
```

De esta manera es más fácil controlar a que DLL se tiene que volver después de usar una nueva temporalmente.

Como usar los recursos de una DLL desde FiveWin.

Hasta ahora hemos visto la primera parte de este embrollo, esto es, como dibujar esos diálogos con sus controles que queremos después incluir en nuestro programa y como darle un toque profesional, pero veamos ahora como hacer que esos recursos echen a andar desde nuestro programa FiveWin.

En el tema anterior hemos visto como activar una DLL desde nuestro programa, pues bien una vez hecho esto ya tenemos a nuestra disposición todos sus recursos y solo nos queda elegir en nuestro código cual queremos presentar en pantalla para que lo muestre en pantalla al usuario.

No vamos a entrar ahora a definir como usar cada uno de ellos al detalle ya que para esto ya hay otro buen número de manuales en FiveWin y también disponemos de la ayuda que viene incorporada en el programa y de sus ejemplos, pero en líneas generales podemos decir que una vez abierta una DLL y cuando queremos usar un recurso hay que indicar a FiveWin que lo use desde la DLL abierta mediante cláusulas de tipo XBASE

Cláusula DEFINE.

Toma el recurso de la DLL mediante su nombre y crea el objeto del tipo que indiquemos. Todos los objetos que mostramos en pantalla, WINDOW, DIALOG, BITMAP, etc. primero deben de ser definidos mediante el correspondiente DEFINE.

```
DEFINE Tipo_objeto Nombre_objeto Nombre_recurso [Opciones]
```

Por ejemplo para crear un objeto de tipo ventana habrá que definirla

```
DEFINE WINDOW oWND
```

Y para crear un objeto de tipo diálogo:

```
DEFINE DIALOG oDlg RESOURCE "CLIENTES" OF Ownd
```

RESOURCE indica cual es el nombre del recurso dentro de la DLL abierta y OF indica a quien pertenece este diálogo. En Windows todos los objetos deben de pertenecer a otro objeto, un diálogo pertenece a una ventana que lo contiene y la ventana principal del programa también pertenecerá al escritorio de Windows, aunque esto como se ve no hace falta indicarlo.

Cláusula ACTIVATE.

Pero DEFINE no muestra el objeto en pantalla solo lo genera en memoria y lo deja listo para utilizar y el objeto solo aparecerá cuando lo ordene el programa mediante la clausula ACTIVATE.

```
ACTIVATE Tipo_objeto Nombre_objeto [Opciones]
```

Por ejemplo para activar la ventana antes definida habría que utilizar:

```
ACTIVATE WINDOW oWND
```

Y para activar el diálogo

```
ACTIVATE DIALOG oDLG
```

Cláusula :END()

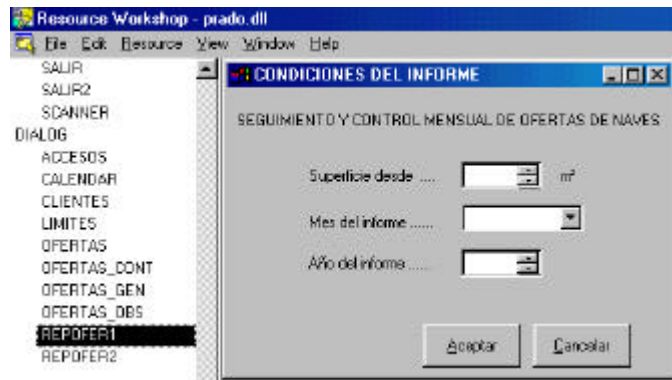
Con esto tendríamos ya en marcha el objeto y lo veríamos en pantalla y podríamos trabajar sobre el, después de hacerlo solo nos queda por ver como eliminarlo de la pantalla y de la memoria t para eso tenemos en método `NOMBRE_OBJETO:END()`

Por ejemplo para finalizar la ejecución de una ventana es necesario indicar `oWnd:End()`
O para finalizar la ejecución de un diálogo habrá que indicar `oDlg:End()`

Trabajando dentro de un diálogo, redefinición de los controles.

Con lo que hemos visto podremos presentar uno de los diálogos que haya en la DLL en pantalla y aparecería con todos sus controles ya definidos puesto que los hemos definido en Resource Workshop, incluso funcionarían básicamente pudiendo escribir sobre ellos y movernos de uno a otro, pero esto no es suficiente para trabajar ya que queremos controlar si lo que se escriba en ellos es correcto, capturarlo para grabarlo en un archivo del disco, etc. Por ello hay que redefinirlos para dotarlos de todas estas posibilidades mediante la clausula `REDEFINE`.

En el siguiente ejemplo se ver como un diálogo se define, se redefinen sus controles indicando que ID tienen asignado desde Resource Workshop y con `OF` que pertenecen a ese diálogo y después se activa la ventana. También se puede comprobar como el botón cancelar acciona el `:END()` del diálogo terminando su ejecución.



```
#include "fivewin.ch"
#include "ORD.ch"
```

```
***** SEGUIMIENTO Y CONTROL MENSUAL DE OFERTAS DE NAVES *****

FUNCTION INFOFERT1()
LOCAL oGET:=ARRAY(5),nMETROS2:=5000,oMESES,nMES:=MONTH( DATE()),nYEAR:=YEAR( DATE())
LOCAL aMESES:={"ENERO","FEBRERO","MARZO","ABRIL","MAYO","JUNIO","JULIO","AGOSTO",;
              "SEPTIEMBRE","OCTUBRE","NOVIEMBRE","DICIEMBRE"}

SET RESOURCES TO "PRADO.DLL"

DEFINE DIALOG oDLG RESOURCE "REPOFER1"

REDEFINE GET oGet[1] VAR nMETROS2 ID 101 OF oDLG PICTURE "@E 999,999" SPINNER ;
ON UP (nMETROS2:=nMETROS2+500,IF(oOFERTAS:RIESGO > 999500, nMETROS2 := 999500,);
,oGet[1]:REFRESH());
ON DOWN (nMETROS2:=nMETROS2-500,IF(nMETROS2 < 500, nMETROS2 := 500,);
,oGet[1]:REFRESH());
MIN 500 MAX 999500 VALID IF(nMETROS2>=500 .AND. nMETROS2<=999500,.T.,;
(MSGINFO("SOLO VALORES DE 500 A 999500","VALOR INCORRECTO"),.F.))

REDEFINE COMBOBOX oMESES VAR nMES ITEMS aMESES ID 102 OF oDLG

REDEFINE GET oGet[2] VAR nYEAR ID 103 OF oDLG PICTURE "@E 9,999" SPINNER ;
ON UP (nYEAR++,IF(nYEAR > 2025, nYEAR := 2025,),oGet[2]:REFRESH());
ON DOWN (nYEAR--,IF(nYEAR < 1997, nYEAR := 1997,),oGet[2]:REFRESH());
MIN 1997 MAX 2025 VALID IF(nYEAR>=1997 .AND. nYEAR<=2025,.T.,;
(MSGINFO("SOLO AÑOS DEL 1997 AL 2025","AÑO INCORRECTO"),.F.))

REDEFINE BUTTON oBtn1 ID 1 OF oDLG ACTION (LOK:=.T.,oDLG:End() )
oBtn1:cTOOLTIP="Comenzar impresión"+CRLF+"del informe"

REDEFINE BUTTON oBtn2 ID 2 OF oDLG ACTION oDlg:End() CANCEL
oBtn2:cTOOLTIP="Cancelar impresión"+CRLF+"del informe"
```

ACTIVATE_DIALOG ODLG

Índice del manual de Workshop.

QUE ES Y PARA QUE SIRVE RESOURCE WORKSHOP.....	1
POR QUE USAR RESOURCE WORKSHOP.....	2
PROYECTOS EN RESOURCE WORKSHOP, COMO CREAMLOS Y GESTIONARLOS.....	3
DLL (Librería de Enlace Dinámico)	3
RC (Recursos Compilado).....	4
QUE HAREMOS Y QUÉ NO HAREMOS CON EL RESOURCE WORKSHOP.....	5
¿Qué haremos desde Resource WorkShop?	5
¿Qué no haremos desde Resource WorkShop?	5
COMENZANDO A TRABAJAR CON RESOURCE WORKSHOP.	6
Creando un proyecto nuevo.	6
Incluyendo un recurso nuevo en un proyecto.	7
Añadiendo un Recurso existente a un Proyecto.....	7
Abrir un proyecto existente.	8
Grabar un proyecto.....	8
Uso de la ventana principal de gestión de Proyectos.....	9
Trabajando con los recursos.	9
Establecer preferencias de funcionamiento.....	10
TIPOS DE RECURSOS, COMO CREAMLOS Y GESTIONARLOS.....	11
Recurso de tipo Accelerator.....	11
Recurso de tipo Menu	11
Recurso de tipo Font	11
Recurso de tipo Bitmap.....	12
Recurso de tipo Cursor	13
Recurso de tipo Icon.....	13

Recurso de tipo Stringtable.....	14
Recurso de tipo Versioninfo.....	14
Recurso de tipo Dialog	15
Menú de herramientas TOOLS.....	17
Menú de herramientas ALIGNMENT.....	19
Dar una acabado profesional a nuestro diseño.....	21
Duplicar controles rápidamente.....	22
Ordenar el salto entre controles.....	22
Menú de propiedades PROPERTIES.....	23
Establecer preferencias de funcionamiento dentro de la edición de un diálogo	26
Control BUTTON (Botón de texto).....	27
Control STATIC TEXT (Texto estático SAY).....	30
Control EDIT TEXT (Texto editable GET).....	32
Control RADIOBUTTON (Botón de radio).....	34
Control CHECK BOX (Caja de selección).....	35
Control SCROLL BAR (Barras de Scroll o de Desplazamiento).....	36
Control LISTBOX (Caja de lista de elementos).....	37
Control COMBOBOX (Caja de lista de elementos combinada con un cuadro de edición).....	39
Control TWBROWSE (Control de FiveWin).....	41
Control TCBROWSE (Control de FiveWin).....	42
Control TBITMAP (Control de FiveWin).....	43
Control TSELECTOR (Control de FiveWin).....	51
Recursos definidos por el usuario	54
COMO USAR LOS RECURSOS DESDE FIVEWIN.....	55
Como activar la DLL para su uso.....	55
Como usar los recursos de una DLL desde FiveWin.....	58
Cláusula DEFINE.....	58
Cláusula ACTIVATE.....	58
Cláusula :END().....	58

Manual escrito por Bingen Ugaldebere
De Mungia Informática

bingen@muninser.com
web@muninser.com

Dedicado a los miembros del foro de FiveWin en Castellano a los que espero pueda ser útil y que lo disfruten.