

Contenidos

Guía de Referencia

Descripción del Producto.....	68
Que es FiveWin?.....	68
Compilando y Enlazando (Linking).....	68
Sus Primeras Aplicaciones.....	69
TUTOR01.PRG.....	69
TUTOR02.PRG.....	69
TUTOR03.PRG.....	69
TUTOR04.PRG.....	69
TUTOR05.PRG.....	69
TUTOR06.PRG.....	69
TUTOR07.PRG.....	70
TUTOR08.PRG.....	70
FWBORL.PRG.....	70
FWBROW.PRG.....	70
Guía de Ejemplos FiveWin.....	70
Comandos xBase	70
Comandos de Ventanas.....	75
Ventanas:.....	75
MDI Múltiple Document Interface en Ventanas.....	75
¿Que es MDI ?.....	76
¿Porqué MDI?.....	76
Elementos de Un Entorno MDI	77
Cómo hacer MDI con FiveWin.....	77
Implementación de Orientación a Objetos en MDI.....	78
Nuevas direcciones MDI en FiveWin.....	78
Comandos de Dialogs.....	79
Dialogs:.....	79
Folders:.....	81
Pages:.....	83
Tabs:.....	883
Comandos WinControls.....	84
Bar:	84
Browses:.....	87
Botones:.....	88
CheckBox:.....	90
ComboBox:.....	92
Gets:	93
ListBoxes	96
Barra de Mensajes:.....	97
RadioButtons:	99
Says.....	100
Scrollbars:.....	101
Trees (Arboles):.....	103
Video:.....	104
VBXs, Visual Basic controls:.....	105
Meters:.....	106
Comandos de Recursos.....	107
Bitmaps:.....	107
Cursors:.....	108
Brushes:.....	109
Icons:.....	109
Fonts:.....	110
Pens (Lápices):.....	111
Menus:.....	112
Comandos misceláneos.....	114
Clipboard:.....	114

DDE (Dynamic Data Exchange):	114
Configurando y manipulando archivos HELP:	115
Manejando archivos INI:	115
MRU, Most Recently used items (Items Usados Recientemente):	115
ODBC, Manejando Bases de Datos mediante ODBC (Open DataBase Connectiviti):	115
Usando la Impresora:	116
Generador de Reportes FiveWin:	117
Timers (Cronómetros):	117
xBase elementos comunes.....	117
Define.....	118
Activate.....	118
:End().....	118
Of:.....	118
Resource/Resname/Disk:.....	119
Filename/File/Disk:.....	119
Size:.....	119
Classes and Objects	120
Clases Windows	120
Clases Dialog.....	120
Clases WinControl	120
Clases para Recursos	120
Clases para Bases de Datos	121
Clases para Comunicaciones	121
Clases para Reportes e impresión	121
Clases Misceláneas.....	121
Objects engine syntax	122
Funciones	123
Clipboard.....	123
Color	123
Comunicaciones	124
Fecha / Hora	124
Facilidades de depuración.....	124
DialogBoxes.....	124
Mantenición de Directorios.	124
DLLs.....	124
Dibujo.....	125
Ejecutando otras aplicaciones	125
Manejo de Archivos	125
Manejo de archivos de Ayuda.....	125
Iconos	126
Archivos INI	126
Keyboard.....	126
Mouse	126
Manejo de archivos NTX/MDX.....	126
Manipulación Numérica	126
OemAnsi	127
OOPS.....	127
Seguridad / Protección de Copia	127
Sound.....	127
Strings (Cadenas de Caracteres)	127
System.....	127
Timers.....	128
Use DialogBox Rápidos.....	128
Manejo de archivos ZIP (solo UnZip)	128

DDE (Dynamic Data Exchange):	114
Configurando y manipulando archivos HELP:	115
Manejando archivos INI:	115
MRU, Most Recently used items (Items Usados Recientemente):	115
ODBC, Manejando Bases de Datos mediante ODBC (Open DataBase Connectiviti):	115
Usando la Impresora:	116
Generador de Reportes FiveWin:	117
Timers (Cronómetros):	117
xBase elementos comunes.....	117
Define.....	118
Activate.....	118
:End().....	118
Of:.....	118
Resource/Resname/Disk:.....	119
Filename/File/Disk:.....	119
Size:.....	119
Classes and Objects	120
Clases Windows	120
Clases Dialog.....	120
Clases WinControl	120
Clases para Recursos	120
Clases para Bases de Datos	121
Clases para Comunicaciones	121
Clases para Reportes e impresión	121
Clases Misceláneas.....	121
Objects engine syntax	122
Funciones	123
Clipboard.....	123
Color	123
Comunicaciones	124
Fecha / Hora	124
Facilidades de depuración.....	124
DialogBoxes.....	124
Mantenición de Directorios.	124
DLLs.....	124
Dibujo.....	125
Ejecutando otras aplicaciones	125
Manejo de Archivos	125
Manejo de archivos de Ayuda.....	125
Iconos	126
Archivos INI	126
Keyboard.....	126
Mouse	126
Manejo de archivos NTX/MDX.....	126
Manipulación Numérica	126
OemAnsi	127
OOPS.....	127
Seguridad / Protección de Copia	127
Sound.....	127
Strings (Cadenas de Caracteres)	127
System.....	127
Timers.....	128
Use DialogBox Rápidos.....	128
Manejo de archivos ZIP (solo UnZip)	128

Guía de Referencia

La Guía de referencias de FiveWin ofrece una detallada descripción de los diferentes componentes de la librería FiveWin. Use esta sección para encontrar como compilar y enlazar con FiveWin y además, la descripción de cualquier comando o función.

Técnicas de Programación y recomendaciones son explicadas en la sección "Guía de Programación" de FiveWin.

La Guía de Programación FiveWin enseña como construir aplicaciones usando FiveWin.

La Guía de Referencia FiveWin explica los detalles que necesita saber para construir esas aplicaciones.

Descripción del Producto

Que es FiveWin?

FiveWin es una librería para CA-Clipper 5.x que se usa de la misma forma que todas las otras librerías. La principal diferencia es ese FiveWin permite construir ejecutables (EXEs) que corren en Microsoft Windows como Programas Windows REALES. ¡Tiene el poder de Windows y el entorno, en la punta de los dedos!

No necesita saber Windows o su complejo sistema de programación. FiveWin controla todo, permite usar sintaxis CA-Clipper de manera fácil y cómoda.

Lo único que requiere es un enlazador (linkeador) capaz de producir ejecutables Windows. Como Microsoft Link.exe versión 5.1 o superior, Optlink, Blinker 3.2 o superior. Otros Linkeadores "DOS" montadores, como RTLink, Blinker 2.x, Plink, CauseWay o Exospace, no tienen la habilidad de crear ejecutables Windows y, por consiguiente, no son útiles.

Si prefiere usar el linkeador de Microsoft, puede encontrarlo con casi cualquier de los lenguajes Microsoft o en internet. Además el sistema operativo IBM OS/ 2 también incluye este linkeador.

No debe usar TLINK.EXE de Borland, no es totalmente compatible con el sistema de enlace Microsoft.

Preferimos Blinker 4,x

Mientras el linkeador es la única herramienta Windows que debe tener, es recomendado obtener una copia del Borland WorkShop. Permite la edición visual de los recursos de una aplicación, pero no se requiere.

Compilando y Enlazando (Linking)

Para compilar sus Aplicaciones FiveWin, debe usar el comando Clipper como es usual:

```
CLIPPER Test /n/w
```

Note: Se asume que los archivos .CH de FiveWin están incluidos en su variable de entorno SET INCLUDE.

Todas las Aplicaciones FiveWin necesitan especificar:

```
#include "Fivewin.ch"
```

al comienzo de su archivo .prg, esto extiende la sintaxis de CA-Clipper para usar Windows.

Linkear con Ms-Link o Blinker:

Proveemos las siguientes cinco Librerías:

Five.LIB, FiveC.LIB, Objects.LIB, WinApi.LIB, 501_520

501_520 es necesaria solo si usa Microsoft LINK.

Producir su EXE con FiveWin requiere un Linkeador Windows. FiveWin es totalmente compatible con:

Microsoft LINK.EXE versión 5.3 y superior. Para usar la versión 5.1 deberá usar una herramienta que proveemos, llamada ExeType.exe para cambiar la versión Windows dentro del ejecutable. Esta utilidad esta en el Directorio de FiveWin.

Formato Microsoft Link:

```
Link /se:700 /NOE <YourApp>,,nul, Fivewin Winapi clipper 501_520, ... def
```

Si usa LINK 5.1, después de Linkear use:

```
ExeType YourApp.exe 3.1
```

Blinker 3.2:

```
Blinker fi YourApp @FiveWin.Ink (probablemente modifique este archivo para indicar sus directorios)
```

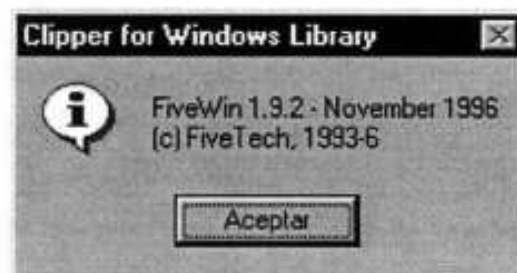
Vea FiveWin.def y FiveWin.Ink para requisitos especiales de las aplicaciones FiveWin.

Sus Primeras Aplicaciones

Vea los siguientes ejemplos en su directorio \SAMPLES.

TUTOR01.PRG

Típica ventana de mensajes, ¡Desde Windows!



TUTOR02.PRG

Nuestra primera Ventana en Windows.

TUTOR03.PRG

Nuestra Primera ventana, ahora con ejemplos de control de eventos y barra de mensajes.

TUTOR04.PRG

Ventana con Menú PullDown.

TUTOR05.PRG

Similar a Tutor04, pero con la ventana maximizada.

TUTOR06.PRG

Construyendo DialogBox directamente desde un Editor de Recursos.

TUTOR07.PRG

Algo más que en TutorO6

TUTOR08.PRG

Primer trabajo con Browsers.

FWBORL.PRG

Este ejemplo muestra un Manejador de Bases de Datos simples.

FWBROW.PRG

Este ejemplo muestra un típico Manejador de Bases de Datos.

Guía de Ejemplos FiveWin

Recomendamos aprender FiveWin por medio de los ejemplos. Pruébelos y modifíquelos, siéntase libre de usar cualquier parte de ellos que requiriera.

ACHOICE.PRC

Usa ListBoxes para ejecutar selecciones simples

ADIR.PRG

Prueba pequeña sólo verifica algunas funciones de arreglos desde Windows.

ALERT.PRG

Reemplazo FiveWin del Alert de CA-Clipper

ATEST.PRG

Comprobación de dimensiones de arreglo bajo Windows

AUTOLAY.PRG

FiveWin - Resource desde una dbf generadora! Úselo para desarrollar Dialog Rápidos.

BARSETUP.PRG

Programa: BarSetup.prg

BIGFOLD.PRG

Comprobación de muchos folders.

BMPTXT.PRG

Pone texto Sobre un bitmap.

BTNSETUP.PRG

ButtonBar Drag&Drop ejemplo de configuración

CALENDAR.PRG

Programa

CARET.PRG

Este ejemplo muestra como conectarse a CA-Ret usando DDE.

CLOSEDLL.PRG

Utilidad para descargar un DLL Abierto desde otra Aplicación.

COMBOS.PRG

Muestra el uso de diferentes estilos de Controles ComboBox.

COMPILE.PRG

Muestra como compilar una aplicación desde una aplicación Fivewin.

COREL.PRG

Usando CorelDraw para generar Templates de Impresión.

CUSTOMER.PRG

Comprobación de Aplicaciones MDI.

DBCTEST.PRG

Aplicación Básica MDI con ejemplo de reportes.

DBF01.PRG

FiveWin - Objeto DataBase Tutorial 01.

DBF02.PRG

FiveWin - Manejando Drag&Drop con browser.

DBF2RC.PRG

.DBF para generar RC. Rápidamente crea Dialogs desde DBFS.

DBFBUILD.PRG

Un constructor de Bases de Datos DBF escrito en FiveWin.

DBFW.PRG

Muestra cómo cambiar de Oem a Ansi y de voice a overs

DDECLIEN.PRG

comprueba las capacidades DDE Client y DDE server con FiveWin.

DDEMLCLI.PRG

Demostración del TDDEMLClient de FiveWin.

DDEMLSVR.PRG

Demostración del DDEML TDDEMLServer de FiveWin.

DDESERVE.PRG

Usando FiveWin como un DDE Server

DEVCAPS.PRG

Testeando DeviceCapabilities()

DLGTEST.PRG

Comprobación del uso masivo de DialogBox.

DLLCALL.PRG

¡Esta muestra cómo hacer una Vinculación Dinámica en runtime!

DLLCLEAR.PRG

Utilidad para remover los DLLs Abiertos en Windows.

EMAIL.PRG

Ejemplo de programa Email escritos por usuarios FiveWin.

ENDAPP.PRG

Ejemplo simple muestra cómo activar y terminar una aplicación externa

FILEBROW.PRG

Simple Browser de archivos en FiveWin.

FIVEDEMO.PRC

"Centro de Control" de demos y ejemplos FiveWin.

FIVEDIT.PRG

Editor de Código Fuente propio de FiveWin - Manejo de Sintaxis con color.

FIVEGURU.PRG

Este Ejemplo muestra como construir aplicaciones en modo visual con FiveWin.

FIVETEST.PRG

Ejemplo de plataformas entre FiveWin, FiveOS2 y FiveDos

FWBORL.PRG

Este ejemplo muestra cómo crear un Manejador de Base de Datos simple.

FWBROW.PRC.

Ejemplo de un rápido Quick Browse.

FWBROW2.PRG	Este ejemplo muestra como crear un típico manejador de Bases de Datos.
FWFRAME.PRG	Ejemplo que muestra los diferentes bordes de los DialogBox.
FWREPORT.PRG	Muestra como crear un Reporte manejando Base de Datos.
FWRES.PRG	Muestra como construir una Aplicación Windows.
GENFOLD.PRG	Use Carpetas y manipúlelos en runTime.
IDXBUILD.PRG	Constructor genérico de Índices. Ver IDE.prg
INKEY.PRG.	Comprobando el InKey() de CA-Clipper.
INSTALL.PRG	Programa de Instalación FiveWin Versión 1.1
KEYBSIM.PRG	Ejemplo de Un teclado "virtual".
LABEL.PRG	Generar Etiquetas de correo - usando el Preview de Reportes en FiveWin.
LABEL2.PRG	Etiquetas FiveWin! desde FiveWin!
LBXDRAW.PRG	Ventanas estándar de ListBox con dibujo de BitMaps.
LEDDISP.PRG	Un buen Odómetro desarrollado por un usuario FiveWin
LINES.PRG.	Dibujando Líneas con manejo de colores.
LOLO.PRG	Aprender a multiplicar números. Por Lolo hijo de Antonio Linares
MDIMEMO.PRG	Propósito: Edición de memos en MDI Hija.
MEMORY.PRG	Comprueba variables de memoria CA-Clipper dentro de FiveWin.
NIYNIBX.PRG	Generado controles VBX contenido en un DialogBox.
NESTED.PRG	Emisor múltiple anidando Cajas del Diálogo
NEWHELP.PRG	Muestra la muestra cómo trabajar con técnicas de las nuevas ayudas de FiveWin.
02.PRG	Contador visual desarrollado por un usuario FiveWin.
OBJINSP.PRC.	FiveWin Object Inspector
ODBCO.PRG	FiveWin ODBC Tutorial 1
ODBCI.PRG	FiveWin ODBC capacidades.
ODO.PRG	Un buen Odómetro desarrollado por un usuario FiveWin
OWNERDR.PRG	Testeo "ownerdraw" ListBoxes y ComboBoxes
PEEKPOKE.PRG	Testeo de las nuevas funciones FiveWin Peek y Poke.
PHONE.PRG	FiveWin Phone - (c) FiveTech 1993-5
PRG2EHT.PRG	Documentación Automática para Código Fuente usando el objeto TLex.
RC2PRG.PRG	Utilitario de conversión de RC Windows a archivo PRG - Desarrollado por Oliver
RCIMPORT.PRG	Utilitario FiveWin para importar Rcs y generar código fuente.
RCLICK.PRG	Ejemplo que muestra el Control de eventos del Click Derecho.
RESMENU.PRG	Usa Menús pulldown desde recursos
SCANNED.PRG	Mezcla imágenes escaneadas con datos.
SM_BAR.PRG	Muestra una Toolbar simple como el Toolbar de MS-Office
TESTI6.PRG	Un prototipo NT ISAPI Windows para usar Clipper + FiveWin como DataBase
TEST32.PRG	Prueba las nuevas capacidades del API bits de FiveWin.
TESTADD.PRG	Ejemplo de las capacidades Enable, Disable, Show y Hide de los Controles
TESTALIG.PRG	Nueva poderosa técnica de alineación automática de FiveWin.
TESTARGS.PRG	Testeo de parámetros en Línea de Comandos.
TESTARR1.PRG	Revisar arreglo- usando TWBrower de Fivewin.
TESTARR2.PRG	Revisar arreglos bidimensionales usando el TWBowser de FiveWin
TESTARR3.PRG	Manejando arreglos sin elementos.
TESTARR4.PRG	Muestra como crear controles dentro de un Ciclo.
TESTARR5.PRG	Browser de un arreglo multi-dimensional creado por Directory().
TESTARR6.PRG	Manejando arreglos con TCBrowse
TESTBAR.PRG	Muestra como crear un manejador de Menú con Button Bar y Menu.
TESTBAR2.PRG	Demostración de Características Avanzadas de ButtonBar.
TESTBAR3.PRG	Esta muestra cómo crear una prueba de Barra con un Get Valido.
TESTBARR.PRG	Demostración de características avanzadas de ButtonBar.
TESTBLOC.PRG	Objetos Codeblock Almacenables en Disco
TESTBMP.PRG	Ejemplo de Imágenes en Bases de Datos con Fivewin desde recursos.
TESTBMP1.PRG	Ejemplo de manejo de BitMaps con adjust, scroll y nuevos efectos de zoom.
TESTBMP2.PRG	Muestra como desplegar Bitmaps Flotantes.
TESTBMPD.PRG	Muestra como crear un simple manejador dinámico de BitMaps.
TESTBORL.PRG	Prueba de Carga y Descarga la librería Borland BWCC.dll
TESTBRAR.PRG	Browser con arreglos bidimensionales con el poderoso TWBrower de FiveWin.
TESTBRU1.PRG	Probando algunos buenos Brushes.
TESTBRUS.PRG	Probando Brushes

TESTBRW.PRG	Algunos ejemplos desarrollados por usuarios.
TESTBRW2.PRG	Ejemplo de Quick Browser.
TESTBTN.PRG	Definiendo Botones ButtonBar con Bitmaps desde Disco.
TESTBTNB.PRG	Probando el TBtnBmp de FiveWin en una DialogBox desde recursos.
TESTBTNC.PRG	Probando la nueva clausula CANCEL, del pushbutton
TESTBTNS.PRG	Una Simple Dialog Box con una Lista de Botones.
TESTBTNS.PRG	Usando el soporte de VBX Borland BiVbx1O.dll.
TESTCAPT.PRG	Ejemplos de Captura de mouse.
TESTCGI.PRG	Escribir aplicaciones CGI-Win usando FiveWin para un Servidor WEB
TESTCHCK.PRG	Prueba simple de un checkbox
TESTCLIP.PRG	Muestra como usar el Clipboard Windows desde FiveWin.
TESTCLR.PRG	Muestra como crear un Test de Gets Coloreados.
TESTCLR2.PRG	Nuestra Primera Ventana en windows.
TESTCLR3.PRG	Una Prueba simple de Color.
TESTCOMB.PRG.	Muestra como crear un Combo.
TESTCOMM.PRG	Ejemplo de Comunicaciones FiveWin - Windows
TESTCRC.PRG	Prueba la nueva función nFileCRC()
TESTCT3D.PRG	Prueba el Look 3D Microsoft con Ctl3d.dll
TESTCURS.PRG	Prueba de Cursor.
TESTCYCL.PRG	Muestra como circular sobre algunos Gets
TESTDAT2.PRG	Muestra Gets de Fechas.
TESTDAT3.PRG.	Trabaja con fechas CA-Clipper bajo Windows.
TESTDATE.PRG	Muestra como Actualizar la Fecha.
TESTDBG.PRG	Muestra como activar una segunda Ventana desde una primaria.
TESTDDE.PRG	Muestra como usar enlaces DDE.
TESTDESK.PRG	Muestra como crear una Ventana con Estilos especiales.
TESTDIR.PRG	Muestra como ver el Directorio Windows.
TESTDIRS.PRG	Ejemplo de manejos de Directorios.
TESTDLG2.PRG	Construyendo una Aplicación con un Dialog Box.
TESTDLG3.PRG	Más pruebas de Dialog y Window.
TESTDLG4.PRG	Más ejemplos combinando Ventanas y Dialog Box.
TESTDLG5.PRG	Prueba de TextMetrics y soluciones. Gracias a Miguel y Derk de Países Bajos
TESTDLG6.PRG	Probando una DialogBox con clausula Valid.
TESTDLG7.PRG	Definiendo una DialogBox con algún estilo
TESTDLG8.PRG	Creando una DialogBox sysmodal.
TESTDLG9.PRG	Prueba de VALID en una DialogBox No-Modal
TESTDLGB.PRG	Pintando un Bitmap sobre un DialogBox
TESTDLGF.PRG	Probando Eventos de Focus en un DialogBox
TESTDLGN.PRG	Probando una DialogBox no modal.
TESTDROP.PRG	Usando el FileManager Windows y Drag&Drop con un archivo sobre esta Ventana.
TESTDRP2.PRG	FiveWin - DropFiles xBase
TESTDRP3.PRG	FiveWin - Drag&Drop nuevas características.
TESTDRP4.PRG	FiveWin - Drag&Drop nuevas características.
TESTDRVS.PRG	Probando Todos los drivers validos del Sistema.
TESTDSN.PRG	Probando la Clausula Diseño.
TESTENCR.PRG	Nuevas funciones FiveWin Encrypt() y Decrypt().
TESTEND.PRG	Probando una ventana simple con un MsgYesNo como validación.
TESTERRO.PRG	Probando el sistema de errores de FiveWin - Prueba el archivo generado Error.Log.
TESTEXIT.PRG	Probando la Clausula Valid de la Ventana
TESTFB.PRG	Prueba de un Folder Simple.
TESTFBIN.PRG	Probando la nueva Clase TFile. Alerta: La vieja TFile es ahora TtxtFile
TESTFDAT.PRG	Probando la función SetFTime()
TESTFILE.PRG	Manejo de Archivos ASCII con FiveWin Orientado a Objetos.
TESTFLAG.PRG	Probando diferentes sistemas información Flags de ventanas.
TESTFLD2.PRG	Dos folders al Mismo Tiempo.
TESTFLD3.PRG	Construye Folder Dinámicos en el Top de una DialogBox Dinámica.
TESTFOC2.PRG	Muestra como detectar el control con Focus dentro de una DialogBox.
TESTFOCU.PRG	Controla el Focus desde una ventana a otra.
TESTFOLD.PRG	Prueba el soporte de controles Folders de FiveWin
TESTFON2.PRG	Prueba las capacidades de los Fonts

TESTFONT.PRG	Muestra como usar diferentes Fonts en la Ventana.
TESTFORM.PRG	Prueba el Nuevo TForm de FiveWin 2.0
TESTGET.PRG	Prueba un GET
TESTGETN.PRG	Prueba un GET editando un número
TESTGETR.PRG	Prueba un GET con justificación a la derecha.
TESTGFIL.PRG	Prueba el nuevo 4to parámetro de la función cGetFile()
TESTGRAF.PRG	Algunos gráficos comerciales usando FiveWin
TESTHARD.PRG	Retorna el numero de serie del Disco Duro.
TESTHASH.PRG	Prueba la nueva función nStrHash() de FiveWin
TESTHD.PRG	La nueva función nSerialHD() de FiveWin para comprobar la serie del Disco Duro.
TESTHELP.PRG	Muestra como usar el sistema de ayudas de Windows.
TESTHEX.PRG	Prueba las funciones de conversión de decimal a Hexadecimal.
TESTICO.PRG	Animaciones usando Iconos y Timers.
TESTICON.PRG	Jugando con Iconos...
TESTIDLE.PRG	Control y Manejo de Tiempos Ociosos con FiveWin, Procesos BackGround.
TESTINI.PRG	Muestra el uso de archivos INI con FiveWin.
TESTIN12.PRG	Usando archivos INI de Windows para configurar la información.
TESTKEY.PRG	Prueba de manejo de Teclado.
TESTKEYS.PRG	SET KEY manejados con FiveWin
TESTLBX.PRG	Ejemplo de uso de ListBox
TESTLBX2.PRG	Poniendo Bitmaps sobre ListBoxes de FiveWin.
TESTLBX3.PRG	Usando ListBoxes para hacer selección Simple.
TESTLBX4.PRG	Usando ListBoxes para hacer selección Múltiple
TESTLFN.PRG	Probando funciones FiveWin para soporte de nombres largos de Windows 95.
TESTLIB.PRG	Funciones FiveWin para manejo de Archivos LIB.
TESTLIB2.PRG	Probando la nueva Clase TLibFile.
TESTLOOP.PRG	Muestra como crear controles desde un ciclo....
TESTMAIL.PRG	Demuestra el soporte FiveWin para - Windows MAPI (servicios de correo).
TESTMBAR.PRG	Prueba diferentes Fonts el la MsgBar
TESTMBRS.PRG	Maneja múltiples ButtonBars
TESTMDI.PRG	Trabajando con entornos MDI.
TESTMDI2.PRG	Parte el aprendizaje de entornos MDI
TESTMDI3.PRG	Parte el aprendizaje de entornos MDI
TESTMDI4.PRG	Pone controles dentro de una MDIChild.
TESTMDI5.PRG	Usando MDIChilds con estilos definidos por el usuario.
TESTMDI6.PRG	Controlando Menús MDI -Cargando Dinámicamente Menús de ventanas MDIChild.
TESTMDI7.PRG	Poniendo un Logo BitMap en un Entorno MDI.
TESTMDI8.PRG	Poniendo un Browser dentro de una ventana MDIChild.
TESTMDI9.PRG	Probando un menú Pulldown con opciones predefinidas
TESTMDIX.PRG	Pone controles dentro de una MDIChild
TESTMDIY.PRG	Usando MDIChildren con estilos definidos por el usuario.
TESTMEM2.PRG	Algunas pruebas de usuarios.
TESTMEMO.PRG	Muestra el manejo de la Edición de un Memo.
TESTMENU.PRG	Usa un menú Simple...
TESTMETR.PRG	Prueba los controles meter de FiveWin
TESTMGET.PRG	Pruebas sobre archivos INI, Dialog e Impresora.
TESTMMD.PRG	Probando el Mouse en un entorno MDI
TESTMMT.PRG	Trabaja con la Función MsgMeter() de FiveWin
TESTMNU.PRG	Almacena un Menú pulldown en una DBF.
TESTMNU2.PRG	Prueba de Menús PullDown.
TESTMNU3.PRG	Ejemplo de Menú PullDown.
TESTMNU4.PRG	Algunos ejemplos de Menú PullDown.
TESTMNU5.PRG	Muestra como definir la acción de un MenuItem.
TESTMOUS.PRG	Probando los eventos del Mouse.
TESTMRU.PRG	La nueva tecnología MRU de FiveWin.
TESTMSG.PRG	Prueba las DialogBox Estándar de Mensajes.
TESTMSG2.PRG	Nuevas funciones Msg... característica conversión automática de Valor a String.
TESTMSG3.PRG	La Nueva MessageBar.
TESTNET.PRG	Usar Servicios de Red Windows en DialogBox.

TESTNOM.PRG	Algunas Pruebas de DialogBox No-Modal
TESTNOMO.PRG	Pruebas de DialogBox No-Modal
TESTNTX.PRG	Prueba de la función cNtxKey() de FiveWin para extraer una clave NTX desde archivo.
TESTOBJ.PRG	Manipulación de Archivos Obj con FiveWin.
TESTOERR.PRG	Prueba del nuevo OBJECTS engine de FiveWin: Error handler definido por el Usuario
TESTOP.PRG	Prueba del nuevo OBJECTS 4.0 engine de FiveWin, operadores de overloading
TESTPAGE.PRG	Probando la nueva Clase TPages de FiveWin.
TESTPARA.PRG	Probando la función GetEnv() de CA-Clipper desde Windows
TESTPARS.PRG	Comprobación de la máquina parser de FiveWin
TESTPASS.PRG	Muestra ventana que solicita PassWord antes que aparezca la ventana Principal.
TESTPIE.PRG	Prueba la función Pie y los parámetros usados
TESTPOLY.PRG	Dibuja Polígonos usando Windows API
TESTPOP.PRG	Prueba de Menús Popup en FiveWin
TESTPOP2.PRG	Prueba Menús Popup desde recursos
TESTPORT.PRG	Prueba nuevas funciones de manipulación de HardWareTesting de FiveWin.
TESTPRN.PRG	Imprimiendo una ventana completa
TESTPRN1.PRG	Muestra como seleccionar desde código la orientación de la impresora.
TESTPRN2.PRG	Muestra el manejo del Objeto Printer
TESTPROF.PRG	FiveWin técnicas de Profiling
TESTRAD.PRG	Manejo de Radio Buttons
TESTRAD2.PRG	Prueba de algunos Radio Buttons
TESTRAD3.PRG	Más ejemplos con RadioButtons
TESTRAD4.PRG	Más ejemplos con RadioButtons
TESTRAD5.PRG	Radios y WHEN en un GET
TESTRAND.PRG	Controlando menú MDI menú Dinámicos de ventanas MDIChild
TESTREAD.PRG	Probando la función Fread() bajo Windows
TESTREP1.PRG	Este es un demo del Report Engine de FiveWin
TESTRES.PRG	Muestra el uso de Dialog Almacenadas en un DLL.
TESTRES2.PRG	Probando la función HasResources(<cExeName>) de FiveWin.
TESTRPT2.PRG	Ejemplo del Report Engine de FiveWin
TESTRPT3.PRG	Más Reportes...
TESTRPT4.PRG	Más Reportes...
TESTRPT5.PRG	Más Reportes...
TESTRUN.PRG	Muestra la nueva función RUN() de FiveWin con Ejecutables.
TESTSAY.PRG	Muestra el uso de SAY() con Fonts.
TESTSCRL.PRG	Uso de Scrollbars
TESTSIZE.PRG	Cómo controlar dimensiones, posición, tamaño y tracking
TESTSOCK.PRG	Manejando sockets TCP/IP desde FiveWin
TESTSTR.PRG	Cargando Strings desde un RC = TestStr.rc
TESTSTY2.PRG	Una DialogBox sin borde ni caption
TESTSTYL.PRG	Usando Stilos alternativos de Ventanas
TESTSYSM.PRG	Redefiniendo el "System Menu" de una ventana
TESTTABS.PRG	Probando los nuevos Tabs de FiveWin
TESTTFN.PRG	Probando la nueva función cTempFile que retorna un nombre de archivo temporal.
TESTTIM2.PRG	Usando Objetos Time
TESTTIM3.PRG	Muestra el llamado a una función en un intervalo de tiempo.
TESTTIME.PRG	Usando Timers. Timers son Objetos
TESTTOK.PRG	Probando la función StrToken()de FiveWin
TESTTOOL.PRG	Probando ToolHelp.dll de Windows
TESTTOP.PRG	Prueba cómo poner una aplicación en el TOP
TESTVBX1.PRG	FiveWin - VBX Visual Basic controls
TESTVBX2.PRG	FiveWin - VBX Visual Basic controls
TESTVBX3.PRG	FiveWin - VBX Visual Basic controls
TESTVBX4.PRG	FiveWin: Usando Tabs VBXs
TESTVBX5.PRG	FiveWin - VBX Visual Basic controls
TESTVBX6.PRG	FiveWin - VBX Visual Basic controls
TESTVERS.PRG	Quiere Ver las versiones Windows & Dos

TESTVID.PRG	Muestra el uso de Multimedia y su expansión de capacidades a VIDEO.
TESTWAIT.PRG	Probando la función MsgWait() de FiveWin
TESTWHEN.PRG	Probando la Clausula WHEN en los controles.
TESTWRIT.PRG	Muestra como manipular Aplicaciones Externas desde FiveWin.
TESTWSTY.PRG	Prueba los estilos de Ventanas.
TESTYACC.PRG	Prueba la maquina YACC de FiveWin.
TESTZIP.PRG	Manejo de Archivos ZIP con unzip de FiveWin
TOOLTIP.PRG	Muestra como implementar Tools-Tips con FiveWin.
TOPAPP.PRG	Prueba cómo poner otra aplicación en el TOP
TSTBTNBM.PRG	Ejemplo de botones con Texto y Bitmap
TUTOR01.PRG	Típica Ventana con mensaje BienVenido de Windows!
TUTOR02.PRG	Nuestra Primera Ventana en Windows.
TUTOR03.PRG	Nuestra Primera Ventana, ahora con control de eventos y barra de mensajes.
TUTOR04.PRG	Construimos un menú PullDown y lo asignamos.
TUTOR05.PRG	Similar a Tutor04, pero con ventana maximizada.
TUTOR06.PRG	Construye una DialogBox Directamente.
TUTOR07.PRG	Más trabajo con el Tutor06
TUTOR08.PRG	Primer trabajo con Browser
TWOCLASS.PRG	Declaramos dos Clases en un Archivo PRG..
VBXCLASS.PRG	Inspector interno de VBX.
VIEWICO.PRG	Muestra iconos.
WIN95_2.PRG	Maneja Arboles de Windows 95

Comandos xBase

Comandos de Ventanas

Ventanas:

Definición :

```
DEFINE WINDOW <oWnd>;
  [FROM <nTop>, <nLeft> TO <nBottom>, <nRight>] ;
  [TITLE <ctitle> ] ;
  [<color: COLOR, COLORS> <nClrFore> [,<nClrBack>] ] ;
  [OF <oParent> ] ;
  [BRUSH <oBrush> ] ;
  [CURSOR <oCursor> ] ;
  [ICON<olcon> ] ;
  [MENU <oMenu>] ;
  [STYLE <nStyle>] ;
  [BORDER [ NONE / SINGLE ] ] ;
  [NOSYMENU / NO SYSMENU ] ;
  [NOCAPTION / NO CAPTION / NO TITLE] ;
  [NOICONIZE / NOMINIMIZE ];
  [NOZOOM / NO ZOOM / NOMAXIMIZE / NO MAXIMIZE] ;
  [ VSCROLL / VERTICAL SCROLL ];
  [ HSCROLL / HORIZONTAL SCROLL]
```

<oWnd>	Referencia a un objeto Window
<nTop>	Coordenadas en la pantalla. Simulamos coordenadas texto.
<nLeft>	
<nBottom>	
<nRight>	
<cTitle>	Titulo de la Ventana.
<oparent>	Referencia a la Ventana Padre.
<oBrush>	Un Objeto Brush Existente para ser usado como fondo de la ventana. Use DEFINE BRUSH ... para crearlo, Este parámetro es opcional.
<oCursor>	Referencia al Objeto Cursor a desplegar sobre este control.

<olcon>	Referencia a un Objeto Icon a ser usado por la Ventana. Use DEFFNE ICON ... para crearlo. Este parámetro es opcional.
<oMenu>	Un Objeto Menu Existente. Use MENU ... ENDMENU desde una función como en los ejemplos. este parámetro es opcional.
<nStyle>	Valor numérico que indica un estilo especial de API. [Sólo para usuarios Avanzados].
<nClrFore>	Color a usar en la ventana. Puede ser en colores RGB a String de Colores xBase.
<nClrBack>	

Activándola:

```
ACTIVATE WINDOW <oWnd>;
  [ICONIZED / NORMAL / MAXIMIZED ] ;
  [ON LEFT CLICK <uLClick> ] ;
  [ON RIGHT CLICK <uRClick>] ;
  [ON MOVE <uMove> ] ;
  [ON RESIZE <uResize> ] ;
  [ON PAINT <uPaint> ] ;
  [ON KEYDOWN <uKeyDown>] ;
  [ON INIT <uInit> ]
```

<oWnd>	Referencia al Objeto Window
<uLClick>	Acción a ejecutar cuando el botón izquierdo del mouse es presionado sobre la ventana.
<uRClick>	Acción a ejecutar cuando el botón derecho del mouse es presionado sobre la ventana.
<uMove>	Acción a ejecutar cuando la ventana se mueva.
<uResize>	Acción a ejecutar cuando se reajuste el tamaño de la ventana.
<uPaint>	Acción, generalmente de dibujo, a ejecutar cuando la ventana se pinte.
<uDown>	Acción a ejecutar cuando una tecla es presionada en la ventana.
<uInit>	Acción a ejecutar, generalmente en la inicialización, cuando la ventana se inicializa.

MDI Múltiple Document Interface en Ventanas

Crear una ventana MDI es similar a crear una ventana normal. La diferencia es que hay que especificar la clausula MDI en la ventana Principal, y especificar la clausula CHILD en la ventana MDI hija.

Cuando crea la ventana MDI otras Cláusulas se pueden especificar:

<nMenuInfo> Es la posición del PopUp en el Menu PullDown donde windows automáticamente informa la explicación de la Ventana MDI Hija

¿Que es MDI ?

En cuanto comience a sentirse cómodo con Windows enfrentará uno de los rasgos más poderosos de él: Múltiple Document Interface.

La mayor parte del Windows aplicaciones comerciales que usamos (el Program Manager de Windows usa MDI) aprovecha de las posibilidades de MDI.

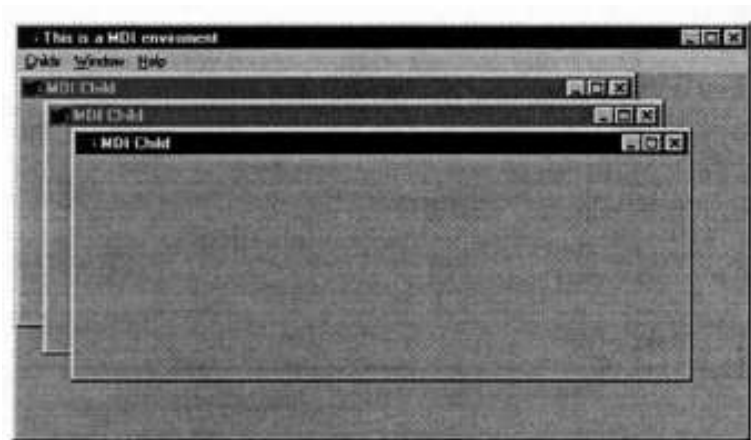
Básicamente el entorno MDI es una Ventana principal que tiene varias Ventanas hijas dentro, que pueden ser reajustadas, tituladas, poner en cascada, maximizar, crear y destruir muy fácilmente. De hecho, un ambiente MDI es como un ambiente de la Ventana entera todo por sí misma.

Estos rasgos, que parecen muy comunes, no trabajarán si trata de hacerlos usando una Ventana normal. Podría hacerlos, pero significaría muchas líneas de código.

Windows, en cambio, ofrece el sistema MDI para crear estas aplicaciones.

¿Porqué MDI?

¡Hacer cosas más fácil! Automáticamente hacemos muchos rasgos buenos y poderosos sin la necesidad de codificación. ¡Windows toma mando de todo!



Elementos de Un Entorno MDI

Existe una gran diferencia a entender sobre la programación MDI:

Windows guarda una Ventana 'invisible' que se extiende en el área 'frame' del cliente de la Ventana:

El 'trick' real de la programación MDI es la Ventana MDI del Cliente que Windows ha implementado.

Desarrollar un entorno MDI en Lenguaje C es complejo. De hecho, está es una de las tareas más difíciles para principiantes. ¡Pero tiene suerte porque al hacerlo con CA-Clipper y FiveWin es una tarea muy sencilla!

Cómo hacer MDI con FiveWin

Para crear la ventana principal del frame MDI usamos los comandos FiveWin normales de creación de Ventanas. La diferencia es que debe especificar la cláusula MDI. como antes se vio.

Ahora, cree ventanas hijas (CHILD) con los mismos simples comandos. No se debe preocupar por la ventana "oculta" MDI del Cliente.

FiveWin automáticamente maneja y se comunica con la Ventana MDI del Cliente.

Para crear una Ventana MDI hija, de nuevo use los comandos normales de creación de ventanas, pero debe especificar la cláusula MDICHILD.

¡Todo lo que ya sabe sobre FiveWin no cambia! Sólo debe recordar unos artículos:

Un Objeto Ventana MDI frame tiene 'DATA' llamado oWndClient, Con otro Objeto Ventana - el 'ghost' MDI Cliente!.

¡De hecho, El arrea que se ve dentro del marco MDI es el MDI Cliente!!!

No olvide eso. Así, si -por ejemplo- piensa poner un bitmap en el área del frame MDI, realmente tiene que ponerlo en el MDI Cliente. Así se hace esto:

@ 2, 2 BITMAP ... OF ownd:oWndClient

¡si olvida esto, y solo lo hace de oWnd, pondrá el Bitmap detrás del MDI Cliente, así no vera el bitmap!
¡Recuérdelo!

Este es el único 'trick' sobre la programación MDI.

Implementación de Orientación a Objetos en MDI

Si profundizamos en la arquitectura FiveWin, descubrirá algunas Clases poderosas que apoyan la implementación de ventanas MDI.

Hemos desarrollado tres Clases principales que heredan de la Clase base WINDOW:

```
TWindow
TMdiFrame
TMdiClient
TMdiChild
```

Todos implementan nuevos métodos para encapsular el poder que ofrece Windows a la programación MDI.

Recuerde esto:

TMdiFrame es la Ventana principal. Tiene un dato llamado oWndClient que es una Ventana TMdiClient.

oWndClient tiene un arreglo CA-Clipper con todas las ventanas hijas (child) que crea. El nombre del dato es awnd. Es un arreglo CA-Clipper normal. Úselo cuando desee.

Nuevas direcciones MDI en FiveWin

En versiones previas de FiveWin, pensamos desarrollar algunas Clases nuevas heredadas de TMdiChild para satisfacer algunas situaciones 'normales' de las aplicaciones comerciales que enfrentará. Pero ahora, hemos encontrado un acercamiento más fácil:

La mayor parte de las ventanas MDI hijas que usará serán muy similar unas a otras -una vez más enfatizamos en el desarrollo de aplicaciones comerciales programadas no 'fancy'-. Todas tendrán una ButtonBar atada a un control principal que se ajustara automáticamente al área de la ventana Hija.

Así, entregamos una manera poderosa, rápida y limpia de desarrollar ambientes MDI:

La manera que sugerimos para manejar ventanas MDI hijas es la siguiente:

- Primero, Cree el MDI Frame, Ver sintaxis más abajo.
- Luego Comience a crear las ventanas hijas, tantas como necesite:

Use la siguiente sintaxis:

```
DEFINE WINDOW [<oWnd>];
  [MDICHILD ];
  [FROM <nTop>, <nLeft> TO <nBottom>, <nRight> ];
  [TITLE <cTitle> ];
  [MENU <oMenu> ];
  [ICON <olco>];
  [OF <oParent>]
```

Ahora cree una ButtonBar para esta ventana hija. Use la siguiente sintaxis común:

```
DEFINE BUTTONBAR oBar OF <oWndChild> //Nombre de su MDI Child
```

Agregue los botones de la forma normal. Cada botón tendrá las acciones que defina:

```
DEFINE BUTTON OF oBar ACTION...
DEFINE BUTTON OF oBar ACTION...
...
```

Use este acercamiento fácil para reforzar la funcionalidad de la Ventana MDI Child sin necesidad desarrollar uno nuevo - está libre hacerlo, pero sólo cuando es muy necesario!-

El último paso es crear un Control - cualquier Control que necesite lo agrega a la Ventana hija:

```
@ 0, 0 GET <oControl> ... MEMO OF <oWndChild>  
@ 0, 0 LISTBOX <oControl> BROWSE ... OF <oWndChild>
```

Ahora, indicamos al MDI Child que use este control como su control principal:

```
<oWndChild>:SetControl( <oControl> )
```

Parta usando, ACTIVATE en la Ventana de la forma normal:

```
ACTIVATE WINDOW <oWndChild> ...
```

¡Y es Todo!

Al hacerlo de esta manera dominará la programación MDI en pocos minutos sin necesita de crear nuevas Clases o aprender el difícil tipo de programación MDI.

Permita que CA-Clipper y FiveWin haga el trabajo duro por usted. ¡Este es el poder de OOP y xBase!

Vea la nueva versión de .prg para un ejemplo real lleno de programación MDI. Cuando se activa la ventana.

Comandos de Dialogs

Dialogs:

Creando:

```
DEFINE DIALOG <oDlg>;  
  [NAME / RESNAME / RESOURCE <cResName> ] ;  
  [TITLE <cTitle> ] ;  
  [FROM <nTop>, <nLeft> TO <nBottom>, <nRight>] ;  
  [LIBRARY / DLL <hResources>] ;  
  [VBX ] ;  
  [STYLE <nStyle>] ;  
  [COLOR / COLORS <nClrText> [,<nClrBack> ] ;  
  [BRUSH <oBrush> ] ;  
  [WINDOW / DIALOG / OF <oWnd>] ;  
  [PIXEL] ;  
  [ICON <olco>] ;  
  [FONT <oFont>] ;  
  [HELP / HELPID <nHelpId>] ;
```

Activating:

```
ACTIVATE DIALOG <oDlg>;  
  [CENTER / CENTERED];  
  [NOWAIT / NOMODAL];  
  [WHEN <uWhen> ];  
  [VALID <uValid> ];  
  [ON [ LEFT] / CLICK <uClick>];  
  [ON INIT <ulnit> ];  
  [ON MOVE <uMoved>];  
  [ON PAINT <uPaint> ];
```

[ON RIGHT CLICK <uRClicked>]

- <oDlg> Variable que soporta el Objeto DialogBox. Es un objeto DialogBox.
- <cResName> Nombre del DialogBox guardado en un recurso. En un archiva RC o DLL. Normalmente se usa la opción 'Resname' de un editor de recursos para dar un nombre al elemento del recurso.
- <cTitle> Título del DialogBox. Puede ser modificado. Esto significa que puede usar el mismo DialogBox para diferentes propósitos sólo cambiando su título.
- <nTop>,
<nLeft>,
<nBottom>,
<nRight> Coordenadas del DialogBox, cuando lo crea desde código fuente.
- <hRes> Manejador optativo de un recurso DLL. Es útil cuando usa varios DLLs al mismo tiempo, por defecto usa el manejador de la aplicación cuando usa archivos RC o el manejador del DLL cuando usa SET RSOUCES TO
- <nStyle> Un estilo opcional de un DialogBox. Existen varios Ejemplos de como manejar este valor en el directorio SAMPLES. Si usa un DialogBox desde un recurso, deberá seleccionar el estilo desde el editor de recursos.
- <nClrText>,
<nClrBack> Colores Activos y de Fondo del DialogBox.
- <oBrush> Un Brush opcional para rellenar la superficie del DialogBox.
- <oWnd> Un objeto Contenedor opcional para el DialogBox. Por defecto usa la Window activa. Si muestra Controles DialogBox de mensajes debe especificar el contenedor.
- <olco> ICON Opcional a mostrar cuando el DialogBox este minimizado.
- <oFont> FONT opcional a usar para todo el DialogBox.
- <uWhen> Clausula WHEN que permite o no que DialogBox se Ejecute.
- <uValid> Clausula VALID para controlar el termino del DialogBox.
- <nHeipld> Identificador del Tópico de Ayuda del DialogBox.
- <uClick> Acción opcional a ejecutar cuando se hace click con el mouse en la superficie del DialogBox, los parámetros: nrow, ncol, nflags son pasados automáticamente.
- nRow Posición Fila (en pixels) donde el mouse fue pulsado.
 - nCol. Posición Columna (en pixels) donde el mouse fue pulsado.
 - nFlags. Flag numérico con información de teclas que se presionaron en el mismo momento.

Descripción de Valores:

- MK_CONTROL La tecla Control se Presiono.
- MK_MBUTTON Se presionó la tecla del medio del mouse.
- MK_RBUTTON Se presionó la tecla derecha del mouse.
- MK_SHIFT La tecla SHIFT se presiono.

```
#define MK_LBUTTON      0X0001
#define MK_RBUTTON    0x0002
#define MK_SHIFT      0x0004
```

```
#define MK_CONTROL      0X0008
#define MK_MBUTTON    0X0010
```

Los inspeccionará usando la función `!And()` de FiveWin:

if land(nFlags, MK_CONTROL)

if land(nFlags, MK_SHIFT, MK_CONTROL)

<uInit> Acción optativa a ejecutar cuando se crea el DialogBox pero aun no se muestra, En esta situación el DialogBox se comporta como una Ventana, podemos hacer todo tipo de funciones sobre el (como poner un ButtonBar)

<uMoved> Acción opcional a ejecutar cuando la DialogBox se mueve.

<uPaint> Acción opcional a ejecutar cuando la DialogBox se pinta.

<uRClicked> Acción opcional a ejecutar cuando se hace click con el botón derecho del mouse en la DialogBox. También usa tres parámetros como <uClick>.

Cláusulas:

VBX Para especificar que usara un control VBX en el DialogBox.

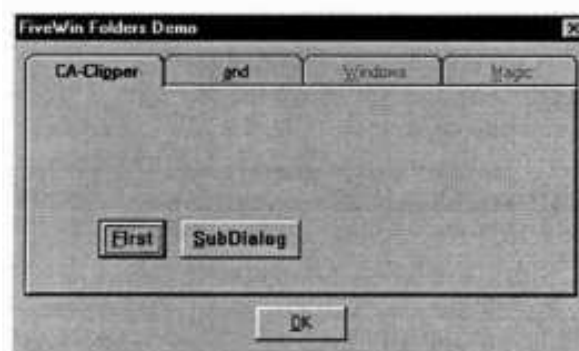
PIXEL Indica que las coordenadas del DialogBox están en PIXELs, solo cuando crea el DialogBox desde código fuente.

CENTER Indica que la DialogBox Aparecerá CENTRADA en la Ventana.

NOWAIT Indica que la DialogBox es NO-MODAL. es decir permite que otras ventanas no-modales trabajen al mismo tiempo, Debe tener especial cuidado con esto:

- Tenga cuidado con las variables locales: Si usa algunas locales recuerde que saldrá de la visibilidad de la función donde creo el DialogBox No-Modal y podría causar problemas. CA-Clipper trabaja perfectamente en estas circunstancias, ya que guarda referencias vivas de las variables locales dentro de codeblocks. Pero sólo éstos serán conocidos fuera de la función.
- La misma DialogBox se puede ejecutar varias veces, Debe informarse del Manejador de DBF. ¡En estas circunstancias su computadora se comporta como una red! Porque la DBF se abre varias veces. Es una buena idea comenzar a aprender programación en red, es muy fácil!
- Normalmente recomendamos que los programadores novicios no desarrollen aplicaciones no-modales en Windows. Cuando sus clientes usan más y más Windows, la programación no-modal es recomendada. Vea Programación MDI (en este manual) para un estilo de programación no-modal elegante y poderoso.

Folders:



Creando Folders desde Código Fuente

@ <nRow>, <nCol> FOLDER [<oFolder>];

```

[OF / WINDOW / DIALOG <oWnd>];
[PROMPT / PROMPTS / ITEMS <cPrompt .... >];
[DIALOG / DIALOGS / PAGE / PAGES <cDlgName1>];
[,<cDlgNameN>];
[PIXEL];
[DESIGN];
[COLOR / COLORS <nClrFore> [,<nClrBack>];
[OPTION <nOption> ];
[SIZE <nWidth>, <nHeight>];
[MESSAGE <cMsg> ]

```

Creando Folders desde Recursos

```

REDEFINE FOLDER [<oFolder>];
[ID <nId>];
[OF / WINDOW / DIALOG <oWnd>];
[PROMPT / PROMPTS / ITEMS <cPrompt,...> ];
[DIALOG / DIALOGS / PAGE / PAGES <cDlgName1> ];
[,<cDigNameN>] ];
[COLOR / COLORS <nClrFore> [,<nClrBack>] ];
[OPTION <nOption> ];
[ON CHANGE <uChange>]

```

<nRow>,
<nCol> Coordenadas donde pondremos el Control Folder (Solo desde código fuente).

<oFolder> Referencia al Objeto Folder que creara.

<oWnd> Objeto Contenedor del control Folder.

<cPrompt,...> Lista con el texto a desplegar en cada pagina.

<cDlg,...> Lista con los nombre de los recursos de las DialogBox a Manejar.

<nClrFore> Color para desplegar el Texto de los Prompts.

<nClrBack> Color para rellenar la superficie del folder.

<nOption> Opción inicial para usar diferentes From 1.

<nWidth>,
<nHeight> Dimensiones del control Folder.

<cMsg> Mensaje opcional a desplegar en el contenedor MsgBar si se definido.

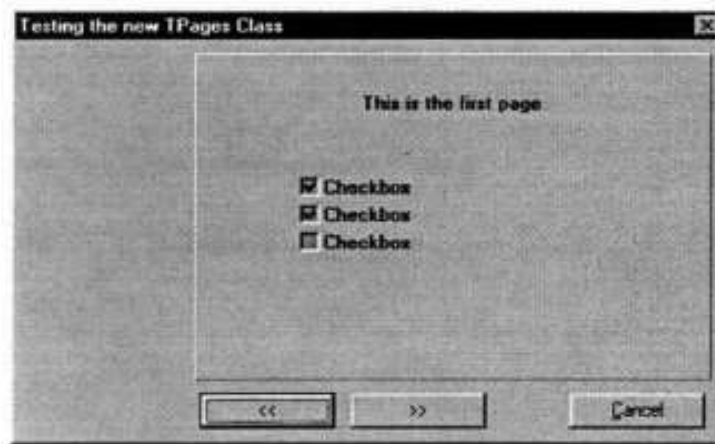
<nId> Numero identificador del Control.

<uChange> Acción a Ejecutar cuando cambie la pagina seleccionada en el folder.

Cláusulas:

PIXEL Para usar coordenadas pixel en vez de coordenadas texto.

DESIGN Usado para que el usuario pueda mover y reajustar interactivamente el control.

Pages:

Creándolos (Solo desde recursos):

```
REDEFINE PAGES <oPag>;  
  [ID <nId>];  
  [OF <oWnd> ];  
  [DIALOGS <DlgName,...> ];  
  [OPTION <nOption> ]
```

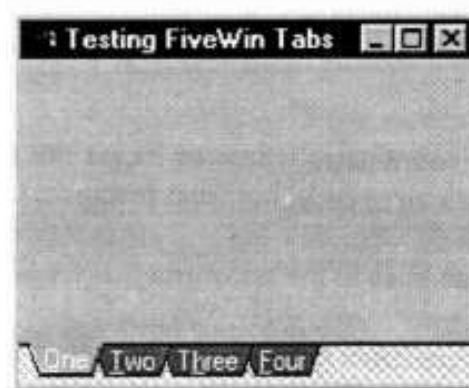
<oPag> Referencia al Objeto TPages que creara.

<nId> Identificador numérico del recurso del control.

<oWnd> Objeto contenedor de los Controles TPages.

<Dlg,...> Lista con los nombres de los recursos DialogBox a manejar. Importante: Estas dialogBox debe tener como estilo `STYLE WS_CHILD | WS_VISIBLE` desde archivo RC o DLL.

<nOption> Pagina inicial opcional a usar distinta de la 1.

Tabs:

Creando un TAB desde código fuente:

```
@ <nRow> , <nCol>TABS [<oTabs>];  
  [OF /WINDOW / DIALOG <oWnd>];  
  [PROMPT /PROMPTS / ITEMS <cPrompt,...>];  
  [ACTION /EXECUTE <uAction>];  
  [PIXEL ];  
  [DESIGNS ];  
  [COLOR / COLORS <nClrFore> [,<nClrBack> ]];  
  [OPTION <nOption> ];  
  [SIZE <nWidth>,<nHeight> ];  
  [MESSAGE <nMsg> ]
```

Creando un TAB desde recursos:

```
REDEFINE TABS [<oTabs>;
  [ID <nId>;
  [OF / WINDOW / DIALOG <oWnd> ];
  [PROMPT / PROMPTS / ITEMS <cPrompt,...> ];
  [ACTION / EXECUTE <uAction> ];
  [COLOR / COLORS <nClrFore> [,<nClrBack> ]];
  [OPTION <nOption>]
```

<nRow>, <nCol>	Coordenadas donde pondremos el control TAB (Solo desde código fuente).
<oTabs>	Referencia al objeto TAB que creara.
<oWnd>	Objeto contenedor de los Controles TAB.
<cPrompt>	Lista de String, uno para cada uno.
<uAction>	Acción a ejecutar cuando seleccione un TAB.
<nClrFore>	Color a Desplegar los prompts Texto.
<nClrBack>	Color para rellenar la superficie del TAB.
<nOption>	Opcional opción inicial distinta de 1.
<nWidth>, <nHeight>	Dimensiones de control TAB.
<cMsg>	Mensaje a desplegar en el MsgBox del contenedor si se definió.
<nId>	Identificador numérico del recurso del control

Cláusulas:

PIXEL	Usara coordenadas pixels en vez de coordenadas texto.
DESIGN	permite al usuario mover y reajustar interactivamente el control.

Comandos WinControls

Bar:



Creando una Barra de Botones desde código fuente:

```
DEFINE BUTTONBAR [ <oBar>;
  [SIZE / BUTTONSIZE / SIZEBUTTON <nWidth>, <nHeight>;
  [3D / 3DLOOK ];
  [TOP / LEFT / RIGHT / DOWN / FLOAT ];
  [OF / WINDOW / DIALOG <oWnd> ];
  [CURSOR <oCursor>]
```

Creando una Barra de Botones desde código fuente e indicando una posición específica:

```
@ <nRow>, <nCol> BUTTONBAR <oBar>;
```

```

[SIZE <nWidth>, <nHeight> ];
[BUTTONSIZE <nBtnWidth>, <nBtnHeight>];
[3D / 3DLOOK ];
[TOP / LEFT / RIGHT / BOTTOM / FLOAT ];
[OF / WINDOW / DIALOG <oWnd> ];
[CURSOR <oCursor> ]

```

Definiendo un Botón de la Barra de Botones.

```

DEFINE BUTTON [<oBtn>];
  [ OF / BUTTONBAR <oBar>];
  [ NAME / RESNAME / RESOURCE <cResName 1>;
  [, <cResName2> ] ];
  [ FILE / FILENAME / DISK <cBmpFile1>;
  [, <cBmpFile2> ] ];
  [ ACTION / EXEC <uAction> ];
  [ GROUP];
  [ MESSAGE <cMsg>];
  [ ADJUST ];
  [ WHEN <uWhen> ];
  [ TOOLTIP <cToolTip>]
  [ ON DROP <bDrop>]

```

Redefiniendo un botón puesto en un DialogBox en un Recurso:

```

REDEFINE BUTTON [<oBtn>];
  [ ID <nId> ];
  [ OF / BUTTONBAR <oBar>];
  [ NAME / RESNAME / RESOURCE <cResName1>
  [, <cResName2> ] ];
  [ FILE / FILENAME / DISK <cBmpFile 1> [, <cBmpFile2>]];
  [ ACTION / EXEC <uAction> ];
  [ MESSAGE <cMsg> ];
  [ ADJUST ];
  [ WHEN <uWhen> ];
  [ UPDATE <lUpdate> ];
  [ TOOLTIP <cToolTip>]

```

Crear un Botón Bitmap desde código fuente y ponerlo en una ventana contenedora:

```

@ <nRow>, <nCol> BUTTON [<oBtn>];
  [ NAME / RESNAME / RESOURCE <cResName1>
  [, <cResName2> ] ];
  [ FILE / FILENAME / DISK <cBmpFile1> [, <cBmpFile2> ] ];
  [ SIZE <nWidth>, <nHeight> ];
  [ ACTION <uAction> ];
  [ OF / WINDOW / DIALOG <oWnd> ];
  [ MESSAGE <cMsg> ];
  [ WHEN <uWhen> ];
  [ ADJUST ];
  [ UPDATE ]

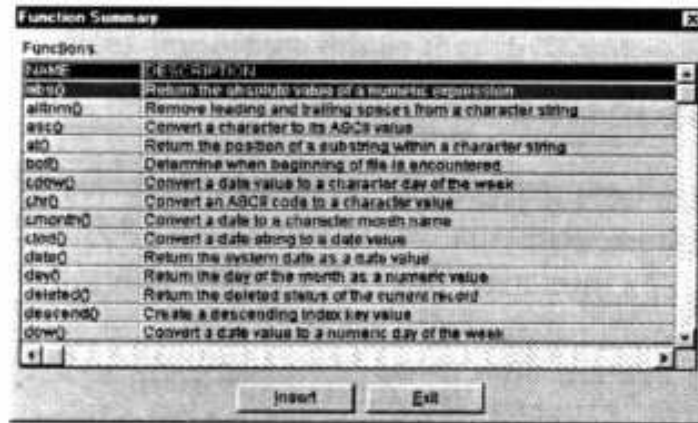
```

Comentario: También puede poner una buttonbar en una DialogBox. para hacerlo, sólo cree el buttonbar en la clausula ON INIT cuando active el DialogBox.

La clausula ON INIT de la DialogBox es similar a la de la ventana estándar.

<oBar> Referencia al Objeto Bar.

<nWidth>	El Alto y Ancho de los botones de la barra.
<nHeight>	Por defecto 28x28 pixels.
<oWnd>	Referencia a la ventana contenedora de la barra.
<oCursor>	Referencia al objeto Cursor del Mouse a usar en la barra.
<oBtn>	Referencia al Objeto BitmapButton.
<cBmp1>	Nombre del BitMap a desplegar. Puede especificar un segundo bitmap para animación
<cBmp2>	Esto es usado para acceder a BitMaps almacenados en Disco Duro.
<cRes1>	Nombre del recurso del Bitmap a desplegar. Puede especificar un segundo Bitmap para
<cRes2>	animación. Esto se usa para acceder a Bitmaps almacenados en archivos de recursos.
<uAction>	Acción a ejecutar cuando se presione el Botón.
<cMsg>	Mensaje a desplegar en la barra de mensajes de la ventana contenedora del objeto Bar que contiene el BitMap Button.
<uWhen>	Expresión que indica si el Botón de la Barra esta habilitado o no.
<nId>	Identificador para el Botón puesto en la DialogBox.
<cToolTip>	Texto a Mostrar asociado al tooltip flotante.
<bDrop>	Acción a ejecutar Cuando se haga Drop.
Cláusulas:	
3D	
3DLOOK	Para desplegar el botón con efecto 3D.
TOP	
LEFT	
RIGHT	
DOWN	Posición de la Barra de Botones respecto a la ventana contenedora. Al hacer click con el botón derecho aparece un popup que permite seleccionar dinámicamente una nueva posición.
FLOAT	Muestra la Barra de Botones flotando en la ventana.
Para los botones en la Barra de Botones.	
ADJUST	Ajusta automáticamente el bitmap al tamaño del Botón.
GROUP	Comienza un nuevo "GRUPO" de botones en la ButtonBar. Puede tener tantos como necesite.
UPDATE	Para los BitMapButtons puestos en una DialogBox (sin una buttonbar). El botón actualizará su imagen automáticamente si se define oDlg:Update().

Browsers:

Definiendo un Objeto Browser desde código Fuente:

```
@ <nRow>, <nCol> LISTBOX [ <oBrw> ];
  FIELDS <cFields,...>;
  [ ALIAS <cAlias>;
  [ FIELDSIZES / SIZE / COLSIZES <aColSizes,...> ];
  [HEAD / HEADER / HEADERS / TITLE <aHeaders....> ];
  [ID <nId> ];
  [ SIZE <nWidth>, <nHeight>;
  [ SELECT <cField> FOR <uValue1> [ TO <uValue2>]];
  [ ON CHANGE <uChange> ];
  [ ON RIGHT CLICK <uRClick> ];
  [ ON [LEFT] CLICK <uLClick>;
  [ ON [LEFT] DBLCLICK <uDbIClick>;
  [ FONT <oFont>;
  [ CURSOR <oCursor>;
  [ COLOR / COLORS <nCirFore> [,<nClrBack>]];
  [ MESSAGE <cMsg>;
  [UPDATE];
  [ PIXEL ];
  [ WHEN <uWhen>;
  [ DESIGN ];
  [VALID <ValidFunc> ];
  [ACTION <uAction,...> ];
  [OF | DIALOG <oDlg>]
```

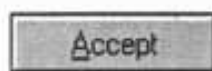
Redefiniendo un Objeto Browser desde recursos:

```
REDEFIN E LISTBOX [ <oBrw>;
  [FIELDS [ <cFields,...>;
  [ALIAS <cAlias>;
  [ID <nId>;
  [FIELDSIZES / SIZES / COLSIZES <aColSizes ....>];
  [HEAD / HEADER / HEADERS <aHeaders,...> ];
  [SELECT <cField> FOR <uValue1> [ TO <uValue2>]];
  [ON CHANGE <uChange> ];
  [ON RIGHT CLICK <uRClick> ];
  [ON [LEFT] CLICK <uLClick> ];
  [ON [LEFT] DBLCLICK <uDbIClick>
  [FONT <oFont> ];
  [CURSOR <oCursor>;
  [COLOR / COLORS <nCirFore> [,<nClrBack>]];
  [MESSAGE <cMsg>;
```

```
[UPDATE];
[WHEN <uWhen>];
[VALID <ValidFunc> ];
[ACTION <uAction,...> ] ;
[OF / WINDOW / DIALOG <oDlg>]
```

- <oBrw> Referencia al Objeto TWBrowse.
- <nRow>
<nCol> Posición en la ventana contenedora, Simula coordenadas texto.
- <cFields> Lista de expresiones texto a desplegar en las columnas, Si una expresión es numérica, El manejador de TWBrowser asume que se trata de un bitmap a desplegar.
- <aColSizes> Lista con el tamaño de cada columna en Pixels.
- <aHeaders> Lista de expresiones para cabeceras.
- <nId> Identificador numérico para el control TWBrowser.
- <nWidth>
<nHeight> Tamaño del control TWbrowse en pixels.
- <uChange> Acción a ejecutar cuando seleccione una nueva Fila.
- <uRClick> Acción a ejecutar cuando haga click con el botón derecho.
- <uLClick> Acción a ejecutar cuando haga click con el botón izquierdo.
- <uDbClick> Acción a ejecutar cuando haga doble click con el botón izquierdo.
- <oFont> Referencia al Objeto Font usado con el LISTBOX.
- <ValidFunc> Expresión o Función para validar cuando el LISTBOX pierda el Focus.
- <nClrFore> Color activo como un numero RGB, o como String xBase.
- <nClrBack> Color de Fondo como un numero RGB.
- <cField> Indica el Campo que marca el INDEX activo.
- <oDlg> Objeto que contiene al control TWBrowse.
- Cláusulas:
- DESIGN Permite que el usuario lo mueva y reajuste mediante el mouse.
- PIXEL Para usar coordenadas Pixel en vez de coordenadas Texto.

Botones:



Creando Botones desde Código Fuente

```
@<nRow>, <nCol> BUTTON <oBtn>
  [PROMPT <cCaption> ] ;
  [ NAME / RESNAME / RESOURCE <cResName1> ] ;
  [, <cResName2> ] ] ;
```

```

[ FILE / FILENAME / DISK <cBmpFile1>;
[,<cBmpFile2> ] ];
[SIZE <nWidth>, <nHeight> ] ;
[ACTION <uAction> ] ;
[OF / WINDOW / DIALOG <oWnd> ] ;
[MESSAGE <cMsg> ] ;
[WHEN <WhenFunc> ] ;
[ADJUST ] ;
[DEFAULT];
[HELP / HELPID / HELP ID <nHelpId> ] ;
[FONT <oFont> ] ;
[PIXEL ] ;
[DESIGN ] ;
[UPDATE];
[VALID <ValidFunc> ] ;
[CANCEL]

```

Usando Botones desde recursos:

```

REDEFINE BUTTON [<oBtn>];
  [ ID <nId> ] ;
  [ PROMPT <cCaption>] ;
  [ OF / BUTTON BAR <oBar>] ;
  [ / WINDOW / DIALOG <oDlg> ] ;
  [ NAME / RESNAME / RESOURCE <cResName 1>] ;
  [,<cResName2>] ] ;
  [ FILE / FILENAME / DISK <cBmpFile>;
  [,<cBmpFile2>] ] ;
  [ACTION / EXEC <uAction>] ;
  [MESSAGE <cMsg>] ;
  [ADJUST ] ;
  [WHEN <WhenFunc>] ;
  [HELP / HELPID/HELP ID <nHelpId>] ;
  [ VALID <ValidFunc>] ;
  [CANCEL]

```

```

DEFINE BUTTON [ <oBtn>] ;
  [ OF / BUTTONBAR <oBar>] ;
  [ NAME / RESNAME / RESOURCE <cResName 1>;
  [,<cResName2>] ] ;
  [ FILE / FILENAME / DISK <cBmpFile1>;
  [,<cBmpFile2>] ] ;
  [ ACTION / EXEC <uAction>] ;
  [ GROUP ] ;
  [ MESSAGE <cMsg>] ;
  [ ADJUST ] ;
  [ WHEN <WhenFunc>]

```

<nRow>,
 <nCol> Coordenadas del botón, simula coordenadas Texto.

<oBtn> Referencia al Objeto Button.

<oBar> Referencia al Objeto Barra (Bar).

<cCaption> Etiqueta del Botón.

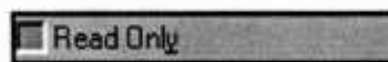
<cBmp1> Nombre del archivo Bitmap que usar como imagen el Botón.

<cBmp2>	Cuando un segundo Bitmap se especifica, se desplegará cuando el botón este presionado.
<nHelpId>	Tópico de Ayuda (help topic) que identifica al Botón.
<oWnd>	Ventana contenedora del Botón.
<uAction>	Acción a ejecutar cuando el botón se presiona.
<oFont>	Referencia al objeto Font usado con el Botón.
<nWidth>	Dimensión del botón.
<nHeight>	
<cRes1>	Recurso Bitmap a desplegar como imagen del Botón.
<cRes2>	Cuando un segundo recurso Bitmap es definido este se mostrará cuando el botón es presionado.
<cMsg>	Mensaje asociado a este control.
<WhenFunc>	Expresión a evaluar para indicar si el control está o no habilitado.
<nId>	Identificador numérico del recurso del Control.
<ValidFunc>	Expresión para validar cuando el control pierda el Focus.

Cláusulas:

DEFAULT	PushButtons por Defecto: Se ejecuta automáticamente cuando el usuario Presiona Enter, simula a otros controles. Use esta cláusula solo cuando construya el botón desde código fuente. Cuando usa recursos, esta cláusula debe ser especificada en el Editor de Recursos.
DESIGN	Permite que el usuario lo Mueva y Reajuste usando el Mouse.
PIXEL	Usa coordenadas Pixels en vez de coordenadas Texto.
UPDATE	Este control será actualizado desde el Método TDialog:Update().

CheckBox:



Creando CheckBox desde Código Fuente:

```
@<nRow>, <nCol> CHECKBOX [ <oCbx> VAR <IVar>;  
  [ PROMPT <cCaption> ];  
  [ OF / WINDOW / DIALOG <oWnd> ];  
  [ SIZE <nWidth>, <nHeight> ];  
  [ HELPID / HELP ID <nHelpId>];  
  [ FONT <oFont> ];  
  [ ON CLICK / ON CHANGE <uClick>];  
  [ VALID <ValidFunc> ];  
  [ COLOR / COLORS <nClrFore>;  
  [, <nClrBack>]];  
  [ DESIGN ];
```

```
[ PIXEL ];
[ MESSAGE <cMsg>];
[ UPDATE];
[ WHEN <WhenFunc> ]
```

Usando CheckBox desde recursos:

```
REDEFINE CHECKBOX [ <oCbx> VAR ] <lVar>;
  [ ID <nId> ];
  [ OF / WINDOW / DIALOG <oWnd>];
  [ HELPID / HELP ID <nHelpId> ];
  [ ON CLICK / ON CHANGE <uClick> ];
  [ VALID <ValidFunc> ];
  [ COLOR / COLORS <nClrFore>;
  [, <nClrBack> ] ];
  [ MESSAGE <cMsg>];
  [ UPDATE];
  [ WHEN <WhenFunc> ]
```

<nRow>, <nCol>	Coordenadas del CheckBox. Imitan coordenadas Texto.
<oCbx>	Referencia al Objeto CheckBox.
<lVar>	Variable Lógica que será manejada por el CheckBox.
<cCaption>	Etiqueta del Check Box.
<oWnd>	Ventana que contiene el CheckBox.
<nWidth>, <nHeight>	Dimensiones del CheckBox en Pixels
<nHelpId>	El tópico de Ayuda (help topic) asociado al CheckBox.
<oFont>	Referencia al Objeto Font usado con el CheckBox.
<uClick>	Acción a ejecutar cuando el CheckBox Cambie.
<ValidFunc>	Expresión o Función para validar cuando el CheckBox pierda el Focus.
<nClrFore>	Color Activo como un numero RGB o como string xBase.
<nClrBack>	Color de fondo como numero RGB.
<cMsg>	Mensaje asociado con este Control.
<WhenFn>	Expresión When como Bloque de Código.
<nId>	Identificador numérico del recurso del Control.

Cláusulas:

DESIGN	Permite al usuarios mover y reajustar el control con el mouse.
PIXEL	Usa coordenadas Pixels en vez de coordenadas Texto.
UPDATE	Este control será actualizado desde el método TDialog:Update().

ComboBox:

Creando un ComboBox desde Código Fuente:

```
@ <nRow>, <nCol> COMBOBOX [ <oCbx> VAR ] <cVar>;
  [ ITEMS / PROMPTS <altems> ] ;
  [ SIZE <nWidth>, <nHeight> ] ;
  [ OF / WINDOW / DIALOG <oWnd> ] ;
  [ HELPID / HELP ID <nHelpId> ] ;
  [ ON CHANGE <uChange> ] ;
  [ VALID <ValidFunc> ] ;
  [ COLOR / COLORS <nClrText>;
  [,<nClrBack>] ] ;
  [ PIXEL ] ;
  [ FONT <oFont> ] ;
  [ UPDATE ] ;
  [ MESSAGE <cMsg> ] ;
  [ WHEN <WhenFunc> ] ;
  [ DESIGN ] ;
  [ BITMAPS <cBitmaps> ] ;
  [ ON DRAWITEM <uBmpSelect> ]
```

Creando un ComboBox desde Recursos:

```
REDEFINE COMBOBOX [ <oCbx> VAR <cVar>;
  [ ITEMS / PROMPTS <altems> ] ;
  [ ID <nId> ] ;
  [ OF / WINDOW / DIALOG <oWnd> ] ;
  [ HELPID / HELP ID <nHelpId> ] ;
  [ ON CHANGE <uChange> ] ;
  [ VALID <ValidFunc> ] ;
  [ COLOR / COLORS <nClrText>;
  [,<nClrBack>]
  [ UPDATE ] ;
  [ MESSAGE <cMsg> ] ;
  [ WHEN <WhenFunc> ] ;
  [ BITMAPS <acBitmaps> ] ;
  [ ON DRAWITEM <uBmpSelect> ] ;
  [ STYLE <nStyle> ] ;
  [ PICTURE <cPicture> ] ;
  [ ON EDIT CHANGE <uEChange> ]
```

<nRow>, <nCol>	Coordenadas del ComboBox, imitan coordenadas texto.
<oCbx>	Referencia al Objeto Combo.
<cVar>	Variable de carácter que contiene el ítem actualmente seleccionado en el ComboBox.
<altems>	Ítems mostrados en el ComboBox.
<nWidth>, <nHeight>	Dimensiones del ComboBox en Pixels.
<oWnd>	Ventana Contenedora del ComboBox.

<nHelpId>	Tópico de Ayuda (help topic) asociado al Control.
<uChange>	Acción a ejecutar cada vez que el ítem seleccionado cambie.
<cMsg>	Mensaje asociado a este Control.
<ValidFunc>	Expresión a Validar cuando el control pierda el Focus.
<nClrText>	Color activo como un número RGB, o como un string xBase.
<nClrBack>	Color de Fondo definido similar a nClrText.
<oFont>	Referencia al Objeto Font a usar con el ComboBox.
<WhenFun>	Expresión para evaluar la Clausula en un Bloque de Código.
<nId>	Identificador numérico del recurso del control.
<cBitmaps>	Lista con archivos o recursos Bitmaps a usar en un ownerdraw ComboBox.
<uBmpSelt>	Bloque de Código para seleccionar un bitmap pintado en un ownerdraw ComboBox. Por defecto, el usado corresponde al índice de la lista de BitMaps.
<nStyle>	Para especificar el estilo dropdown en el ComboBox. Use: STYLE CBS_DROPDOWN.
<cPicture>	Picture a usar en el contenedor DATA cuando se usa el comboBox con DropDown, si se especifica STYLE CBS_DROPDOWN.
<uEChange>	Expresión a evaluar cuando el oGet DATA cambie si se especifico STYLE CBS_DROPDOWN.

Cláusulas:

PIXEL	Para usar coordenadas pixel en vez de coordenadas texto .
UPDATE	Permite actualizar este control desde el método TDialog:Update().
DESIGN	Permite que el usuario mueva y reajuste el control con el mouse.

Gets:

Creando un Get desde código fuente:

```
@<nRow>, <nCol> GET [ <oGet> VAR ] <uVar>;
  [ OF / WINDOW / DIALOG <oWnd> ] ;
  [ PICTURE <cPict> ] ;
  [ VALID <ValidFunc> ] ;
  [ WHEN <WhenFunc> ] ;
  [ COLOR / COLORS <nClrFore>;
    [, <nClrBack> ] ] ;
  [ SIZE <nWidth>, <nHeight> ] ;
  [ FONT <oFont> ] ;
  [ HSCROLL ] ;
  [ CURSOR <oCursor> ] ;
  [ PIXEL ] ;
  [ MESSAGE <cMsg> ] ;
  [ UPDATE ] ;
  [ WHEN <WhenFunc> ] ;
```

```

[CENTER / CENTERED];
[RIGHT ] ;
[READONLY / NO MODIFY ] ;
[DESIGN ] ;
[ON CHANGE <uChange> ] ;
[PASSWORD ] ;

```

Para editar Memos:

```

@ <nRow>, <nCol> GET [ <oGet> VAR <uVar>];
[OF / WINDOW / DIALOG <oWnd> ] ;
[MULTI LINE / MEMO / TEXT ] ;
[COLOR / COLORS <nClrFore>;
[,<nClrBack>] ] ;
[SIZE <nWidth>, <nHeight>];
[FONT <oFont>];
[DESIGN ] ;
[CURSOR <oCursor>];
[PIXEL ] ;
[MESSAGE <cMsg>];
[UPDATE];
[WHEN <WhenFunc> ] ;
[CENTER / CENTERED];
[RIGHT ] ;
[ON CHANGE <uChange> ] ;
[HSCROLL];
[READONLY / NO MODIFY ] ;
[VALID <ValidFunc>];

```

Usando GET desde una recurso DialogBox.

```

REDEFINE GET [ <oGet> VAR ] <uVar>;
[ID <nId> ] ;
[OF / WINDOW / DIALOG <oWnd> ] ;
[ HELPID / HELP ID <nHelpId> ] ;
[ VALID <ValidFunc>];
[ PICTURE <cPict> ] ;
[ COLOR / COLORS <nClrFore>;
[,<nClrBack>] ] ;
[ FONT <oFont>];
[ CURSOR <oCursor> ] ;
[ MESSAGE <cMsg> ] ;
[UPDATE];
[ WHEN <WhenFunc>];
[ ON CHANGE <uChange>];
[ READONLY / NO MODIFY];

```

Para editar Memos:

```

REDEFINE GET [ <oGet> VAR ] <uVar>;
[MULTILINE / MEMO / TEXT];
[ID <nId> ] ;
[OF / WINDOW / DIALOG <oWnd> ] ;
[HELPID/ HELP ID <nHelpId> ] ;
[COLOR / COLORS <nClrFore> [,<nClrBack>] ] ;
[FONT <oFont> ] ;
[CURSOR <oCursor> ] ;
[MESSAGE <cMsg> ] ;

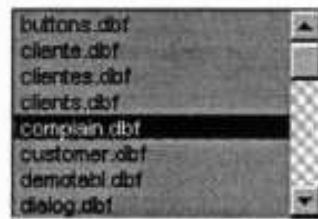
```

[UPDATE];
[WHEN <WhenFunc>];
[READONLY / NO MODIFY];
[VALID <ValidFunc>];
[ON CHANGE <uChange>];

<nRow>, <nCol>	Coordenadas del GET, simula coordenadas texto.
<oGet>	Referencia al Objeto GET.
<uVar>	Variable para manipular el Objeto GET.
<oWnd>	Ventana Contenedora del GET.
<cPict>	Clausula típica de PICTURE xBase.
<ValidFun>	Expresión o Función de Validación, similar a DOS
<WhenFun>	Clausula When. No disponible aun.
<cColor>	Color del GET, no disponible aún
<nWidth>, <nHeight>	Dimensiones en Pixels del GET
<oFont>	Referencia al Objeto FONT usado en el GET.
<oCursor>	Referencia al Objeto CURSOR a desplegar sobre este control.
<cMsg>	Mensaje asociado a este control.
<nId>	Identificador numérico del recurso de este control.
<nHelpId>	Identificador del tópico de ayuda para el recurso GET.
<nClrFore>	Color activo como un numero RGB o como string xBase.
<nClrBack>	Color de fondo como un numero RGB.
<uChange>	Acción a ejecutar cuando cambie el contenido del GET.
Cláusulas	
MULTILINE MEMO TEXT	Usa un GET de Múltiples Líneas para un memo. editing.
UPDATE	Este control puede actualizarse desde el método TDialog:Update().
CENTER	El contenido del GET aparece Centrado.
RIGHT	El contenido del GET aparece justificado a la Derecha.
Importante:	Para justificar a la derecha un GET en un DialogBox, debe definirlo como MultiLine y Right, además de usar el Set de Caracteres Estándar.
PIXEL	Para usar coordenadas Pixels en vez de Coordenadas Texto.
PASSWORD	Ingreso con una PassWord.

DESIGN	Permite a usuario mover y reajustar el control con el mouse.
HSCROLL	Scroll Horizontal.
READONLY NO MODIFY	Deshabilita la edición en este Control.

ListBoxes



Creando un ListBox desde código fuente.

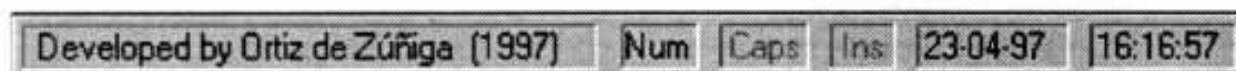
```
@ <nRow>, <nCol> LISTBOX [ <oLbx> VAR <cnVar>;
  [ ITEMS / PROMPTS <aList>;
  [ SIZE <nWidth>, <nHeight>;
  [ ON CHANGE <uChange> ];
  [ ON [LEFT] DBLCLICK <uLDbIClick> ] ;
  [ OF / WINDOW / DIALOG <oWnd> ] ;
  [ VALID <uValid> ] ;
  [ COLOR / COLORS <nClrFore> [,<nClrBack>]] ;
  [ PIXEL ] ;
  [ DESIGN ] ;
  [ FONT <oFont> ] ;
  [ MESSAGE <cMsg> ] ;
  [ UPDATE ] ;
  [ WHEN <uWben> ] ;
  [ BITMAPS <acBitmaps> ] ;
  [ ON DRAWITEM <uBmpSelect> ] ;
  [ MULTI / MULTIPLE / MULTISEL ] ;
  [ SORT ] ;
```

Creando un ListBox desde recursos:

```
REDEFINE LISTBOX [ <oLbx> VAR ] <cnVar>;
  [ ITEMS / PROMPTS <altems> ] ;
  [ FILES / FILESPEC <cFileSpec> ] ;
  [ ID <nId> ] ;
  [ ON CHANGE <uChange> ] ;
  [ ON [LEFT] DBLCLICK <uLDbIClick> ] ;
  [ OF / WINDOW / DIALOG <oWnd> ] ;
  [ HELPID / HELP ID <nHelpId> ] ;
  [ BITMAPS [ SIZE <nSize> ] ] ;
  [ VALID <uValid> ] ;
  [ COLOR / COLORS <nClrFore> [,<nClrBack>] ] ;
  [ MESSAGE <cMsg> ] ;
  [ UPDATE ] ;
  [ WHEN <uWhen> ] ;
  [ BITMAPS <acBitmaps> ] ;
  [ ON DPAWITEM <uBmpSelect> ] ;
```

<nRow>,

<nCol>	Coordenadas del ListBox, imita coordenadas Texto.
<oLbx>	Referencia al Objeto ListBox.
<cnVar>	Variable Clipper, string o numérica, donde FiveWin automáticamente dejara el ítem seleccionado en el ListBox.
<aList>	Arreglo Clipper de strings, usado para llenar los ítems del ListBox.
<nWidth>, <nHeight>	Dimensiones del ListBox en Pixels.
<uChange>	Bloque de Código a evaluar cada vez que se selecciona un ítem en el ListBox.
<uDbIcClick>	Bloque de código a evaluar cuando el usuario hace doble click con el botón izquierdo del mouse sobre el ListBox.
<oWnd>	Referencia al objeto contenedor del ListBox, normalmente una Window o una DialogBox.
<uValid>	Bloque de código para la Clausula VALID del ListBox.
<nClrFore>	Color del texto en los ítems del ListBox.
<nClrBack>	Color de fondo para los Items del ListBox.
<oFont>	Objeto FONT opcional para los items del ListBox.
<cMsg>	Mensaje a mostrar en la messagebar del contenedor del ListBox.
<uWhen>	Bloque de código para la clausula WHEN del ListBox.
<cFileSpec>	Mascara usada para mostrar archivos desde disco, Ejemplo: "*. *".
<nId>	Identificador numérico del recurso de control.
<nHelpId>	Identificador de ayuda para el ListBox.
<nSize>	Tamaño opcional para el bitmap usado en el dibujo del ListBox.
Cláusulas:	
PIXEL	Para utilizar coordenadas pixels en vez de coordenadas texto.
DESIGN	Permite que el usuario mueva y reajuste el control con el mouse.
UPDATE contenedor.	Este control se actualizará con el método <oDlg>:Update(), donde oDlg es el objeto
MULTI MULTIPLE MULTISEL	Permite múltiple selección.
SORT	Ordena los elementos del ListBox.

Barra de Mensajes:

Creando una MsgBox desde código fuente (formato xBase tradicional):

```
SET MESSAGE OF <oWnd>;
  TO <cPrompt> ;
  [ CENTERED];
  [FONT <oFont>];
  [CLOCK / TIME];
  [DATE];
  [KEYBOARD];
  [ COLOR / COLORS <nClrFore>;
  [,<nClrBack>];
  [ NOINSET ]
```

Alternativa de uso (Creación de Objetos):

```
DEFINE MESSAGE OF <oWnd>;
  TO <cPrompt> ;
  [ PROMPT <cMsg>]
  [CENTERED];
  [ FONT <oFont>];
  [ CLOCK / TIME>];
  [DATE];
  [KEYBOARD];
  [ COLOR / COLORS <nClrFore>;
  [,<nClrBack>]]
  [ NOINSET]
```

definiendo sus propios MsgItems:

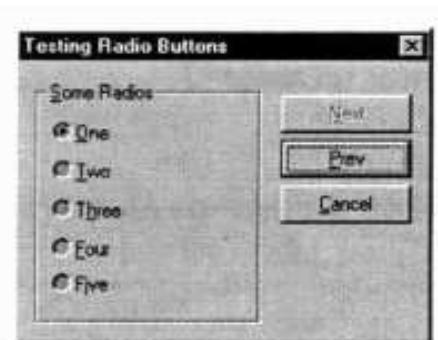
```
DEFINE MSGITEM [ <oMsgItem> ];
  [OF <oMsgBar>];
  [PROMPT <cPrompt>];
  [SIZE <nSize>];
  [FONT <oFont>];
  [COLOR / COLORS <nClrFore>;
  [, <nClrBack>];
  [ ACTION <uAction>];
```

<oWnd>	Referencia al Objeto Contenedor.
<cPrompt>	Mensaje a mostrar por defecto.
<oFont>	Referencia opcional a un objeto FONT para desplegar el mensaje.
<nClrFore>	Color activo como numero RGB, o String xBase.
<nClrBack>	Color de fondo como un numero RGB.
<oMsgItem>	Referencia al objeto MsgItem.
<oMsgBar>	Referencia al contenedor MsgBox.
<nSize>	Tamaño del MsgItem.

Cláusulas:

CENTERED Para mostrar centrado el texto.

CLOCK TIME.	Muestra la hora.
DATE	Muestra la Fecha.
KEYBOARD	Muestra el estado del teclado.
NOINSET	Define apariencia no-inset en la MsgBar.

RadioButtons:

Creando un RadioButton Menu desde código fuente.

```
@ <nRow>, <ncol> RADIO [ <oRadMenu> VAR <nVar>;
  [ PROMPT / ITEMS <cltems .... >;
  [ OF / WINDOW / DIALOG <oWnd > ];
  [ HELPID / HELP ID <nHeipld,...> ];
  [ ON CLICK / ON CHANGE <uChange>
  [ COLOR / COLORS <nClrFore>;
  [,<nClrBack>]];
  [ MESSAGE <cMsg>;
  [UPDATE];
  [ WHEN <uWhen > ];
  [ SIZE <nWidth>, <nHeight>;
  [ VALID <ValidFunc > ];
  [ DESIGN];
  [3D];
  [ PIXEL];
```

Creando un RadioButton Menu desde recursos:

```
REDEFINE RADIO [ <oRadMenu> VAR ] <nVar>;
  [ ID <nId .... >;
  [ OF / WINDOW / DIALOG <oWnd > ];
  [ HELPID / HELP ID <nHeipld,...> ];
  [ ON CLICK / ON CHANGE <uChange>
  [ COLOR / COLORS <nClrFore>;
  [,<nClrBack>] ];
  [ MESSAGE <cMsg>;
  [UPDATE];
  [WHEN <uWhen>;
  [ VALID <ValidFunc>;
```

<nRow>, <nCol>	Coordenadas del RadioButton Menu. Simula coordenadas texto.
<oRadMen>	Referencia al Objeto RadMenu.
<nVar>	Variable numérica con el control del RadioButton Menu.

<cltems>	Lista de Strings para definir los prompts usados como elementos radio.
<oWnd>	Referencia a la ventana contenedora del Radio Button Menu.
<nHelpId>	Lista de identificadores de Ayudas para cada control radio.
<uChange>	Acción a ejecutar cuando se selecciona un radio en el RadioButton Menu.
<cMsg>	Mensaje asociado a este control.
<WhenFun>	Expresión a evaluar para indicar si el control esta o no habilitado.
<nWidth>	
<nHeigth>	Dimensiones del Control.
<ValidFunc>	Expresión para evaluar cuando el control pierde el Focus.

Cláusulas:

3D	
3DLOOK	Apariencia 3D.
UPDATE	este control puede ser actualizado con el método TDialog:Update().
DESIGN	el usuario puede mover o reajustar el control con el mouse.
PIXEL	Usa coordenadas Pixels en vez de coordenadas Texto.

Says

Definiendo un Objeto SAY desde código fuente:

```
@ <nRow>, <nCol> SAY [ <oSay> PROMPT/ VAR] <cText>;
  [OF / WINDOW / DIALOG <oWnd>];
  [PICTURE <cPicture> ] ;
  [COLOR / COLORS <nClrFore>;
  [,<nClrBack>] ] ;
  [ FONT <oFont> ] ;
  [CENTERED / CENTER];
  [RIGHT ] ;
  [BORDER];
  [PIXEL / PIXELS]
  [SIZE <nWidth>, <nHeight>];
  [DESIGN ];
  [UPDATE];
  [SHADED / SHADOW];
  [BOX];
  [RAISED];
```

Redefiniendo un Objeto SAY desde recursos:

```
REDEFINE SAY <oSay>;
  [PROMPT <cText> ] ;
  [ID <nId> OF / WINDOW / DIALOG <oDlg> ] ;
  [CENTERED]
  [ COLOR / COLORS <nClrFore>;
  [,<nClrBack>]];
```

[PICTURE <cPicture>];
[UPDATE];
[FONT <oFont>];

<nRow>, <nCol>	Coordenadas del Objeto SAY. imita coordenadas Texto.
<oSay>	Referencia al Objeto SAY.
<cText>	Etiqueta o Texto del Objeto SAY.
<oWnd>	Dialog contenedor del control SAY.
<cPicture>	Típica Clausula PICTURE de un SAY.
<cColor>	String con el COLOR para el SAY. No habilitada aún.
<oFont>	Objeto Font para usar con el SAY. Debe ser creado antes usando DEFINE FONT....
<nClrFore>	Color activo como numero RGB, o por un string xBase.
<nClrBack>	Color de Fondo como numero RGB.
<nWidth>, <nHeight>	Dimensiones de este Control.
<nId>	Identificador numérico del recurso del objeto SAY. es asignado desde el editor de recursos.

Cláusulas:

CENTERED CENTER	Centra el Texto.
RIGHT	Justificación a la derecha.
BORDER	Crea un borde alrededor del Control.
DESIGN	Permite que el usuario mueva y reajuste el control con el mouse.
PIXEL	Usa coordenadas Pixels en vez de coordenadas Texto.
UPDATE	Este control puede ser utilizado usando el método TDialog:Update().
SHADED SHADOW	Con sombra.
BOX	Pone el Objeto SAY en un recuadro.
RAISED	Pone el Objeto SAY en un recuadro con sombra.

Scrollbars:

Crea una ScrollBar desde código fuente:

@<nRow>, <nCol> SCROLLBAR [<oSbr>];
[HORIZONTAL / VERTICAL];

```

[RANGE <nMin>, <nMax> ];
[PAGESTEP <nPgStep>];
[SIZE <nWidth>, <nheight>];
[UP | ON UP <uUpAction>];
[DOWN / ON DOWN <uDownAction>];
[PAGEUP / ON PAGEUP <uPgUpAction>];
[PAGEDOWN | ON PAGEDOWN <uPgDownAction>];
[ON THUMBPOS <uPos> ];
[PIXEL ];
[COLOR / COLORS <nClrFore>;
,<nClrBack> ]];
[MESSAGE <cMsg>];
[UPDATE];
[WHEN <WhenFunc> ];
[VALID <ValidFunc> ];
[DESIGN ];
[OF / WINDOW / DIALOG <oDlg>]

```

Crea una ScrollBar desde recursos:

```

REDEFINE SCROLLBAR [ <oSbr> ];
[ ID <nID> ];
[ RANGE <nMin>, <nMax> ];
[ PAGESTEP <nPgStep> ];
[ UP / ON UP / ON LEFT <uUpAction>];
[ DOWN / ON DOWN / ON RIGHT <uDownAction>];
[ PAGEUP / ON PAGEUP <uPgUpAction> ] ;
[ PAGEDOWN / ON PAGEDOWN <uPgDownAction>];
[ ON THUMBPOS <uPos> ];
[ COLOR / COLORS <nClrFore>;
,<nClrBack> ]];
[ OF / WINDOW / DIALOG <oDlg>]
[ MESSAGE <cMsg> ];
[UPDATE];
[WHEN <WhenFunc>];
[VALID <ValidFunc> ];

```

Para "ScrollBars falsas" (Cuando usa estilos WS_VSCROLL o WS_HSCROLL):

```

DEFINE SCROLLBAR [ <oSbr> ];
[ HORIZONTAL / VERTICAL ] ;
[ RANGE <nMin>, <nMax> ] ;
[ PAGESTEP <nPgStep> ] ;
[ UP / ON UP <uUpAction> ] ;
[ DOWN / ON DOWN <uDownAction> ] ;
[ PAGEUP / ON PAGEUP <uPgUpAction> ] ;
[ PAGEDOWN / ON PAGEDOV <uPgDownAction> ] ;
[ ON THUMBPOS <uPos> ] ;
[ COLOR / COLORS <nClrFore>;
,<nClrBack> ]];
[ MESSAGE <cMsg> ] ;
[ WHEN <WhenFunc> ] ;
[ VALID <ValidFunc> ] ;
[ OF / WINDOW / DIALOG <oDlg>]

```

<nRow>,
 <nCol> Coordenadas del ScrollBar, imita coordenadas Texto.
 <oSbr> Referencia al objeto ScrollBar.

<oDlg>	Ventana contenedora del ScrollBar.
<nMin>, <nMax>	Rango de valores representados por la ScrollBar.
<nPgStep>	Valor de incremento cuando se selecciona PageUp o PageDown en la ScrollBar.
<nWidth> <nHeight>	Dimensiones en Pixels del ScrollBar
<UpAction>	Acción a ejecutar cuando se presiona subir en la ScrollBar, esto es cuando se hace click sobre la flecha arriba o cuando se presiona la tecla K_UP.
<DnAction>	Acción a ejecutar cuando se presiona bajar en la ScrollBar, esto es cuando se hace click sobre la flecha abajo o cuando se presiona la tecla K_DN.
<PgUpAct>	Acción a ejecutar cuando se presiona pagina arriba en la ScrollBar, esto es cuando se hace click entre la flecha de arriba y el tumb.
<PgDnAct>	Acción a ejecutar cuando se presiona pagina abajo en la ScrollBar, esto es cuando se hace click entre la flecha de abajo y el thumb.
<uPos>	Acción a ejecutar cuando el usuario cambia la posición del Thumb. La nueva posición es provista por el valor npos.
<nId>	Identificador numérico del recurso del Control.
<nClrFore>	Color activo como numero RGB o String xBase.
<nClrBack>	Color de fondo como numero RGB.
<ValidFunc> <cMsg>	Expresión a validar cuando el control pierda el focus. Mensaje asociado al control.
<WhenFun>	Expresión a evaluar para indicar si el control esta o no habilitado.

Cláusulas:

HORIZONTAL

VERTICAL Permite usar barras Horizontales o Verticales, Solo desde código Fuente.

PIXEL Usa coordenadas Pixels en vez de coordenadas Texto.

DESIGN Permite que el usuario mueva y reajuste el control con el mouse.

UPDATE Este control puede ser actualizado desde el Método TDialog:Update().

Trees (Arboles):

Creando un Árbol:

```
TREE [<oTree>;  
      [BITMAPS <cBmpOpen>, <cBmpClose>]  
  
      TREEITEM [<oltem> PROMPT I <cLabel>;  
               [RESOURCE <cResName 1> [, <cResName2>] ];  
               [FILENAME <cBmpOpen> [, <cBmpClose> ] ];  
               [OPENED / OPEN]
```

...

```

[TREE <oSubTree>]
    [TREEITEM ... ]
    [...]
[ENDTREE]

```

...
ENDTREE

<oTree> A reference to the TTree object just created.

<cBmpOp> Optional bitmaps files or resources names for
<cBmPCl> the tree nodes open and close states images.

<oltem> A reference to the TTreeltem object just created.

<cLabel> The prompt of an Treeltem element.

<cResl> Optional bitmaps resources names for the ítem

<cBmpCl>
<cRes2> Abren y cierran imágenes de estado.
<cBmpOpen>
<cBmpClosed> Se usan por defecto si
<cResName 1 >
<cResName2> No se especifican.

<cBmpOp> Nombre de archivo Bitmap Opcional para el Ítem.

<oSubTree> Referencia al Objeto TTree creado.

Video:

Definiendo un Control VIDEO desde código fuente:

```

@<nRow>, <ncol> VIDEO <oVideo>;
    [ SIZE <nWidth>, <nheight> ] ;
    [ FILE / FILENAME / DISK <cAviFile> ] ;
    [ OF / WINDOW / DIALOG <oWnd>]

```

Redefiniendo un Control VIDEO desde recursos:

```

DEFINE VIDEO <oVideo>;
    [ID <nId> ] ;
    [FILE I FILENAME I DISK <cAviFile> ] ;
    [OF I WINDOW I DIALOG <ownd> ]
    [WHEN <u\Vhen> ] ;
    [VALID <uvalid> ] ;

```

Activando el VIDEO:

ACTIVATE VIDEO <o Video>

<nRow>,
<ncol> Coordenadas del control, imita coordenadas Texto.

<ovideo> Referencia al Objeto VIDEO.

<nWidth>, <nheight>	Dimensiones del Control.
<cAviFile>	Nombre del archivo AVI a ejecutar.
<ownd>	Referencia a la ventana Contenedora del Control.
<WhenFun>	Expresión a evaluar para indicar si el control esta o no habilitado.
<ValidFunc>	Expresión para validar cuando el control pierda el Focus.

VBXs, Visual Basic controls:

Creando un VBX desde código fuente:

```
@<nRow>, <ncol> VBX [<oVbx>];
  [OF <ownd> ] ;
  [SIZE <nWidth>, <nheight>] ;
  [FILE I FILENAME I DISK [<cVbxFile> <cVbxClass> ] ;
  [ON <cclause I > <uaction I >;
  [ON <cClauseN> <uActionN>] ] ;
  [WHEN <uWhen>] ;
  [VALID <uValid> ] ;
  [PIXEL ] ;
  [DESIGN ]
```

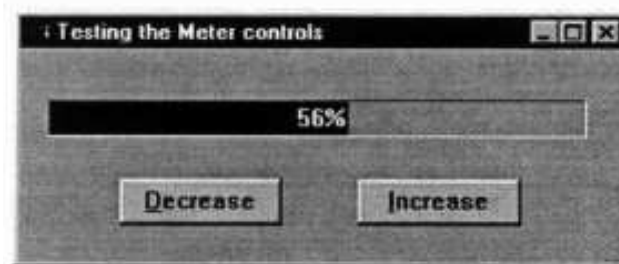
Advertencia: Algunos controles VBXs necesitan datos específicos de la inicialización y no trabajan apropiadamente si los usa en ventanas. En esos casos úselos sólo en DialogBox usando REDEFINE VBX

creando VBX desde recursos:

```
REDEFINE VBX [<oControl >];
  [ID <nLd> ] ;
  [OF <oDlg> ] ;
  [COLOR <nClrFore> [,<nClrBack>] ] ;
  [ON <cclauseI> <uactionI>;
  [ON <cClauseN> <uActionN>] ] ;
```

<nRow>, <nCol>	Coordenadas del control, imita coordenadas texto.
<oVbx>	Referencia a Objeto VBX.
<nWidth>, <nHeight>	Dimensiones del Control VBX.
<cVbxFile>	Nombre del Archivo VBX.
<cVbxCl>	Nombre opcional del VBX especificado por el fabricante.
<cClause>	Nombre del evento, especificado por el fabricante.
<uAction>	Acción a ejecutar para este evento.
<nld>	Identificador del control VBX dentro del DialogBox.

- <odlg> Referencia al DialogBox contenedor del Objeto.
- <nClrFore>, <nClrBack> Colores del VBX desde DialogBox.
- <WhenFun> Expresión a evaluar para indicar si el Control esta o no Habilitado.
- <ValidFun> Expresión para validar cuando el control pierda el focus.
- Cláusulas:
- DESIGN Permite al usuario mover y reajustar el control con el mouse.
- PIXEL Usa coordenadas Pixels n vez de coordenadas Texto.

Meters:

Creando un Meter desde Código Fuente:

```
@ <nRow>, <nCol> METER [ <oMeter> VAR <nActual>;
  [TOTAL <ntotal> ];
  [SIZE <nWidth>, <nheight>];
  [OF <ownd>];
  [UPDATE];
  [PIXEL] ;
  [FONT <ofont>];
  [PROMPT <cprompt>];
  [NOPERCENTAGE ];
  [COLOR | COLORS <nClrPane>, <nClrText>];
  [BARCOLOR <nClrBar>, <nClrBText>];
  [DESIGN ];
```

Creando un Meter desde recursos:

```
REDEFINE METER [ <oMeter> VAR ] <nActual>;
  [TOTAL <nTotal> ] ;
  [ID <nId> ] ;
  [OF <oWnd> ] ;
  [UPDATE ] ;
  [FONT <oFont>] ;
  [PROMPT <cPrompt>] ;
  [NOPERCENTAGE ];
  [COLOR / COLORS <nClrPane>, <nClrText>] ;
  [BARCOLOR <nClrBar>, <nClrBText>] ;
```

- <nRow>, <nCol> Coordenadas del control, imita coordenadas texto.
- <oMeter> Referencia al Objeto METER.


```

[ID <nId> ];
[OF / WINDOW / DIALOG <oWnd>];
[NAME / RESNAME / RESOURCE <cResName>];
[FILE / FILENAME / DISK <cBmpFile> ];
[ON CLICK / ON LEFT CLICK <uLClick> ];
[ON RIGHT CLICK <uRClick>];
[SCROLL];
[ADJUST ];
[CURSOR <oCursor>];
[MESSAGE <cMsg>];
[UPDATE];
[WHEN <WhenFunc>];
[VALID <uValid>];

```

<nRow>, <nCol> Coordenadas en pantalla, imita coordenadas texto.

<oBmp> Referencia al Objeto Bitmap.

<cRes> Nombre del recurso Bitmap cuando se carga de archivos .RC, .RES o .DLL.

<cBmpFile> Nombre del archivo BMP cuando se carga directamente desde Disco.

<nWidth> Ancho del Bitmap.

<nheight> Alto del Bitmap.

<oWnd> Ventana contenedora del control BITMAP.

<uLClick> Acción a ejecutar cuando haga Click con el botón izquierdo sobre el BitMap.

<uRClick> Acción a ejecutar cuando haga Click con el botón derecho sobre el BitMap.

<oCursor> Referencia al Objeto Cursor a usar sobre este control.

<cMsg> Mensaje asociado con el control.

<nId> Identificador numérico del recurso del control.

<WhenFun> Expresión que indica cuando este control esta activo o no.

Cláusulas:

NO BORDER No pone un borde alrededor del Control.

UPDATE este control puede ser actualizado desde el método TDialog:Update().

PIXEL Usa coordenadas Pixels en vez de coordenadas Texto.

SCROLL Habilita el uso de Barras de Scroll para mover el Bitmap dentro del área de visibilidad. Se usa cuando el Bitmap es más grande que el Control.

ADJUST Ajusta automáticamente el Bitmap al tamaño del Control.

DESIGN Permite que el usuario mueva y reajuste el control con el mouse.

Cursors:

Creando un cursos desde recursos:
DEFINE CURSOR <oCursor>;

```
[RESOURCE / RESNAME / NAME <cResN> ];  
[ARROW / ICON / SIZENS / SIZEWE / SIZENWSE,;  
SIZENESW / IDEAM / CROSS ]
```

<oCursor> Referencia al Objeto cursor.

<cResN> String con el nombre del recurso CURSOR. es similar al identificador numérico.

Brushes:

Definiendo un Brush desde Código Fuente:

```
DEFINE BRUSH <oBrush>;  
[STYLE<Style>];  
[COLOR <nRGBClr> ];  
[FILE / FILENAME / DISK <cBmpFile>];  
[RESOURCE / NAME / RESNAME <cBmpRes>]
```

Asignando el Brush a la Ventana:

```
SET BRUSH [ OF <ownd> ] [ TO <obrush>
```

<obrush> Referencia al Objeto Brush.

<Style> Uno de los estilos definidos más adelante.

<nRGBClr> Color en formato RGB. Use la función NRGB().

<cBmpFile> Nombre del Archivo BMP para construir el Brush.

<cBmpRes> Nombre de un Recurso BMP.

<oWnd> Referencia a la ventana contenedora del Brush.

Estilos Diferentes:

Nuevos Estilos BORLAND, TILED, BRICKS

Estilos Normales DIAGONAL,HORIZONTAL,VERTICAL,FDIAGONAL,BDIAGONAL,CROSS,DIAGCROSS

Icons:

Creando un control ICON desde Código Fuente:

```
@<nRow>, <ncol> IC ON [ <olcon> ] ;  
[NAME / RESOURCE / RESNAME <cResName> ] ;  
[FILE / FILENAME / DISK <icofile> ] ;  
[BORDER ] ;  
[COLOR / COLORS <nClrFore>; [,<nClrBack> ] ;  
[ON CLICK <uClick>  
[OF / WINDOW / DIALOG <oWnd> [UPDATE];  
[WHEN <WhenFunc> ] ;
```

Redefiniendo un ICON desde Recursos:

```
REDEFINE ICON <olcon> [ ID <nId> ];  
[NAME / RESOURCE / RESNAME <cResN> ];  
[FILE / FILENAME / DISK <icofile> ] ;  
[ON CLICK <uclick> ] ;  
[OF / WINDOW / DIALOG <ownd> [UPDATE];
```

[WHEN <VwhenFunc>] ;

<nRow>,
<nCol> Coordenadas del control en pantalla, imita coordenadas Texto.

<olcon> Referencia al Objeto ICON.

<cResN> Nombre del recurso ICON desde archivos RC, RES o DLL.

<clcoFile> Nombre del ARCHIVO .ICO cuando es cargado desde Disco.

<uClick> Acción a ejecutar cuando se hace Click con el mouse sobre el Icon.

<oWnd> Objeto Contenedor del ICON.

<WhenFunc> Expresión a evaluar para indicar si el control esta activo o no.

<nClrFore> Color activo como un numero RGB o como un string xBase.

<nClrBack> Color de Fondo como un numero RGB.

Definiendo un ICON sin mostrarlo:

DEFINE ICON <olcon>;
[NAME / RESOURCE / RESNAME <cResN>];
[FILE / FILENAME / DISK <clcoFile>];
[WHEN <WhenFunc>];

<olcon> Referencia al Objeto ICON.

<cResN> Nombre de recurso ICON desde archivo RC, RES o DLL.

<clcoFile> Nombre del Archivo ICO si se carga directo desde Disco.

Cláusulas:

UPDATE Permite actualizar este control por el método <oDlg>:Update() del contenedor.

BORDER crea un Borde alrededor del control.

Fonts:

Definiendo un FONT:

DEFINE FONT <oFont>;
[NAME <cname>];
[SIZE <nWidth>, <nheight>];
[FROM USER];
[BOLD];
[ITALIC];
[UNDERLINE];
[WEIGHT <nWeight>
[OF <oDevice>];
[NESCAPEMENT <nEscape>];

Activando un FONT:

ACTIVATE FONT <oFont>

DesActivando un FONT:

DEACTIVATE FONT <oFont>

Soltar un FONT:

RELEASE FONT <oFont>

Fijando un FONT a una ventana:

SET FONT [OF <ownd> TO <ofont>]

| | |
|------------------------|------------------------------------------|
| <oFont> | Referencia al Objeto FONT. |
| <cName> | Nombre del Font Ej: Arial, Roman, etc... |
| <nWidth>,
<nHeight> | Dimensiones del FONT |
| <oWnd> | Referencia a la Ventana contenedora. |
| <oDevice> | Dispositivo dueño del font. |
| <nWeight> | Ancho del Font. |
| <nEscape> | El escape del FONT. |

Cláusulas:

| | |
|-----------|-------------------------------------------------|
| FROM USER | Despliega el DialogBox para selección del FONT. |
| BOLD | Crea un Font GRUESO. |
| ITALIC | Crea un Font CURSIVO. |
| UNDERLINE | Crea un Font SUBRAYADO. |

Pens (Lápices):

Definiendo un Control PEN:

```
DEFINE PEN <oPen>;
  [ STYLE <nStyle>];
  [ WIDTH <n Width>];
  [ COLOR <nRGBClr>];
  [ OF / WINDOW / DIALOG <ownd>];
```

Activando un Control PEN:

ACTIVATE PEN <open>

| | |
|-----------|-----------------------------------------------------|
| <oPen> | Referencia al Objeto PEN. |
| <nStyle> | Estilo opcional del PEN. |
| <nRGBClr> | Color del PEN como un numero RGB o un String xBase. |

Menus:

Creando un Menu PullDown desde Código Fuente:

```
MENU [ <oMenu> ] [POPUP]
    MENUITEM [ <omenuitem> PROMPT I <cPrompt>;
        [MESSAGE <cmsg> ];
        [CHECK / CHECKED / MARK ];
        [ENABLED / DISABLED ];
        [FILE I FILENAME I DISK <cBmpFile>];
        [RESOURCE / RESNAME / NAME <ResName> ];
        [ACTION <uAction>];
        [BLOCK <bAction> ];
        [OF / MENU / SYSMENU <oMenu>];
        [ACCELERATOR <nState>, <nVirtKey>];
        [HELP ];
        [HELP ID / HELPID <nHelpId>];
        [WHEN <uWhen> ]

    SEPARATOR [<oMenuitem>]

MENUITEM...

    MRU <oMrU>;
        [INI / ININame / FILENAME / NAME / DISK <clniFile> ];
        [SECTION <csection>];
        [SIZE | ITEMS <nItems>];
        [MESSAGE <cmsg>];
        [ACTION <uAction>]

ENDMENU
```

Creando un Menu PullDown desde recursos:

```
DEFINE MENU <oMenu>;
    [RESOURCE / NAME / RESNAME <cResName> ];
    [POPUP];
REDEFINE MENUITEM [ <omenuitem>;
    PROMPT <cprompt> ];
    [ID <nId> OF / MENU <oMenu> ];
    [ACTION <uAction>];
    [BLOCK<bAction> ];
    [MESSAGE <cMsg>];
    [CHECK / CHECKED / MARK ];
    [ENABLED / DISABLED ];
    [FILE / FILENAME / DISK <cBmpFile> ];
    [RESOURCE / RESNAME / NAME <cResName>];
    [ACCELERATOR <nState>, <nVirtKey> ];
    [WHEN <uwhen> ]
```

Activando un Menu PopUp:

```
ACTIVATE POPUP <oMenu>;
    AT <nRow>, <nCol> ];
    OF I WINDOW I DIALOG <ownd>]
```

Redefiniendo el System Menu de una Ventana:

```
REDEFINE SYSMENU <oMenu>;
```

[OF / WINDOW / DIALOG <ownd>]**- Menuitems son similares a los anteriores.-****ENDSYSMENU**

| | |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <oMenu> | Referencia al Objeto MENU. |
| <oMenuItem> | Referencia a uno de los Menuitems del Menú. Cada MenuItem es un Objeto. La Clausula SEPARATOR puede ser un opcional MenuItem . |
| <cMsg> | Mensaje asociado al control. |
| <cBmpFile> | Nombre de un archivo BMP usado como etiqueta para un menuitem del menú. |
| <cResName> | Nombre del recurso del menú. |
| <uAction> | Acción a ejecutar por un MenuItem específico. |
| <bAction> | La misma acción expresada como un bloque de código. |
| <nRow>,
<nCol> | Coordenadas donde pondremos el Menú PopUp |
| <oWnd> | Ventana contenedora del menú. |
| <nState> | Estado de la tecla aceleradora (ver VKEY.CH) |
| <nVirtKey> | Valor ASCII para la tecla aceleradora. |
| <nHelpId> | Identificador del típico de ayuda para un menuitem específico. |
| <uWhen> | Expresión a evaluar que indica si el menuitem esta o no activo. |
| <nId> | Identificador numérico del recurso del menuitem. |
| <cPrompt> | Etiqueta del menuitem. |
| <oMrU> | Referencia a un nuevo objeto MRU creado. |
| <clnifile> | Nombre del archivo INI a usar con el objeto MRU. |
| <cSection> | Sección dentro del archivo INI donde se almacenan los ítems del MRU. |
| <nItems> | Numero de items a manejar por el objeto MRU. |
| <uAction> | Acción a ejecutar cuando seleccione un ítem MRU. |
| Cláusulas: | |
| CHECK
CHECKED
MARK | Si el menuitem aparece inicialmente chequeado |
| ENABLED | El menuitem aparece inicialmente habilitado (Por Defecto). |
| DISABLED | El menuitem aparece inicialmente DesHabilitado |
| HELP | Para indicar ayuda en el menuitem del menú pulldown. |
| POPUP | Para crear un menú popup desde el menú pulldown. |

Comandos Misceláneos

Clipboard:

Creando un Objeto ClipBoard:

```
DEFINE CLIPBOARD <oClp>;  
    FORMAT TEXT / OEMTEXT / BITMAP / DIF];  
    OF <oWnd> ]
```

```
ACTIVATE CLIPBOARD <oClp>
```

<oClp> Referencia al Objeto ClipBoard.
 <oWnd> Ventana con la que se maneja el ClipBoard.
 <cText> Texto a poner en el ClipBoard.

Cláusulas:

TEXT Texto Normal.
 OEMTEXT
 BITMAP No implementado aun.
 DIF No implementado aun.

DDE (Dynamic Data Exchange):

definiendo una nueva conexión:

```
DDEDEFINE [ DDE / LINK <oDde>;  
    /SERVICE <cService>];  
    /TOPIC <cTopic> ];  
    /ITEM <cltem>];  
    [ACTION <uAction>];  
    [VALID<uEnd>];
```

Activando el DDE:

```
ACTIVATE [DDE / LINK ]<oDde>
```

Desactivando el DDE:

```
DEACTIVATE [ DDE / LINK ] <oDde>
```

Finalizando la conexión:

```
RELEASE [ DDE / LINK ] <oDde>
```

<oDde> Referencia al Objeto DDE.
 <cService> Servicio con el cual establecer la conexión.
 <cTopic> Tópico con el que establecemos la conexión.
 <cltem> Ítem con el que establecemos la conexión.

<uAction> Acción a ejecutar.
 <uEnd> Acción a ejecutar cuando el DDE termina.

Configurando y manipulando archivos HELP:

Los comandos del Sistema de Ayuda se explica en la “Guía de Programación” en la sección “Documentando”.

Manejando archivos INI:

Manejando un archivo INI:

INI <olni> [FILE / FILENAME / DISK <clniFile>]

```
GET <uVar>;
  [SECTION <cSection>];
  [ENTRY <cEntry> ] ;
  [DEFAULT <uDefault> ];
  [OF / INI <olni>]
```

```
SET;
  [SECTION <cSection>];
  [ENTRY <cEntry>];
  [TO<uVal> ];
  [OF / INI <olni>]
```

ENDINI

<olni> Referencia al Objeto INI.

<clniFile> Nombre del archivo INI.

<uVar> Variable con el valor cargado desde archivo INI. El valor del get se convierte automáticamente al tipo de la variable. Por lo mismo es importante inicializar adecuadamente la variable antes de cargar un valor.

<cSection> [SECTION] -Formato típico dentro del INI -

<cEntry> Entrada=Valor

<uDefault> Valor por defecto a usar cuando la entrada no se encuentra.

<uVal> Nuevo valor a asignar.

MRU, Most Recently used items (Items Usados Recientemente):

Ver explicación en la sección “Menús” de este manual.

ODBC, Manejando Bases de Datos mediante ODBC (Open DataBase Connectiviti):

Los comandos ODBC son explicados en “Guía de Programación” en la Sección “Manejando ODBC”.

Usando la Impresora:

Usando la Impresora:

```
PRINT [ <oPrint>;
  [TITLE /NAME /DOC <cDoc> ];
  [FROM USER];
  [PREVIEW ]
  [TO <xModel>]

  PAGE

    <oPrint>:Say( <nRow>, <nCol>, <cText>;
      [<oFont>], [<nWidth>];
      [<nClrText>]

    ...
  ENDPAGE

ENDPRINT
```

<oPrint> Referencia al Objeto Impresora.

<cDoc> Nombre del documento a generar, este nombre aparece en el Spooler.

<nRow>,
<nCol> Las coordenadas de inicio de la impresión medidas en unidades de la resolución de la impresora. Esto significa que en impresoras diferentes el resultado será diferente.

Para evitar esto (quiere el mismo resultado en impresoras diferentes) debe usar el método oPrn:CmSay() o oPrn:InchSay() para usar coordenadas en centímetros o pulgadas.

<cText> Valor a imprimir. Puede ser de cualquier tipo. FiveWin automáticamente los convertirá a expresiones String.

<oFont> Objeto FONT usado cuando imprima el valor <cText>. Vea los comandos xBase de FONT para manejarlos apropiadamente. Este es un parámetro opcional.

Importante: Si especifica una dimensión para el font independiente de la impresora usada, deberá especificar la Clausula OF... cuando cree el FONT, y debe hacerlo después que el objeto print se ha creado:

local oPrn, oFont

```
PRINT oPrn
  DEFINE FONT oFont NAME "Arial" SIZE 0, -8 OF oPrn
  PAGE

  ...
  ENDPAGE
ENDPRINT
```

oFont:End()

Importante: Debe :End() el objeto font solo cuando ha terminado y completado su trabajo de impresión.

<nWidth> Ancho en unidades específicas de impresora usado cuando imprima el valor <cText>. Opcional.

<nClrText> Color opcional usado cuando imprima el valor <cText>.

Cláusulas:

| | |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PAGE | Parte con una nueva pagina. |
| ENDPAGE | Termina la pagina de impresión. |
| ENDPRINT | Termina la impresión. |
| FROM USER | Permite al usuario seleccionar la impresora a usar, si no se especifica esta clausula FiveWin imprime en la impresora instalada por defecto en su sistema. |
| PREVIEW | Llama al sistema de Vista Previa de FiveWin, desde aquí el usuario decide si envía el trabajo a impresión o no. |

Generador de Reportes FiveWin:

Los comandos del generador de Reportes son explicados en “Guía de Programación” en la Sección “Impresión”.

Timers (Cronómetros):

Definiendo un nuevo Timer:

```
DEFINE TIMER <oTimer>;  
    [INTERVAL <nInterval> ] ;  
    [ACTION <uAction> ] ;  
    [OF / WINDOW / DIALOG <ownd> ]
```

Activando un Timer:

```
ACTIVATE TIMER <oTimer>
```

Desactivando un timer:

```
DEACTIVATE TIMER <oTimer>
```

Finalizando un Timer:

```
RELEASE TIMER <oTimer>
```

| | |
|-------------|--------------------------------------------------------------|
| <oTimer> | Referencia al Objeto Timer. |
| <nInterval> | Define el intervalo del timer en milésimas de Segundos. |
| <uAction> | Acción a ejecutar cuando el periodo del intervalo se cumpla. |
| <oWnd> | Referencia a la ventana contenedora. |

xBase elementos comunes

Hemos puesto particular atención a la sintaxis de FiveWin, vemos como se usa la misma cláusula para hacer la misma cosa en circunstancias similares. Esto significa que si dominamos estas pocas reglas, controlaremos mucho mejor diferentes rasgos de FiveWin.

Nos extendemos en este concepto básico para manejar todas las partes diferentes del API Windows. Teniendo una única interface xbase que permitirá manejar el API Windows, logrando que se concentre en la meta de su aplicación y no en los detalles.

Define

Todos los Objetos en FiveWin se crean usando la palabra DEFINE:

DEFINE <Class> <Object> [<more clauses>]

Como ejemplo, para crear una window haremos:

DEFINE WINDOW oWnd ...

| | |
|--------------------------|-----------------------|
| Para crear un DialogBox: | DEFINE DIALOG oDlg... |
| Para crear un ICON: | DEFINE ICON olco ... |
| Para crear un Report: | DEFINE REPORT oRpt... |

¡Acuérdese de usar DEFINE para crear cualquier cosa!

Activate

Todo en FiveWin comienza a trabajar al usar la palabra ACTIVATE

ACTIVATE <Class> <Object> [<more clauses>]

Como ejemplo, para activar una ventana haremos:

ACTIVATE WINDOW oWnd ...

| | |
|----------------------------|--------------------------|
| Para activar un DialogBox: | ACTIVATE DIALOG oDlg ... |
| Para activar un Report: | ACTIVATE REPORT oRpt ... |
| Para activar un Timer: | ACTIVATE TIMER oTmr ... |

¡Acuérdese de usar ACTIVATE para hacer que las cosas trabajen!

:End()

: End() Aunque éste es un mensaje OOP, lo usamos en FiveWin para terminar la ejecución de algo:

<Object>:End()

Como ejemplo, para terminar la ejecución de una ventana hacemos:

oWnd:End()

| | |
|-----------------------------|------------|
| Para terminar un DialogBox: | oDlg:End() |
| Para terminar un Report: | oRpt:End() |
| Para terminar un Timer: | oTmr:End() |

Usamos mensajes OOP en lugar de un comando xbase, porque lo ocupamos en diferentes lugares, donde no es posible ocupar un comando xbase:

REDEFINE BUTTON ID ID_END OF oDlg ACTION oDlg:End()

Of:

Siempre que necesitamos especificar al dueño o contenedor de un Objeto, usamos OF:

REDEFINE BUTTON ID ID_OK OF oDlg

La cláusula OF indica: "pertenece a," "se contiene en," "su dueño es".

Por ejemplo para especificar el dueño de una DialogBox:

DEFINE DIALOG oDlg RESOURCE "Test" OF oWnd

Para una ventana hija: **DEFINE WINDOW oWnd2 TITLE "Test" OF oWnd**

Para un Control: **REDEFINE GET cName ID ID_NAME OF oDlg**

Para un Font: **DEFINE FONT oFont NAME "Arial" SIZE 0, -20 OF oPm**

Resource/Resname/Disk:

Usamos esta Clausula para especificar que un objeto se construye desde un recurso.

DEFINE ... RESOURCE <cResourceName>

Por ejemplo para crear un ICONO desde recursos, usamos:

DEFINE ICON olco RESOURCE "MyIcon"

Para crear una DialogBox: **DEFINE DIALOG oDlg RESOURCE "Clients"**

Para crear un Cursor: **DEFINE CURSOR oCrs RESOURCE "Eye"**

Recuerde: Use siempre la clausula RESOURCE para crear un Objeto desde un recurso almacenado dentro del mismo EXE (Recursos RC) o en un DLL.

Filename/File/Disk:

Usamos esta clausula para especificar que un Objeto es cargado desde un archivo en disco.

DEFINE ... FILENAME <cFileName>

Por ejemplo para usar un botón bitmap con un archivo BMP dentro, usamos:

DEFINE BUTTON OF oBar FILENAME "Exit.bmp"

Para crear un Control BitMap desde disco, usamos:

**REDEFINE BITMAP obmp ID ID_PHOTO OF oDlg;
FILENAME "Photo.bmp"**

Recuerde: Siempre la clausula FILENAME para usar archivos almacenados en disco.

Size:

Siempre que tenemos que especificar dimensiones pixel <nHeight> usamos la cláusula SIZE.

Recuerde: Siempre usar SIZE para especificar dimensiones.

Clases y Objetos

En este capítulo verá una pequeña descripción de las clases disponibles en FiveWin, ordenadas por categoría. Para obtener más información de clases específicas consulte el archivo Help de FiveWin o las guías Norton.

Clases Windows

| | |
|-------------------|----------------------------------------------------------|
| TWindow | Clase base de ventanas. |
| TMdiChild | Interface de Múltiple despliegue Hija |
| TMdiClient | Interface de Múltiple despliegue Cliente. |
| TMdiFrame | Interface de Múltiple despliegue Área de Trabajo (Frame) |

Clases Dialog

| | |
|----------------|-----------------------------------------------------|
| TDialog | Clase Dialog Box |
| TFolder | Objeto Folder |
| TTabs | Manejador de Tabs |
| TPages | DialogBox múltiple manejo de paginas |
| TPanel | Auto-alineación característica de soporte de panels |

Clases WinControl

| | |
|-------------------|------------------------------------------------|
| TBtnBmp | Objetos Botones BitMaps |
| TBar | Objetos Barra |
| TBitmap | Objetos Bitmap |
| TButton | Controles de Botones de comandos |
| TcBrowse | Browser con objetos Columnas |
| TcColumn | Objetos Columnas para la Clase TcBrowse |
| TCheckBox | Controles Lógicos CheckBox (Cajas de Chequeo) |
| TComponent | Manipulación visual de objetos no-visuales. |
| TComboBox | ComboBox de Windows como objetos. |
| TControl | Base para todos los controles GUI. |
| TGet | Clase de los objetos GET |
| TGroup | Clase de los controles Group Dialog |
| TListBox | Clase de los controles ListBox |
| TMci | Clase Media Control Interface |
| TMetaFile | MetaFiles Windows como objetos |
| Tmeter | Manejo de los controles Meter. |
| TMGet | GETs de Múltiples líneas |
| TMsgBar | Manejo de Objetos MessageBar |
| TMsgItem | Manejo de los items de MessageBar como objetos |
| TRadio | Un control Radio (Radio Buttons) |
| TRadMenu | Manejo de los controles RadioButton |
| TTreeltem | Soporte de los items de listas de enlaces |
| TVbControl | Clase de Controles VBx (Visual Basic controls) |
| TVideo | Soporte de archivos AVI como objetos |
| TWBrowse | Control de Browser como Dialog Boxes |
| TScrollBar | Objetos control de ScrollBar |
| TSay | Clases de Objetos SAY |

Clases para Recursos

| | |
|----------------|------------------------------------------|
| Tbrush | Objetos Brushes (Brochas) |
| TCursor | Objetos para manejo del cursor del Mouse |
| TFont | Clase de Objetos Fonts |
| Tico | Clases de Archivos ICO (Iconos) |

| | |
|------------------|--------------------------------------------------|
| TMenu | Manejo de menú Windows. |
| TMenuItem | Ítems de menú como Objetos |
| TMru | MRU (Most Recently used Menu lists) como objetos |
| Tpen | Manejo de Pens de Windows (Lápices) |

Clases para Bases de Datos

| | |
|------------------|------------------------------------------------|
| TField | Campos manejados como objetos |
| TDataBase | Manejo de Bases de Datos Orientadas a Objetos |
| TDbOdbc | Manejo de Bases de Datos vía ODBC como Objetos |
| TDbm | Manejador de Bases de Datos |
| TOdbc | Manejo de secciones ODBC como objetos |
| TIndex | Índices como Objetos |

Clases para Comunicaciones

| | |
|----------------|------------------------------------------------|
| TSocket | Implementación de sockets TCP/IP para InterNet |
| TComm | Comunicación como Objeto |

Clases para Reportes e impresión

| | |
|-----------------|--------------------------------------------------|
| TPrinter | Manejo de la impresora como un objeto |
| TRColumn | Columnas basadas en objetos columnas del browser |
| TReport | Generador de reportes |
| TrColumn | Columnas del Reporte |
| TrGroup | Grupos del Reporte |
| TrLine | Líneas del Reporte |

Clases Misceláneas

| | |
|-------------------|--------------------------------------------------------|
| Tblock | Clase Bloque de Código almacenable en disco de FiveWin |
| TClipboard | Manejo del Clipboard de Windows como Objetos |
| TDDE | Dynamic Data Exchange como objetos |
| TDDEServer | Dynamic Data Exchange nuevo DDML Server |
| TDDEClient | Dynamic Data Exchange nuevo DDML Client |
| TDict | Clase Diccionario de Bases de Datos |
| Tfile | Manejo de archivos a bajo nivel como objetos |
| Tini | Calse de Archivos INI |
| TLex | Analizador léxico de FiveWin |
| TLibFile | Manejo de archivos LIB |
| TLinkList | Listas de enlace (Base de la Clase Trees) |
| TMail | Correo electrónico Electronic mail as Objects |
| TObjFile | Archivos OBJ manejados orientados a objetos |
| TParser | parser dinámico de FiveWin |
| TRect | Manejo de Rects de Windows |
| TTime | Manejo de Objetos tiempo |
| TTimer | Manejo de cronómetros como objetos |
| TTxtEdit | Edición visual de archivos ASCII |
| TTxtFile | manejo de archivos ASCII como objetos |
| TYacc | Maquina Yacc de FiveWin |

Objects engine syntax

CLASS <TMyClass> [FROM <TParentClass>]

[EXPORT /PUBLIC]

```

[PROTECTED / READONLY]
[LOCAL / HIDDEN]

DATA <DataName> [ AS CHARACTER / NUMERIC / DATE/;
LOGICAL / CODEBLOCK / NIL ]
...
METHOD <MethodName>( [<params,...>] );
[ CONSTRUCTOR SETGET I VIRTUAL I BLOCK

METHOD <MethodName>( [<params .... >] );
INLINE <uCode,...>

MESSAGE <cMsgName> METHOD <MethodName>

```

ENDCLASS

- <TMyClass> Nombre de la nueva clase que crearemos. no puede exceder a diez caracteres.
- <TPartClass> Nombre de la cual heredara esta nueva clase.
- <DataName> Nombre del elemento DATE (variables de instancia) usada para Objetos en esta nueva clase.
- <MetName> Nombre de Método (Acción) del Objeto.
- <params,...> Lista opcional de elementos pasados por parámetros al método.

Cláusulas:

CONSTRUCTOR Cuando vamos a inicializar un Objeto y queremos que automáticamente retorne una referencia a si mismo. Es una buena práctica primero especificar los métodos que construyen un objeto (constructores).

SETGET Son Datos calculados. Significa que manejaremos este nombre de método como un dato.

VIRTUAL Métodos que no hacen nada. Útiles para herencias o cuando comienza a crear y probar una clase.

BLOCK Sumamente rápido y de fácil código, use esta cláusula cuando quiera declarar métodos 'inline' sólo desde la definición de la Clase. Usamos mucho ellos en las clases FiveWin. Tiene que especificar como primeros parámetro 'Self'.

Todo métodos automáticamente requieren un primer parámetro que es 'Self'. Objects.ch automáticamente proporciona 'Self' en Métodos normales, pero en Métodos BLOCK se debe especificar. Así, BLOCK requiere se especifique "Self" y los parámetros del bloque de código:

```

METHOD <MethodName> BLOCK;
 / Self, <arg 1>, <arg2>, ..., <argN> /...

```

La ventaja de métodos BLOCK sobre los Métodos normales es que no requieren escribir código relacionado, cuando se incluye en la misma declaración. Así reduce el número de símbolos CA-Clipper requeridos, pero tiene la desventaja que siempre se debe evaluar un bloque de código. En un MÉTODO normal la evaluación de un bloque de código no es necesaria, evitando una sobre ejecución.

INLINE La misma funcionalidad que la cláusula BLOCK, pero es más fácil de usar porque no es necesario construir un bloque de código -se construye automáticamente- es lo mejor del xBase. Planeamos en próximas versiones de FiveWin convertir todo nuestros métodos BLOCK en cláusulas INLINE.

Los métodos INLINE se convierten automáticamente en definiciones de métodos BLOCK. Es sólo una manera más fácil y rápida de codificar un método BLOCK. Recomendamos que use métodos INLINE en lugar de métodos BLOCK.

Como los métodos INLINE se preprocesan automáticamente en métodos BLOCK, se proporciona 'self' automáticamente así no necesita especificarlo.

Codificación de Métodos:

```
METHOD <MethodName>( [<params ,... >] ) CLASS <TMyClass>
  - tenemos un "Self" local listo para usar aquí.
  - use ::< Datos> para acceder a Self:<Data>.
return
```

Incluya solo **una** clase por PRG.

Funciones

En este capítulo verá una pequeña descripción de las funciones disponibles en FiveWin, ordenadas por categoría. Para obtener más información de funciones específicas consulte el archivo Help de FiveWin o las guías Norton.

Clipboard

| | |
|-------------------------|------------------------------------------------|
| OpenClipboard() | Abre el ClipBoard de Windows para uso. |
| CloseClipboard() | Cierra el ClipBoard. |
| SetClpData() | Pone datos en el ClipBoard. |
| GetClpData() | Toma datos del manejador activo del ClipBoard. |
| EmptyClipboard() | Limpia el ClipBoard. |

Color

| | |
|-----------------------|------------------------------------------------------|
| ChooseColor() | Selección de Color desde la DialogBox normal. |
| nColorToN() | Convierte un string de Color DOS a Numero de color. |
| nRGB() | Genera un valor RGB. |
| nGetForeRGB() | Retorna el valor RGB activo de un string de Color. |
| nGetBackRGB() | Retorna el valor RGB de Fondo de un string de Color. |
| GetPixel() | Retorna el color RGB de un pixel. |
| SetTextColor() | Fija un específico color de Texto |
| SetBkColor() | Fija un específico color de Fondo. |
| SetBkMode() | Fija un específico modo de Fondo. |
| GetTextColor() | Retorna el color texto activo. |
| GetBkColor() | Retorna el color de Fondo activo. |
| GetSysColor() | Recupera el system color del elemento especificado. |

Comunicaciones

| | |
|-------------------|----------------------------------|
| OpenComm() | Abre una puerta de comunicación. |
|-------------------|----------------------------------|

| | |
|------------------------|-------------------------------------------------------------------|
| Closecomm() | Cierra un dispositivo de comunicación y libera memoria. |
| ReadComm() | Lee el contenido de una puerta de comunicación . |
| FlushComm() | Transfiere todos los caracteres de la transmisión o la recepción. |
| GetCommError() | Recupera el valor del último error de comunicación. |
| BuildCommDCB() | Transfiere un string de un dispositivo a un Control Block |
| WriteComm() | Escribe hacia la puerta de comunicaciones |
| EnableCommNot() | Activa la notificación automática de mensajes. |

Fecha / Hora

| | |
|------------------|-----------------------------|
| SetDate() | Cambia la fecha del sistema |
| SetTime() | Cambia la hora del sistema |

Facilidades de depuración

OutputDebugString() Despliega un msg de debugging a Microsoft dbwin.exe

DialogBoxes

| | |
|-----------------------------|-------------------------------------------------------------------------|
| CreateDlg() | Crea una DialogBox Modales desde Recursos. |
| CreateDlgIndirect() | Crea una DialogBox Modales desde Recursos de memoria. |
| DialogBox() | Crea una DialogBox Modal desde Recursos. |
| DialogBoxIndirect() | Crea una DialogBox Modal desde Recursos de memoria. |
| EndDialog() | Cierra una DialogBox modal y libera los recursos. |
| GetDialogBaseUnits() | Retorna la unidades bases usadas para crear la DialogBox. |
| GetDlgItem() | Recupera el manejador de un control. |
| MapDialogRect() | Convierte las unidades de la DialogBox en unidades de pantalla (pixels) |
| nDlgBox() | Función de prueba automática de DialogBox. |

Mantenición de Directorios.

| | |
|---------------------|---------------------------------------------------|
| CurDrive() | Retorna el Driver activo como una letra A,B,C. |
| aDrives() | Retorna un arreglo con los drivers validos |
| IsCdRom() | Verifica si el Driver es un CDRom |
| IsDiskette() | Verifica si el Driver es una Dikettera |
| IChDir() | Cambia de un directorio a otro |
| lIsDir() | Verifica si existe un directorio. |
| IMkDir() | Crea un nuevo directorio |
| IRmdir() | Elimina un directorio |
| GetWinDir() | Recupera el path del directorio de Windows |
| GetSysDir() | Recupera el path del directorio de Windows System |

DLLs

| | |
|-------------------------|---------------------------------------------------------------------|
| LoadLibrary() | Carga una Librería de Enlace Dinámico para usarla |
| FreeLibrary() | Libera la librería DLL en uso o decrementa su contador de "En uso". |
| GetProcAddress() | Recupera la dirección del módulo de la función |
| CallDll() | Llama una función puesta dentro de un DLL |

Dibujo

| | |
|----------------|-------------------------|
| Arc() | Dibuja un arco elíptico |
| Chord() | Dibuja un cordón |

| | |
|-----------------------|-------------------------------------------------------------|
| Ellipse() | Dibuja una elipse |
| FloodFill() | Rellena un área de la pantalla |
| LineTo() | Dibuja una línea de la posición actual o otra posición |
| MoveTo() | Cambia el punto de origen para empezar a dibujar una línea. |
| Pie() | Dibuja un sector de un Pastel en un arco elíptico |
| PolyPolygon() | Dibuja varios polígonos |
| Rectangle() | Dibuja un rectángulo gráfico |
| Say3D() | Dibuja texto con efecto 3D |
| WndBoxIn() | Dibuja un rectángulo 3D intercalado |
| WndBoxRaised() | Dibuja un rectángulo 3D elevado |
| WndRaised() | Da efecto elevado a un control |

Ejecutando otras aplicaciones

| | |
|--------------------------|------------------------------------------------------------------|
| WaitRun() | Ejecuta otras aplicaciones Windows (con espera) |
| WinExec() | Ejecuta otras aplicaciones Windows (sin espera) |
| ExitWindowsExec() | Salen de windows, Ejecuta una aplicación DOS y retorna a Windows |

Manejo de Archivos

| | |
|-------------------------|------------------------------------------------------------------|
| cFileDisk() | Recupera el driver desde un nombre de archivo con path. |
| cFileExt() | Recupera la extensión desde un nombre de archivo. |
| cFileMask() | Recupera la mascara desde un nombre de archivo especificado. |
| cFileName() | Recupera el nombre de archivo sin path |
| cFileNoExt() | Recupera el nombre de archivo sin path ni extensión. |
| cFileNoPath() | Recupera el nombre de archivo sin path (como cFileName()) |
| cFilePath() | Recupera el path desde el nombre de archivo con path. |
| cFileSubDir() | Recupera el nombre del subdirectorío desde el nombre de archivo. |
| cNewFileName() | Recupera un nuevo nombre de archivo desde un nombre base. |
| cTempFile() | Recupera un nuevo archivo temporal. |
| FSize() | Recupera el tamaño de un archivo sin abrirlo. |
| LzCopyFile() | Copia y expande un archivo Windows comprimido. |
| GetFTime() | Recupera la fecha y hora de un archivo. |
| SetFTime() | Fija una diferente hora y fecha para un archivo |
| SetHandleCount() | Incrementa el máximo número de archivos abiertos. |

Manejo de archivos de Ayuda

| | |
|-----------------------|------------------------------------------------------------------|
| GetHelpFile() | Recupera el archivo help seleccionado por defecto. |
| GetHelpTopic() | Recupera el Tópico de ayuda por defecto |
| HelpIndex() | Muestra índice del archivo help por defecto |
| HelpPopup() | Muestra el popup de contenido de la ayuda |
| HelpSearch() | Muestra el DialogBox de búsqueda en el archivo help por defecto. |
| HelpSetPos() | Cambia la posición del de la ventana WinHelp |
| HelpSetTopic() | Selecciona un Tópico de ayuda por defecto |
| HelpTopic() | Muestra un Tópico de ayuda |
| SetHelpFile() | Selecciona un archivo de ayuda por defecto |
| WinHelp() | Activa el programa WinHelp de Windows |

Iconos

| | |
|----------------------|--------------------------------------------------------|
| nlcons() | Retorna el numero de iconos |
| LoadIcon() | Carga un específico recurso icono |
| ExtractIcon() | Recupera un manejador de Icono desde un EXE, DLL o ICO |

DrawIcon() Dibuja un icono

Archivos INI

GetProfInt() Recupera un valor entero de una entrada del archivo Win.ini
GetProfStr() Recupera un valor string de una entrada del archivo Win.ini
GetPvpProfile() Recupera un valor entero de una entrada de un archivo INI
GetPvpProfString() Recupera un valor string de una entrada de un archivo INI
WriteProfS() Escribe una entrada en Win.ini
WritePProString() Escribe una entrada en un archivo INI.

Keyboard

GetASyncKeyState() Retorna se una tecla Presionada o soltada
GetKeyState() Retorna el estado de teclas virtuales especificas.
GetKeyToggle() Retorna el estado de una tecla toggled
KeyToggle() Toggles una tecla especifica

Mouse

SetCapture() Captura toda actividad del mouse en una ventana.
ReleaseCapture() Permite que las otras aplicaciones usen el mouse
GetCapture() Recupera el manejador de la ventana que capturo el mouse.

Manejo de archivos NTX/MDX

DelDbfMdx() Restablece un archivo MDX existente a un archivo DBF DBFMDX
cNtxKey() Retorna la descripción de la clave de un archivo índice NTX.
NtxPos() Retorna la posición relativa de un numero de registro.

Manipulación Numérica

B2Bin() Convierte un byte a un string de 1byte de largo
C2Hex() Convierte un byte a un string hexadecimal (2 caracteres)
D2Bin() Convierte un float a un string de 8 bytes de largo
12Hex() Convierte un word (2 bytes) a un string hexadecimal (4 caracteres)
L2Hex() Convierte un long (4 bytes) a un string hexadecimal (8 caracteres)
lAnd() Recupera el AND lógico de valores numéricos
nAnd() Genera un valor AND numérico desde algunos números.
nOr() Genera un valor OR desde algunos números.
nHiByte() Retorna el Byte Alto de una Word (2 bytes)
nLoByte() Retorna el Byte Bajo de una Word (2 bytes)
nHiWord() Retorna el HiWord de un long (4 bytes)
nLoWord() Retorna el LoWord de un long (4 bytes)
nMakeLong() Genera un Long (4bytes) desde dos Words (2 bytes)
MakeIntResource() Convierte un recurso ID entero a Formato Windows.
nStrHash() Retorna un algoritmo de hasing sobre un string.
nRandom() Genera un numero Random

OemAnsi

| | |
|--------------------|-----------------------------------------------------------|
| AnsiToOem() | Rutina de Conversión de String Windows a OEM (Ansi/ASCII) |
| IsAnsi() | Consulta si es un String Ansi |
| IsOem() | Consulta si es un String OEM (ASCII) |
| OemToAnsi() | Rutina de Conversión de OEM a string Windows (Ascii/Ansi) |

OOPS

| | |
|----------------------|-------------------------------------------------------------------------|
| aOData() | Retorna un arreglo con todos nombre de los Objeto de Clases de los DATA |
| OCClone() | Recupera un duplicado de un Objeto CA-Clipper |
| OSend() | Función para enviar mensajes genéricos a un Objeto |
| ASend() | Función para enviar mensajes genéricos a un arreglo de Objetos |
| ISaveObject() | Guarda un objeto en Disco |
| uLoadObject() | Recupera un objeto de Disco |
| ISaveVar() | Guarda un arreglo a Disco |
| uLoadVar() | Recupera un arreglo de disco |

Seguridad / Protección de Copia

| | |
|--------------------|------------------------------------------------------|
| nStrCRC() | Retorna el CheckSum de un texto (Suma de Validación) |
| nFileCRC() | Retorna el CheckSum de un archivo |
| Encrypt() | Encripta un texto con una clave opcional |
| Decrypt() | Desencripta un texto con una clave opcional |
| nSerialHD() | Retorna el numero de serie de un disco duro |

Sound

| | |
|-----------------------|---------------------------------------------------------|
| MessageBeep() | Ejecuta un system alert con sonido desde Wave o un tone |
| MsgBeep() | MessageBeep con tono Beep |
| SndPlaySound() | Ejecuta un archivo WAV |
| SndPlayRes() | Ejecuta un archivo WAV desde recursos |

Strings (Cadenas de Caracteres)

| | |
|----------------------|----------------------------------------------------------------------|
| cValToChar() | Convierte un valor entero a un string |
| OemToAnsi() | Cambia un String OEM DOS a un string ANSI Windows. |
| AnsiToOem() | Cambia un String ANSI Windows a un String OEM DOS. |
| xPadL() | Llena por la izquierda un string hasta una cierta dimensión en pixel |
| xPadC() | Llena por ambos lados un string hasta una cierta dimensión en pixel |
| xPadR() | Llena por la derecha un string hasta una cierta dimensión en pixel |
| B2Bin() | convierte un byte en un string de largo 1 Byte |
| StrCapFirst() | deja el primer carácter en mayúsculas y el resto en minúsculas. |
| StrChar() | Extrae un carácter simple de un string |
| StrToken() | Extrae un token desde un string |

System

| | |
|----------------------|---------------------------------------------|
| InportByte() | Lee un byte desde un puerto de Hardware |
| InponWord() | Lee una palabra desde un puerto de HardWare |
| nPeekByte() | Lee un Byte de Memoria |
| nPeekWord() | Lee una palabra de memoria |
| nPokeByte() | Escribe un Byte a Memoria |
| nPokeWord() | Escribe una palabra a Memoria |
| OutportByte() | Envía un byte a un puerto de Hardware |