



## Contenido de ayuda de ScriptTools

Para mayor información acerca de Ayuda, pulse F1 o seleccione Uso de ayuda en el menú Ayuda.

### Cómo...

- [Utilizar nuestro sistema de ayuda](#)
- [Crear un nuevo guión](#)
- [Compilar y probar un guión](#)
- [Ejecutar guiones de ScriptTools](#)
- [Crear un Cuadro de diálogo](#)
- [Cambiar una definición de un cuadro de diálogo](#)
- [Crear un suceso](#)
- [Editar archivos de guiones](#)
- [Registrar una serie de acciones](#)
- [Insertar un nombre de ventana](#)
- [Configurar el registro de conducta de una sesión](#)
- [Trabajar con aplicaciones del DOS](#)
- [Depurar su guión](#)

### Funciones de idioma

- [Funciones de ScriptTools por categoría](#)
- [Lista alfabética de funciones de ScriptTools](#)
- [Funciones del sistema](#)
- [Variables del sistema](#)
- [Nombres de las variables](#)
- [Tipos de datos](#)
- [Operadores](#)

### Menús

- [Archivo](#)
- [Editar](#)
- [Crear](#)
- [Opciones](#)
- [Ventana](#)
- [Ayuda](#)

## Acerca de ScriptTools

Use ScriptTools para crear guiones para Windows.

Usted puede

- Realizar una serie de acciones mientras ScriptTools escribe el archivo de guiones por usted.
- Diseñar y crear fácilmente sus propios cuadros de diálogo para entradas del usuario.
- Definir el teclado, mouse u otros sucesos para iniciar la ejecución de un guión.
- Agregar guiones compilados al menú o barra de herramientas de cualquier aplicación de PC Tools para Windows y ejecutarlos como podría ejecutar cualquier otro comando.

## Teclas de método corto de ScriptTools



Imprimir esta página.

<b>Tecla</b>	<b>Acción o comando</b>
Alt + letra subrayada	Activar el menú o el comando.
Alt	Activar la barra del menú.
Ctrl + A	Crear un nombre de anexo.
Ctrl + C	Copiar
Ctrl + D	Crear un diálogo
Ctrl + E	Crear un suceso
Ctrl + G	Ir a la línea
Ctrl + L	Crear idioma
Ctrl + R	Crear sesión de grabación
Ctrl + T	Comprobar ejecución
Ctrl + V	Pegar
Ctrl + X	Cortar
Ctrl + Z	Deshacer
Supr (Del)	Eliminar
F1	Ayuda

Mayúsculas, Ctrl, Pausa      Para detener un guión desbocado , pulse las teclas en secuencia, no al mismo tiempo. La tecla de Pausa es usualmente F15.

## Guión desbocado

Los errores de tiempo de ejecución de un programa pueden colocar a su guión en un interminable ciclo u otra condición errónea.

- Para detener un guión que no está haciendo lo que se esperaba pulse Mayúsculas, luego Ctrl y luego Pausa (F15). Pulse las teclas una detrás de otra, no al mismo tiempo.

## Menú Archivo

Use los comandos de este menú para administrar los archivos de guiones.

<u>Nuevo</u>	Crea una nueva ventana de fichero llamada Sin título.
<u>Abrir</u>	Abre una nueva ventana que contiene un archivo de guiones existente.
Cerrar	Cierra el fichero en la ventana activa.
Guardar	Guarda una copia del fichero en uso. Si éste es una ventana sin título, se le pedirá que le ponga un nombre.
<u>Guardar como</u>	Graba una copia del fichero en uso, con un nombre de archivo y ruta diferentes.
<u>Imprimir</u>	Imprime todo o parte del fichero en uso.
<u>Comprobar ejecución</u>	Compila el código fuente del guión en la ventana activa. Si se compila sin errores, puede ejecutar el guión desde el cuadro de diálogo Comprobar ejecución.
Salir	Sale de ScriptTools. Si no ha grabado los archivos en la ventana de edición, se le pedirá que lo haga.

## Menú Archivo

Nuevo

Abrir

Cerrar

Salir

Imprimir

Comprobar ejecución

Guardar

Guardar como



## **Nuevo**

Abre una nueva ventana de edición llamada Sin título

Las ventanas existentes aparecen en mosaico en la pantalla.



## Abrir

Abre un fichero de guiones existente.

Ver también

[Cuadro de diálogo Abrir archivo](#)



### **Cuadro de diálogo Abrir archivo**

Selecciona y abre un archivo de guiones de ScriptTools existente.

#### **Opciones del cuadro de diálogo**

Abrir nombre de archivo La selección por defecto es \*.SCT.

- |             |  |
|-------------|--|
| Directorios | Se muestra el nombre de la ruta en uso. Use el árbol de directorios para elegir un directorio. |
| Archivos    | La lista de los archivos de guiones en el directorio activo.                                   |
| Unidades    | Use el cuadro de lista para seleccionar una unidad.  |



## **Guardar como**

Seleccione Guardar como para grabar la versión del fichero con un nombre diferente o en un directorio diferente. La versión original se mantiene intacta.

[Ver también](#)

[Cuadro de diálogo Guardar como](#)



## Cuadro de diálogo Guardar como

Graba un fichero con un nombre diferente o en una localización diferente sin modificar el archivo original.

### Opciones del cuadro de diálogo

Nombre de archivo	Aparece el nombre del fichero en uso. Se puede editar el nombre o cambiar la unidad o directorio para guardarlo con su antiguo nombre en un nuevo lugar.
Directorios	Se muestra el nombre de la ruta en uso. Use el árbol de directorios para seleccionar un directorio.
Archivos	La lista de los archivos de guiones existentes en el directorio en uso.
Unidades	Use el cuadro de lista desplegable para seleccionar una unidad.



## **Imprimir**

Imprime todo el archivo de guiones activo, o sólo un bloque especificado de líneas.

Se puede especificar cualquier impresora que haya configurado.

[Ver también](#)

[Cuadro de diálogo Imprimir](#)



### **Cuadro de diálogo Imprimir**

Imprime todo el archivo de guiones en uso o sólo un bloque de líneas especificado.

- ▶ Para detener la impresión en cualquier momento, pulse Esc o seleccione Cancelar.
- ▶ Para seleccionar una impresora diferente o cambiar la orientación de la impresión y el tamaño del papel, seleccione el botón Instalar.

#### **Opciones del cuadro de diálogo**

- |                    |  |
|--------------------|--|
| Imprimir           | Por defecto, se imprime todo el archivo. Seleccione Sólo las líneas para especificar la gama de números de línea que desea imprimir.                           |
| Dispositivo/Puerto | Se muestra el nombre de la impresora en uso. Seleccione Instalar para cambiar de impresora o cualquiera de sus opciones, como orientación apaisada o vertical. |

## Menú Editar

Use los comandos de este menú para manipular, buscar o reemplazar cadenas de texto, y para desplazar el cursor a una línea especificada.

Deshacer	Restaura el último texto eliminado.
Copiar	Copia el texto seleccionado en el portapapeles.
Cortar	Elimina la selección activa y la coloca en el portapapeles. Si no desea borrar la selección, elija Copiar.
Pegar	Pega texto del portapapeles dentro del archivo activo y en la posición del cursor. El texto pegado reemplaza cualquier texto seleccionado.
Eliminar	Elimina la selección activa. Utilice Deshacer para recuperar un texto borrado.
<u>Buscar/Reemplazar</u>	Busca el texto especificado y lo reemplaza de acuerdo con las instrucciones. Busca hacia adelante, a partir de la posición del cursor hasta que llega al final del archivo. Marque el cuadro Emparejar mayúsculas/minúsculas para realizar esa acción.
Ir a la línea	Desplaza el cursor al número de código de línea que usted indica. Este comando es útil cuando corrige los errores en el archivo .ERR, que indica números de línea para los errores. La línea activa aparece en la línea de estado de la ventana de edición.
Seleccionar todo	Selecciona todo el texto del archivo en uso.

**Edit Menu**

Deshacer

Cortar

Eliminar

Ir a la línea

Copiar

Paste

Buscar / Reemplazar

Seleccionar todo



## **Buscar/Reemplazar**

Busca una cadena de caracteres y, si se le indica, la reemplaza con otra cadena. Se puede hacer coincidir mayúsculas y minúsculas, así como buscar en dirección inversa.

Ver también

[Cuadro de diálogo Buscar y Reemplazar](#)



### Cuadro de diálogo Buscar y reemplazar

Busca, o busca y reemplaza, caracteres en su archivo.

#### Opciones del cuadro de diálogo

Buscar: Escriba la cadena de caracteres que quiere buscar o reemplazar.

Cambiar a: Si desea reemplazar la cadena con otra, escriba la nueva.

Emparejar mayúsculas y minúsculas

Marque Emparejar mayúsculas y minúsculas para encontrar o reemplazar solamente las cadenas de caracteres que coinciden exactamente con las mayúsculas y minúsculas que usted escribió como criterio.

Buscar hacia atrás Buscar hacia el extremo inicial del fichero.

Botones Cambiar Buscar siguiente  
Cambiar todo

## Menú Crear

Use los comandos de este menú para automatizar la codificación de las instrucciones de idioma de ScriptTools.

Nombre de anexo Crea el nombre de anexo que identifica cualquier ventana de la pantalla.

Suceso Muestra el cuadro de diálogo de producción de sucesos que puede utilizar para crear el código para definir un suceso de pantalla, teclado, mouse, menú, ventana o tiempo.

Diálogo Inicia el editor de cuadro de diálogo Crear diálogo, donde utiliza una maqueta para diseñar cuadros de diálogo que aceptan entradas de parte del usuario.

Idioma Introduce prototipos de instrucción de idioma dentro del archivo en uso. Usted substituye sus propios nombres de variables y otros parámetros.

Sesión de grabación Inicia una sesión de grabación que genera códigos para cada acción que usted realiza, y escribe el código en la posición del cursor en el archivo en uso en la ventana de edición.

## Menú Crear

Nombre de anexo Idioma

Suceso Sesión

Diálogo



### **Nombre de anexo**

Abre el cuadro de diálogo que genera el **nombre de anexo** para cualquier ventana en la pantalla. Por la razón de que cada ventana tiene un nombre, clase y número de identificación únicos, es más fácil utilizar el cuadro de diálogo Nombre de anexo que recordar la información y escribirla.

Ver también

[Cuadro de diálogo Nombre de anexo](#)



### Cuadro de diálogo Nombre de anexo

Identifica el nombre de una ventana y la introduce como argumento en el comando Anexar, o para obtener información de ventana para los comandos Editar texto, Cuadro de lista, ComboBox y Barra de desplazamiento.

▶ Para identificar el cursor Identificar seleccione Identificar. El cursor del mouse adopta la forma de una lupa. Siga los pasos siguientes:

1. Mueva el cursor de lupa sobre la pantalla y observe la información de ventana que se modifica en el cuadro de diálogo Nombre de anexo.
2. Para grabar la información de ventana que se muestra en este momento, haga clic con el botón izquierdo del mouse.
3. Para pegar el identificador de ventana en su archivo .SCT en la posición del cursor, elija Pegar.

▶ Para pegar la información de Nombre de anexo dentro del fichero .SCT en uso en la ventana de edición, elija Pegar.

#### Opciones del cuadro de diálogo

Título El título de la aplicación. Este es el nombre que aparece bajo su icono minimizado.

Nombre de anexo El código que será escrito en el fichero .SCT.

Nombre del programa El nombre del archivo .EXE que ejecuta el programa.

ID El identificador interno de la sección de ventana del programa.

Instancia El número de instancia le indica a cuál copia del programa pertenece la ventana si se ejecuta más de una instancia del programa.

Clase La clase de ventana que describe esta sección de ventana.

Ver también

[Insertar un nombre de ventana](#)



## Diálogo

Inicia la ventana de edición de cuadro de diálogo, Crear diálogo, donde puede utilizar una maqueta para diseñar cuadros de diálogo para entradas del usuario. El código de definición del cuadro de diálogo se genera en forma automática.

Cuando selecciona Transferir diálogo en el menú Diálogo de la ventana Crear diálogo, el código de definición del cuadro se escribe en la ventana de edición activa de ScriptTools en el lugar donde se encuentra el cursor.

---

Ver también

[Crear un Cuadro de diálogo](#)

[Cambiar una definición de Cuadro de diálogo](#)



## Suceso

Use el cuadro de diálogo generador de sucesos para crear el código correcto para definir un suceso de pantalla, teclado, ventana, mouse, menú o tiempo.

Utilice los sucesos como arrancadores de una acción del guión compilado.

---

Ver también

[Cuadro de diálogo Suceso](#)



## Cuadro de diálogo Suceso

Especifica un nombre y tipo para el suceso que desea definir.

Cada tipo de suceso emplea un cuadro de diálogo diferente para definir los parámetros de los sucesos.

▶ Para usar el cuadro de diálogo que define la clase de suceso especificada, escriba el nombre del suceso y seleccione el tipo y luego Aceptar.

▶ Para crear arrancadores únicos de sucesos, combine más de uno de los sucesos definidos en el comando Esperar. También puede lanzar una acción si solamente ocurre uno de un grupo especificado de sucesos definidos.

### Opciones del cuadro de diálogo

Nombre                      Escriba un nombre de suceso. Utilice sólo caracteres alfa-numéricos y sin espacios.

Tipo                          Seleccione la clase de suceso que desea definir: pantalla, tiempo, teclado, menú, mouse o ventana.

Ver también

[Cuadro de diálogo Suceso de pantalla](#)

[Cuadro de diálogo Suceso de tiempo](#)

[Cuadro de diálogo Suceso de teclado](#)

[Cuadro de diálogo Suceso de menú](#)

[Cuadro de diálogo Suceso de mouse](#)

[Cuadro de diálogo Suceso de ventana](#)

[Comando Esperar](#)

[Comando Siempre que](#)

## Cuadro de diálogo Suceso de tiempo

Configura los parámetros de una acción lanzada por la fecha y hora.

Por defecto, el calendario y el reloj del cuadro de diálogo están ajustados para la fecha y hora en curso.

### Opciones del cuadro de diálogo

Fecha	Use la lista desplegable y el calendario para especificar una fecha.
Mes	Seleccione un mes en particular, o especifique Cualquiera.
Año	Seleccione un año en particular, o especifique Cualquiera.
Fecha	Seleccione una fecha en el calendario para elegir un día específico del mes.
Disparar	Indica el intervalo o tiempo específico para el suceso.
Cada	A intervalos de tiempo en los días especificados. Se puede indicar hasta 24 horas.
A las:	A una hora especificada en los días indicados.
Cada	Para ignorar un día específico del mes que aparece en la hoja Fecha, seleccione la opción que describe qué tan a menudo debe ejecutarse el suceso.
Día	Diariamente, independientemente del día, mes o año.
Semana	Semanalmente, independientemente del día, mes o año.
Ninguno	El suceso se ejecuta sólo en las fechas indicadas y sólo a la hora especificada.

## Cuadro de diálogo Suceso de teclado

Define una tecla especial para lanzar una acción

### Opciones del cuadro de diálogo

Tecla	Use este cuadro de lista desplegable para seleccionar una tecla de función u otra tecla especial, como Retroceso; también puede escribir un carácter alfanumérico como 8, B o b. Los sucesos de teclado identifican las teclas mayúsculas y minúsculas.
Teclas auxiliares	Seleccione los botones que especifican las teclas auxiliares para utilizar la tecla indicada en el cuadro Tecla. Por ejemplo, si especifica F10, puede también elegir Alt como tecla auxiliar y el usuario necesitará pulsar Alt+F10 para lanzar la acción. Puede seleccionar las tres teclas auxiliares (Alt, Ctrl y mayúsculas) para una sola función; también puede usar una tecla alfanumérica.
En	Especifica la ventana y módulo en donde el suceso está activado. Si la tecla debe estar activa todo el tiempo, seleccione Todas las ventanas.  Si la ventana del programa donde el suceso de teclado debe estar activo está en la pantalla, seleccione el botón Identificar para cargar la utilidad <u>Nombre de anexo</u> y recuperar un identificador de ventana en particular.
Instancia	Por defecto, el suceso de menú está activo en todas las <u>instancias</u> de un programa. Se puede especificar una instancia en particular; por ejemplo, quizá usted desee restringir las funciones en una segunda instancia de un programa.
Descartar	Marque <u>Descartar</u> para interceptar la pulsación de una tecla e impedir que se pase a la aplicación.

### Advertencia

Si marca Descartar, la pulsación de la tecla no hace más que lanzar una acción; la aplicación no puede usarla.

## Cuadro de diálogo Suceso de menú

Especifica un comando de menú como arrancador de una acción. Los sucesos de menú se asocian con ítems de menú creados por el comando Definir menú, y no con los ítems de menú estándar de una aplicación.

### Opciones del cuadro de diálogo

Menú	Escriba la identificación utilizada por el comando Definir menú o el parámetro Texto especificado por el comando Definir menú que creó el ítem de menú.
En	Especifica la ventana y módulo donde el suceso está activo. Si la tecla debe estar activa todo el tiempo, seleccione Todas las ventanas.  Si la ventana del programa donde el suceso de teclado debe estar activo está en la pantalla, seleccione el botón Identificar para cargar la utilidad <u>Nombre de anexo</u> y recuperar un identificador de ventana en particular.
Instancia	Por defecto, el suceso de menú está activo en todas las <u>instancias</u> de un programa. Se puede especificar una instancia en particular; por ejemplo, quizá usted desee restringir las funciones en una segunda instancia de un programa.
Descartar	Marque <u>Descartar</u> para interceptar la pulsación de una tecla e impedir que se pase a la aplicación.

### Advertencia

Si marca Descartar, la pulsación de la tecla no hace más que lanzar una acción; la aplicación no puede usarla.

---

Ver también

Función MENUSET

## Cuadro de diálogo Suceso de mouse

Especifica clics del mouse para lanzar una acción de un script.

Los únicos campos que se necesitan para definir un suceso de mouse son botón y clic. Los otros campos son opcionales.

### Advertencia

Use Descartar con mucho cuidado. Puede desactivar el mouse si sus acciones no se pasan al programa.

### Opciones del cuadro de diálogo

Botón	Seleccione el botón que desee: Izquierdo, Medio o Derecho. Por defecto se usa el botón derecho.
Clic	Seleccione una de estas tres acciones: Abajo Pulsar el botón del mouse. Arriba Liberar el botón del mouse. Para indicar un clic simple, seleccione Arriba. Doble Un doble clic completo, abajo y arriba dos veces.
Area	Para limitar el suceso de mouse a una área específica de la pantalla, marque el cuadro Area. Luego especifique la posición Izquierda y Superior del área activa, así como el ancho y altura en pixeles.
En	Especifica la ventana y módulo donde el suceso está activo. Si la tecla debe estar activa todo el tiempo, seleccione Todas las ventanas.  Si la ventana del programa donde el suceso de teclado debe estar activo está en la pantalla, seleccione el botón Identificar para cargar la utilidad <u>Nombre de anexo</u> y recuperar un identificador de ventana en particular.
Instancia	Especifica la <u>instancia</u> de la aplicación. La opción por defecto es Cualquiera.
Descartar	Marque <u>Descartar</u> si desea interceptar el suceso de mouse sin pasarlo a la aplicación. Esta opción, sin embargo, puede desactivar al mouse.

## Cuadro de diálogo Suceso de pantalla

Define una cadena de texto específica que aparecerá en una ventana determinada para lanzar una acción.

### Opciones del cuadro de diálogo

Texto	Escriba la cadena de texto que se debe detectar. El suceso de pantalla empareja mayúsculas y minúsculas en la cadena.
Visualizado en	Especifica la ventana y módulo donde el suceso está activo.  Si la ventana del programa donde el suceso debe estar activo está en la pantalla, seleccione el botón Identificar para cargar la utilidad <u>Nombre de anexo</u> y recuperar un identificador de ventana en particular.
Instancia	Puede especificar la <u>instancia</u> que desee. La opción por defecto es Cualquiera.

## Cuadro de diálogo Suceso de ventana

Especifica un estado de ventana que se desea utilizar para lanzar una acción.

### Opciones del cuadro de diálogo

Ventana cuando	Seleccione el estado de ventana que desea utilizar para lanzar la acción. Elija Activar, Minimizar, Maximizar, Restaurar, Destruir.
En	Especifica la ventana y módulo donde el suceso está activo. Si la tecla debe estar activa todo el tiempo, seleccione Todas las ventanas.  Si la ventana del programa donde el suceso debe estar activo está en la pantalla, seleccione el botón Identificar para cargar la utilidad <u>Nombre de anexo</u> y recuperar un identificador de ventana en particular.
Instancia	Puede especificar la <u>instancia</u> del programa que desee. La opción por defecto es Cualquiera.



### Crear Sesión de grabación

Inicia la grabación de sus acciones y las convierte en código de ScriptTools. Las instrucciones de código se escriben en el fichero de guiones activo mientras usted trabaja.

Use el cuadro de diálogo Configurar del menú Opciones para especificar la conducta por defecto para estas sesiones de grabación.

Cuando termine con las acciones que desea grabar, detenga la Sesión de grabación.

▶ Si ScriptTools se encuentra minimizado, haga clic sobre su icono y seleccione Parar en el cuadro de diálogo que se le muestra.



Si la ventana de edición está abierta en la pantalla, seleccione de nuevo Sesión de grabación en la barra de herramientas o el menú Crear.

|



## Idioma

Introduce prototipos de instrucciones de idioma en el archivo en uso. Escriba las variables que aparecen entre paréntesis y elimine los paréntesis.

Hay una breve descripción de cada prototipo.

---

Ver también

[Funciones de ScriptTools](#)

[Funciones del sistema](#)

[Variables del sistema](#)

[Tipos de variables y datos](#)



### **Cuadro de diálogo Idioma**

Muestra un prototipo de la instrucción de idioma seleccionado y una breve descripción de su función.

Seleccione Pegar para introducir el prototipo dentro de su código en la posición del cursor. Luego escriba las variables que se muestran en paréntesis angulares y elimine los paréntesis.

Si introduce una variable como una cadena literal, como un nombre real de archivo o de ventana, inclúyalo dentro de dobles comillas. Si escribe el nombre de una variable, como OldPath\$, no use las dobles comillas.

## Menú Opciones

Use los comandos de este menú para especificar sus valores por defecto preferidos.

Personalizar      Abre las series de cuadros de diálogo donde puede modificar comandos en menús y la barra de herramientas.

Barra de herramientas    Al seleccionarla, se muestra la barra de herramientas.

Barra de estado      Al seleccionarla, se muestra la barra de estado, que muestra el número de la línea activa en el archivo de guiones.

### Configuración de registro

Abre el cuadro de diálogo donde puede configurar los valores por defecto para las sesiones grabadas.

## Menú Opciones

Personalizar

Barra de herramientas

Configuración de registro

Barra de estado



## Cuadro de diálogo Configuración de registro

Especifica la conducta por defecto de ScriptTools durante una sesión de grabación, cuando ScriptTools registra sus acciones y las escribe en el fichero de guiones en forma automática.

Las opciones que especifica aquí no tienen efecto en los guiones que crea en forma manual o con otras funciones del menú Crear, como Suceso.

### Opciones para la creación de códigos

**Variables** Genera nombres de variables para números enteros con base en el objeto que ese número representa; por ejemplo, Wk\_AboutNotepad = 59.

**Comentarios** Introduce comentarios automáticamente en los códigos creados; por ejemplo:

```
; Mover ventana principal de Bloc de notas a las coordenadas 100,200  
Move To 100, 200
```

**Tamaño de aprendizaje** Registra las dimensiones y la posición de la ventana principal de una aplicación cuando arranca por primera vez. Use esta opción para asegurarse de que los clics del mouse en la ventana se harán en el lugar correcto en el guión ya compilado.

### Opciones de configuración

**Velocidad de muestra del mouse**

La velocidad de muestra del mouse especifica qué tan a menudo la posición del mouse se escribe en el archivo de guiones cuando se mantiene la presión sobre uno de sus botones. Escriba la velocidad en décimas de segundo, de 0 a 50.

**Umbral de pausa** La configuración del umbral de pausa especifica el número de segundos que usted debe esperar para que la Sesión de grabación escriba una instrucción de pausa en el guión. Escriba el número de segundos, de 1 a 50, que deben transcurrir antes de que una instrucción de pausa se escriba automáticamente. Escriba 0 para desactivar las instrucciones de pausa. También puede introducir instrucciones de pausa en forma manual después que el guión ha sido grabado.

### Aprendizaje de la envolvente

La configuración de aprendizaje de la envolvente especifica cuántas operaciones se aprenden en las envolventes (shells) de lanzamiento del programa de Windows como MultiDesk, Administrador de programas, Administrador de archivos y Ejecutivo MS-DOS. Hay tres opciones: Apagado, Parcial, y Completo.

**Minimizar durante la sesión de grabación**

Minimiza la ventana de ScriptTools en un icono cuando una Sesión de grabación está activa. Si hace clic sobre el icono minimizado, se le muestra el cuadro de diálogo Sesión de grabación para permitirle que detenga la grabación y vuelva a la ventana completa de ScriptTools o seguir con el procedimiento.

Ver también

[Grabación de una serie de acciones](#)

## Apagado

Cuando Aprendizaje de la envolvente se inactiva, ScriptTools no escribe ninguna operación de envolvente en el código automáticamente.

Las operaciones de la envolvente incluyen cualquier acción a realizar utilizando el programa especificado como envolvente en su archivo SYSTEM.INI de Windows.

## Parcial

Cuando Aprendizaje de la envolvente es Parcial, ScriptTools escribe todas las acciones en el código en forma automática, con excepción de los clics del mouse.

En esta modalidad, no inicie aplicaciones con clics del mouse.

## Completo

Cuando Aprendizaje de la envolvente es Completo, ScriptTools escribe todas las acciones en el código automáticamente incluyendo los clics, arrastres y movimientos del mouse.

Para posicionar objetos, como los iconos de archivos, este modo es muy importante: el gui3n no se ejecuta correctamente si los objetos no se encuentra en la posici3n esperada.

### **Velocidad de muestra del mouse**

Un intervalo de muestra de baja velocidad, como cada décima de segundo, proporciona una repetición del mouse más suave cuando se ejecuta el guión, aunque el mismo es más largo. Un intervalo de muestra de alta velocidad produce una repetición más grosera, pero un guión más pequeño.

En general, un intervalo de muestra de alrededor de cada 4 décimas de segundo proporciona un buen resultado, aunque se puede necesitar una velocidad más baja para los programas que usan intensivamente el mouse, como los paquetes de pintura y dibujo.

## **Cambiar**

Cambia la ocurrencia actual de una cadena y busca la próxima.

**Cambiar todos**

Cambia todas las ocurrencias de la cadena en todo el archivo.

## Buscar siguiente

Busca la siguiente instancia de la cadena especificada.

## Menú Ventana

Use los comandos de este menú para organizar las ventanas en la ventana de edición de ScriptTools.

Los nombres de las ventanas de archivos .SCT abiertos aparecen en la parte inferior del menú Ventana.

► Para seleccionar un archivo en su ventana, haga doble clic sobre su nombre en la lista.

Mosaico Coloca en mosaico todas las ventanas de archivo abiertas

Cascada Organiza todas las ventanas abiertas en una cascada a partir del borde superior de la pantalla.

Organizar iconos Organiza todas las ventanas de archivo que aparecen como iconos.

Cerrar todas Cierra todas las ventanas de fichero, pidiéndole que grabe los archivos aún no guardados.



## Menú Ayuda

Use los comandos de este menú para mostrar información sobre el sistema de ayuda de ScriptTools y de Microsoft Windows.

Contenido            Abre la tabla de contenido para este archivo de ayuda.

Buscar ayuda sobre:    Muestra una lista de términos en índice para este archivo de ayuda. Siga las instrucciones del cuadro de diálogo para buscar un tema de ayuda.

Uso de ayuda            Proporciona información en detalle sobre el uso de la ayuda en línea de Microsoft Windows.

### Acerca de ScriptTools

Muestra la versión de la aplicación, un aviso de derechos reservados, así como información del entorno de Windows y del usuario. Esto incluye el Modo de Windows (mejorado o estándar), la presencia de un co-procesador matemático en su ordenador, y la cantidad de memoria y recursos libres en Windows.

---

Ver también

[Uso de nuestro sistema de ayuda](#)

## **Autores del sistema de ayuda**

Colin B.

Ed B.

Hanna M.

Jack D.

Robin D.

## **Memoria y recursos libres**

Es la cantidad de memoria sin usar, en kilobytes, y el porcentaje de recursos del sistema libres y disponibles en Windows.

En el modo mejorado, la memoria incluye memoria virtual, que es el espacio del disco que se usa como memoria y no como área de almacenamiento de archivos. Los recursos del sistema incluyen varias áreas de la memoria dejadas a un lado por las aplicaciones de Windows en ejecución. Use esta información para formarse una idea de cuándo su sistema está llegando al límite de su capacidad.

**Cómo**

## Creación de un nuevo guión

Para crear un nuevo guión siga los pasos siguientes:

1. Seleccione Nuevo en el menú Archivo.
2. Cree el guión fuente usando uno o más de los métodos siguientes:
  - ▶ Escriba las instrucciones de idioma en la ventana de edición de ScriptTools. Puede usar el cuadro de diálogo Crear idioma , pegar un prototipo de comando o función en su guión y luego hacer las modificaciones al prototipo.
  - ▶ Use Crear sesión para generar instrucciones de idioma mientras realiza una serie de acciones.
  - ▶ Use Crear diálogo para definir un cuadro de diálogo.
  - ▶ Use Crear suceso para definir un mouse, teclado u otro suceso.
3. Compile el archivo fuente usando Comprobar ejecución en el menú Archivo.

Si el compilador no encuentra errores, puede ejecutar el guión compilado inmediatamente. Si el compilador le informa que hay errores, utilice el fichero .ERR para saber cuáles son y corregirlos. Luego seleccione Comprobar ejecución para compilar el guión otra vez.

---

Ver también

[Grabación de una serie de acciones](#)

[Comprobación de un guión](#)

[Depuración de su programa](#)

## Creación de un cuadro de diálogo

La forma más fácil de crear un cuadro de diálogo es usar la utilidad Hacer diálogo, que le muestra una maqueta del cuadro con la que puede trabajar. Las instrucciones de idioma resultantes se escriben automáticamente en su fichero de guiones cuando selecciona Transferir en el menú Diálogo.

Para crear un código para una definición de cuadro de diálogo en forma rápida y fácil, siga los pasos siguientes:

1. En la ventana de edición de ScriptTools, seleccione Diálogo en el menú Crear. Aparece la ventana Hacer diálogo.
2. Seleccione Nuevo en el menú Hacer archivo de diálogo. Aparece el cuadro de diálogo Propiedades del diálogo.
3. Escriba un nombre y título para su cuadro de diálogo en la casilla de diálogo Propiedades del diálogo.  
No es necesario escribir el tamaño del cuadro de diálogo. Es más fácil definirlo si arrastra la maqueta hasta que tenga las dimensiones deseadas. Para colocar el cuadro de diálogo con precisión en la pantalla, escriba las coordenadas de su esquina superior izquierda en los cuadros de texto Posición.
4. En el menú Controles de Hacer diálogo, seleccione cuadro de lista, botón de radio, casilla para marcar y otros controles.  
Aparece un nuevo control de cuadro de diálogo en la esquina superior izquierda de la maqueta. Arrástrelo a la posición que desea y ajuste el tamaño de la maqueta para que los controles se adapten cómodamente.
5. Seleccione Transferir diálogo en el menú Diálogo de Hacer diálogo para copiar la definición de su cuadro dentro del guión de la ventana de edición de ScriptTools activa.

▶ Para visualizar un cuadro de diálogo definido en un guión compilado, use la instrucción de idioma Dialog.

Ver también

[Cambiar una definición de cuadro de diálogo](#)

[Diálogo](#) for a description of the Dialog function and an extended example.

## Creación de un suceso

Utilice los sucesos definidos como puntos de lanzamiento de las acciones de archivo de guiones. Por ejemplo, puede definir un suceso de teclado de tal forma que cuando el usuario pulse una tecla determinada, se realice en forma automática una acción.

Cada cuadro de diálogo de suceso tiene ayuda sensible al contexto, que le proporciona información en detalle sobre los parámetros de suceso.

Siga estos pasos para crear el código de un suceso definition.

1. Elija Suceso en el menú Crear de la ventana de edición de ScriptTools y escriba un nombre descriptivo de código para el suceso; ejemplo: EventCloseFile.
2. Elija el suceso que desea definir: pantalla, hora, teclado, menú, mouse o ventana.
3. Escriba la información sobre el suceso en el cuadro de diálogo que se le muestra.
4. Cuando elija Aceptar para cerrar el cuadro de diálogo de suceso, la definición del suceso se pega dentro del archivo en uso en la ventana de edición de ScriptTools en la posición del cursor.

▶ Para probar un suceso, utilice el comando Esperar.

Ver también

Esperar for an example that tests the occurrence of an event.

## Edición de archivos de guiones

El editor de Scripttools opera como la mayoría de editores de Windows. Sus características las describimos abajo.

- Mover el cursor Las teclas Inicio (Home) y Fin (End) desplazan el cursor al principio y final de la línea. Control+Fin lo mueve al final del archivo, Control+Inicio lo coloca al principio del archivo. RePág y AvPág lo mueven arriba y abajo el número de líneas que se muestra en la pantalla.
- Seleccionar texto Haga clic en un extremo de la cadena, y pulse la tecla de mayúsculas y haga clic sobre el otro extremo; otros métodos: mantenga la presión sobre la tecla de mayúsculas y pulse un tecla de flecha o mantenga la presión sobre el botón del mouse y arrastre el puntero sobre el texto que desea seleccionar.
- Eliminar Pulse la tecla Supr (Del) o Retroceso para borrar un solo carácter o el texto seleccionado. También puede sobrescribir un texto seleccionado.
- Deshacer eliminación Seleccione Deshacer en el menú Editar para recuperar el último texto que borró.
- Buscar Seleccione Buscar/Reemplazar en el menú Buscar y escriba los caracteres que desea buscar. Marque la casilla Emparejar mayúsculas/minúsculas para realizar esta acción.
- Para buscar hacia el principio del archivo, seleccione Buscar hacia atrás.
- Para buscar la próxima ocurrencia de una cadena, seleccione Buscar siguiente.
- Reemplazar Seleccione Buscar/Reemplazar en el menú Buscar, escriba la palabra que desea sustituir y su reemplazo. Marque la casilla Emparejar mayúsculas/minúsculas si es eso lo que desea. Luego elija Cambiar para reemplazar solamente la próxima ocurrencia, o Cambiar todos para reemplazar todas las ocurrencias.

## Grabación de una serie de acciones

Una de las formas más fáciles de iniciar un archivo de guiones es permitir a ScriptTools grabar las acciones que realiza en el entorno de Windows. Después de haber creado un archivo con este método, puede modificar el código resultante para ajustarlo con más precisión a sus necesidades.

También puede emplear una sesión de grabación para introducir un código en un guión ya existente.

▶ Para definir la conducta por defecto en las sesiones de grabación, utilice el cuadro de diálogo Configuración de registro en el menú Opciones.

▶ Para grabar una secuencia de acciones y escribir instrucciones de código para las acciones dentro del guión en forma automática, haga lo siguiente:

1. Seleccione Sesión de grabación en el menú Crear de la ventana de edición de ScriptTools.

Si seleccionó Minimizar durante la sesión de grabación en el cuadro de diálogo Configuración de registro del menú Opciones, ScriptTools se minimiza en un icono que se enciende y se apaga.

2. Realice la secuencia de acciones que desea introducir en el guión.

3. Cuando haya terminado, haga clic sobre el icono de ScriptTools y seleccione Parar grabación en el cuadro de diálogo que se le muestra; también puede seleccionar el menú Crear en la ventana de edición y desactivar el comando Sesión.

ScriptTools introduce las instrucciones que describen las acciones que realizó, dentro del archivo en uso y en el lugar donde está el cursor en la ventana de edición.

### Nota

Si el guión que archivó desplaza un ítem del escritorio, como un icono de carpeta o de una aplicación que no está en ejecución, tiene que editar el guión resultante. Agregue una instrucción MouseMove después de la instrucción Attach (anexo) para que el ítem sea desplazado, como lo mostramos en el siguiente ejemplo. Las instrucciones agregadas aparecen en color azul.

```
; Use la ventana padre "USER #327691"
```

```
Attach "USER_#327691", 1
```

```
    MouseClick Left Button Down at 805,725
```

```
    MouseMove 805, 723
```

```
; Use la ventana padre "WNCPS1_CPS_spidr_slave1"
```

```
Attach "WNCPS1_CPS_spidr_slave1", 1
```

```
    MouseMove 819 441 ; LINEA AGREGADA para desplazar el ítem
```

```
    MouseClick Left Button Up At 819, 441
```

[Ver también](#)

[Configuración de la conducta de la Sesión de grabación](#)

## Trabajo con aplicaciones del DOS

ScriptTools no puede grabar una aplicación del DOS, ni identificar una ventana de aplicación del DOS. Después de invocar al DOS desde ScriptTools, todas las pulsaciones de teclas debe introducirse con instrucciones de comando Type.

Aunque la utilidad Sesión de grabación de ScriptTools no puede registrar pulsaciones de teclas, sí puede grabar clics del mouse dentro de una aplicación del DOS. Sin embargo, la advertencia estándar sobre la relación entre tamaño de ventana y clics del mouse sí es aplicable. Puede ser difícil también imaginarse a dónde pertenecen los clics del mouse en relación con las pulsaciones de tecla omitidas que necesita escribir con el comando Type.

A causa de que tanto ScriptTools como el DOS necesitan poco tiempo para iniciar y cerrar, agregue una instrucción de Pausa (Pause) después del comando Attach y cada comando Type. El siguiente fragmento de guión le muestra cómo se puede anexar al intérprete de comandos del DOS y ejecutar la utilidad CHKDSK del DOS.

```
;Anexar a la ventana del DOS y hacer una pausa de un segundo.  
Attach "WINOA386_MS-DOS Prompt1", 1  
Pause 1  
;Escriba el nombre del comando y la pausa  
Type "CHKDSK {Return}  
Pause 1  
;Salir del DOS, pausa, y volver a Windows  
Type "Exit {Return}  
Pause 1
```

### Nota

ScriptTools puede usar una aplicación del DOS solamente si está en una ventana y no en modalidad de pantalla completa.

[Ver también](#)

[Comando Exec](#)

[Comando Attach](#)

[Grabación de una serie de acciones](#)

## Insertar un nombre de ventana

ScriptTools usa un nombre construido especialmente para identificar cada ventana de aplicación. Este nombre se construye con el nombre de la aplicación y un código numérico de ventana. A causa de que es difícil imaginarse o recordar esos nombres, debería recuperar nombres de ventanas con el comando Attach Name en el menú Crear de la ventana de edición de ScriptTools.

Para recuperar un nombre de ventana:

1. Asegúrese de que la aplicación para la que desea recuperar nombres de ventana se encuentra abierta en la pantalla.
2. Seleccione Nombre de anexo en el menú Crear.
3. En el cuadro de diálogo que se le muestra, seleccione el botón Identificar.  
El mouse adopta la forma de una lupa. Al desplazarse esa lupa por la pantalla, el cuadro de diálogo muestra el Nombre del anexo y otra información sobre la ventana que está debajo del cursor.
4. Cuando aparece el nombre de anexo de la ventana que desea, haga clic con el botón izquierdo del mouse para grabar la información en el cuadro de diálogo Nombre de anexo.
5. Seleccione Pegar, para copiar el nombre de anexo en la posición del cursor en el archivo .SCT activo en la ventana de edición.

---

Ver también

[Attach](#)

## Cambiar una definición de cuadro de diálogo

La manera más fácil de modificar una definición de cuadro de diálogo es copiar la definición en Hacer diálogo y editarla allí, donde podrá ver el resultado.

Para transferir una definición de diálogo a Hacer dialogo:

1. Copie en el portapapeles toda la sección del código de Definir diálogo.  
Comience con la instrucción Define Dialog y termine con EndDef.
2. Abra Hacer diálogo y pegue la definición desde el portapapeles a la ventana. Aparecerá una maqueta de cuadro de diálogo.
3. Edite el cuadro de diálogo de la misma forma que cuando lo creó.
4. Cuando haya hecho todas las modificaciones, seleccione Transferir en el menú Diálogo y traslade la definición de regreso a la ventana de edición.  
Si dejó la barra de selección sobre la antigua definición en el archivo en uso en la ventana de edición, la nueva definición la reemplaza.

Ver también

[Crear instrucciones de definición de cuadro de diálogo](#)

## Compilación y prueba de un guión

Se puede compilar su guión y probar el resultado en el cuadro de diálogo Compilador de ScriptTools, que se invoca con el comando Comprobar ejecución.

▶ Para compilar y probar un programa

1. Abra o cree el archivo de guión fuente en la ventana de edición.
2. Seleccione Comprobar ejecución en el menú Archivo.

Si no ha puesto nombre al archivo de guiones, ScriptTools le solicita que lo haga. Luego el compilador de ScriptTools compila el archivo de guión.

3. Si el compilador no encuentra errores, seleccione Ejecutar en el cuadro de diálogo Compilador de ScriptTools para ejecutar el programa.

Elija Cancelar para regresar a la ventana de edición sin ejecutar el programa.

4. Si el compilador encuentra errores, los escribe en un archivo con el nombre de su fichero de guión con la extensión .ERR y muestra un recuadro de advertencia. Seleccione Aceptar en esta casilla de advertencia y corrija los errores de código de fuente antes de elegir Comprobar ejecución de nuevo.

### Nota

Si prefiere emplear un editor diferente para crear los archivos de código de guión, puede invocar el compilador directamente. El compilador de ScriptTools es WNAC.EXE.

---

Ver también

[Depuración de su programa](#)

## Compilador de ScriptTools

Si creó o modificó su archivo .SCT con otro editor de textos, porque así lo prefiere o porque el archivo .SCT es mayor de 32K, lo puede compilar ejecutando directamente el compilador de ScriptTools.

Use las opciones del cuadro de diálogo para seleccionar el fichero que desea compilar y luego elija Aceptar.

### Opciones del cuadro de diálogo

Archivo a compilar Seleccione el archivo .SCT en la lista que aparece en el cuadro.

Directorios Se muestra el directorio activo. Seleccione el directorio que contiene el fichero .SCT que desea compilar.

Unidades Seleccione la unidad donde están localizados los ficheros .SCT.

## Depuración de su programa.

Los errores de compilación son informados durante la Comprobación de ejecución y se escriben en un fichero.

El compilador crea un archivo .ERR si encuentra errores en el código. Este fichero tiene el nombre de su archivo de guión y la extensión .ERR. Contiene el texto del mensaje de error y el número de la línea del archivo donde se produjo el error. El fichero .ERR se muestra automáticamente en la ventana de edición.

▶ Para leer las descripciones de los mensajes de error, seleccione Errores en la barra de botones de ayuda en la parte superior de la ventana de ayuda.

▶ Para desplazarse directamente a un número de línea específico, seleccione Ir a la línea: en el menú Editar o pulse Ctrl+G. El número de la línea se muestra en la barra de estado en la parte inferior de la ventana del código fuente.

Incluso si el compilador no informa errores, el programa de guión puede no comportarse como deseaba. Por ejemplo, su guión puede entrar en un ciclo interminable o tal vez olvidó un paso importante. Si eso ocurre, posiblemente sepa qué es lo que salió mal y puede revisar el archivo de guión.

▶ Para detener un guión desbocado , pulse Mayúsculas, Ctrl y Pausa en secuencia, no al mismo tiempo. La tecla de Pausa es usualmente F15.

## La ejecución de su guión

Antes de poder ejecutar un guión compilado, el tipo de fichero de su guión compilado debe ser asociado con el programa WNSRUN.EXE que ejecuta los ficheros de guiones.

El tipo de archivo por defecto para los guiones compilados es .RUN.

▶ Para asociar un archivo con una aplicación, utilice el comando Asociar en el menú Archivo del Administrador de programas, o Asociar accesorio de archivo en la carpeta Administración de archivos del escritorio de PC Tools.

Cuando haya asociado sus archivos de guiones con el programa de ejecución, puede ejecutar su guión de diferentes formas. Esta es una lista de cuatro:

1. Haga doble clic sobre el nombre del archivo en una lista, como las que se muestran en SmartFind o Administrador de archivos.
2. Cree un ítem de archivo y un icono para el guión, y láncelo desde el escritorio.
3. Use el Programador y configure un tiempo de ejecución.
4. Use el cuadro de diálogo Ejecutar en el Administrador de archivos del menú Archivo, o arrastre el nombre del fichero a Ejecutar accesorio de archivo y colóquelo allí.

## Mensajes de error del compilador

El compilador envía los siguientes mensajes de error:

### **Array "nombre" ya fue definido**

Utilizó el mismo nombre para definir más de una matriz o arreglo

### **Carácter erróneo en archivo fuente**

El compilador sólo interpreta caracteres estándar ASCII o ANSI.

### **Número hexadecimal erróneo**

Escribió un número hexadecimal con caracteres no válidos. Los caracteres hexadecimales válidos son: del 0 al 9 y de A a la F.

### **No se puede definir sucesos/diálogos en un bloque**

No se puede definir un suceso o diálogo dentro de una función o procedimiento, o en un bucle If, Select o While. Defina en forma global los sucesos y cuadros de diálogo, luego invóquelos a partir de construcciones de código condicionales.

### **No se puede definir función/procedimiento en un bloque**

No se puede definir una función o procedimiento dentro de una función o procedimiento, o un bucle If, Select o While. Defina la función o procedimiento fuera de la construcción condicional y luego invóquela.

### **Case fuera deSwitch**

Una instrucción de Case debe estar dentro de una construcción Switch...EndSwitch.

### **Default fuera de Switch**

La instrucción Default debe estar dentro de una construcción Switch...EndSwitch. Default define la conducta si no se ha evaluado una instrucción Case como verdadera.

### **Dialog 'nombre' ya fue definido**

Utilizó el mismo nombre para definir más de un cuadro de diálogo.

### **Else sin If**

Una instrucción Else debe estar después de una instrucción If.

### **EndCase sin Case**

Una instrucción Case debe estar antes de una instrucción EndCase, y ambas deben ser incluidas en una construcción Switch...EndSwitch.

### **EndFunction sin Función**

Una instrucción Function debe ir antes de una instrucción EndFunction.

### **EndIf sin If**

Una instrucción If debe ir antes de una instrucción Endif.

### **EndProc sin Proc**

Una instrucción Proc debe ir antes de una instrucción EndProc.

### **EndSwitch sin Switch**

Una instrucción Switch debe ir antes de una instrucción EndSwitch.

### **EndWhile sin While**

Una instrucción While debe ir antes de una instrucción EndWhile.

### **Event 'nombre' ya fue definido**

Utilizó el mismo nombre para definir más de un suceso.

### **Expresión Booleana esperada**

El compilador esperaba una expresión que podía ser evaluada como Verdadera o Falsa.

### **Etiqueta esperada**

El compilador esperaba una etiqueta.

**Nombre de variable local esperada**

El compilador esperaba el nombre de una variable local.

**Constante numérica esperada**

El compilador esperaba una constante numérica.

**Expresión numérica esperada**

El compilador esperaba una expresión numérica.

**Constante de cadena esperada**

El compilador esperaba una constante de cadena.

**Expresión de cadena esperada**

El compilador esperaba una expresión de cadena

**Expresión de cadena/numérica esperada**

El compilador esperaba una expresión de cadena o numérica.

**Definición de matriz esperada**

El compilador esperaba una definición de matriz (arreglo).

**Asignación esperada**

El compilador esperaba una instrucción de asignación como x=357

**EndFunction esperada**

Usted dejó fuera una instrucción EndFunction que cierra una definición de función.

**EndProc esperado**

Olvidó la instrucción EndProc que cierra una definición de procedimiento.

**Nombre de suceso esperado**

El compilador esperaba el nombre de un suceso definido.

**EndCase esperado**

Olvidó la instrucción EndCase, que cierra la serie de acciones en una instrucción de Case .

**EndIf esperado**

Olvidó la instrucción Endif que cierra una construcción con If.

**EndSwitch esperado**

Olvidó la instrucción EndSwitch que cierra una construcción con Switch.

**EndWait esperado**

Olvidó la instrucción EndWait que cierra una construcción con Wait.

**EndWhile esperado**

Olvidó la instrucción EndWhile que cierra una construcción While.

**Nombre de diálogo esperado**

El compilador esperaba un nombre de un cuadro de diálogo definido.

**Nombre de función esperado**

El compilador esperaba el nombre de una función definida.

**Nombre de procedimiento esperado**

El compilador esperaba el nombre de un procedimiento definido.

**Until esperado**

Olvidó la instrucción Until que especifica el final de una construcción Repeat.

**Expresión muy compleja**

La expresión es muy compleja para el compilador. El compilador no puede analizar instrucciones entre paréntesis que no han sido anidados suficientemente. Use

variables locales temporales para dividir la expresión.

**Sintaxis de expresión**

Verifique la sintaxis de la expresión para asegurarse de que es correcta.

**Función 'nombre' ya fue definida**

Usted ya usó este nombre para definir otra función.

**Identificador muy largo**

Un nombre de variable u otro identificador tiene más de 16 caracteres.

**Archivo de Include 'nombre' no encontrado**

El fichero especificado por una instrucción Include no se encontró en el directorio en uso o en la ruta de ScriptTools especificada en WIN.INI.

**Número entero fuera de la gama**

Se asignó un número fuera de la gama -32768 a 32767 fue asignado a una variable de número entero. Declare esta variable como Long.

**Expresión no válida**

Escribió una expresión no válida. Quizá deletreó mal el nombre.

**Identificador no válido**

Usó un nombre de variable que contiene caracteres no válidos o es una palabra reservada.

**Número entero no válido**

Escribió un número no válido. Sólo se permiten números enteros. Vea: "[Variables, Tipos de datos y operadores](#)".

**Etiqueta 'nombre' ya fue definida**

Usted usó este nombre para definir otra etiqueta.

**Variable local "nombre" ya fue definida**

Una variable local fue definida más de una vez dentro de un solo procedimiento.

**Constante numérica/cadena en lugar erróneo**

El compilador encontró una constante numérica o de cadena en un lugar erróneo.

**Falta segunda comilla**

Se le olvidó una segunda comilla.

**Operador omitido en una expresión**

Olvidó uno de los operadores en una expresión.

**Sin espacio de símbolo**

El compilador se quedó sin espacio interno de símbolo. Divida su guión en ficheros más pequeños y use [Run](#) o [Chain](#) para invocarlos.

**Procedimiento "nombre" ya fue definido**

Ya usó este nombre para definir otro procedimiento.

**Fin de archivo inesperado**

El archivo fuente terminó sin una definición o expresión de cierre.

**Until sin Repeat**

Una instrucción Until debe ser parte de una construcción Repeat.

**Tipo de operador erróneo en la expresión**

Ha usado un operador erróneo en la expresión. Consulte "[Nombres de variable, Tipos de datos y Operadores](#)".

## **Funciones de idioma en orden alfabético**

Esta lista contiene todos los comandos, funciones y variables predefinidas de ScriptTools. Haga clic sobre un nombre para saltar a ver su definición, descripción y un ejemplo de su uso.

Los comandos, funciones y variables se agrupan de acuerdo con su categoría funcional en [Instrucciones de idioma de ScriptTools](#). Una lista de las funciones sólo está contenida en las [Funciones del sistema](#). Las variables predefinidas aparecen en la lista de [Variables del sistema](#).

### **A**

[\\_ActiveWindow](#)

[ActiveWindow\\$\(\)](#)

[AmPm\\$\(\)](#)

[Array](#)

[Attach](#)

[\\_AttachMouseX, \\_AttachMouseY](#)

[\\_AttachWindow](#)

### **B**

[Beep](#)

[Button](#)

### **C**

[Call](#)

[Cancel](#)

[CapsLock On/Off](#)

[Capture\\$\(\)](#)

[Chain](#)

[ChangeDir](#)

[CheckBox](#)

[Chr\\$\(\)](#)

[ComboBox](#)

[ComboText](#)

[CommandLine\\$\(\)](#)

[Copy](#)

[Create](#)

### **D**

[Date\\$\(\)](#)

[\\_DateMode](#)

[\\_DayMode](#)

[Day\\$\(\)](#)

[DDEexecute\(\)](#)

[DDEInitiate\(\)](#)

[DDEPoke\(\)](#)

[DDERequest\\$\(\)](#)

[DDETerminate\(\)](#)

[Define Dialog](#)

[Define Event](#)

[Delete](#)

[Dialog](#)

[DLLCall\(\)](#)

## **E**

EditText  
\_EOF  
Exec  
Exit  
ExitWindows

## **F**

FileExists()  
Fill  
Focus\$()  
Focus()  
FocusWindow\$()  
Function

## **G**

GetEnvironment\$()  
GetText\$()  
GoTo

## **H**

Hour()

## **I**

If...Then...Else...EndIf  
Include  
InStr()

## **K**

Keyboard On/Off

## **L**

Left\$()  
Length()  
Listbox  
Long  
LowerCase\$()

## **M**

Macro  
MakeDir()  
Maximized()  
Maximize  
MenuDelete  
MenuSelect  
MenuSet  
MessageBox()  
Minimize  
Minimized()  
Mins()  
Month\$()  
Mouse On/Off

MouseClicked  
MouseMove  
MouseWindow\$()  
\_MouseX, \_MouseY  
Move

## **N**

NetAttach()  
NetConnect()  
NetDetach()  
NetDisconnect()  
NetLoaded()  
NetLogin()  
NetLogout()  
NetSendMessage()  
NetUserName\$()  
NumLock On/Off

## **P**

PadString\$()  
\_Path\$  
Pause  
\_PauseMode  
Permit  
Prevent  
Procedure  
PromptBox()

## **R**

RadioButton  
Read  
ReadFile\$  
ReadIni(), ReadIni\$()  
\_ReadPos  
Rem  
RemoveDir()  
Remove  
Rename  
Repeat  
Restore  
Resume  
Right\$()  
Run  
Running()

## **S**

ScrollBar  
ScrollLock On/Off  
Secs()  
SetFocus  
Size

Stop

SubStr\$()

Suspend

Switch

SysMenuSelect

**T**

TextPanel

Time\$()

Time()

TopInstance()

TopWindow\$()

Type

**U**

UpperCase\$()

**V**

Version\$()

Version()

**W**

Wait

Weekday\$()

Weekday()

Whenever

While

WindowExists()

Write

\_WriteFile\$

WriteIni(), WriteIni\$()

\_WritePos

**Y**

Year()

## Información rápida de ayuda

Esta es una breve guía de consulta para el uso de la ayuda de PC Tools para Windows.

### Símbolos en la página

▶ Este símbolo marca procedimientos.

Icono de comando Haga clic sobre el icono que aparece junto al título de la página de ayuda de un comando o cuadro de diálogo para visualizar otros comandos en el menú.

### Botones de ayuda

Use los botones que aparecen en la parte superior de la ventana de ayuda para visualizar funciones especiales de ayuda.

Contenido Vuelve a la página de Contenido para este archivo de ayuda.

Buscar Busca temas con contraseña, como usar un índice.

Retroceder Vuelve a la última página de ayuda que usted visualizó. Se puede volver atrás haciendo clic sobre Retroceder o pulsando B.

Resumen Muestra una lista de temas de ayuda que usted leyó más recientemente. Haga doble clic sobre un tema para volver a leerlo.

Glosario Muestra un glosario de términos utilizados en los archivos de ayuda de PC Tools para Windows.

Métodos cortos Muestra teclas de método corto y acciones de mouse.

### Otras utilidades...

Imprimir páginas de ayuda Seleccione Imprimir tema en el menú Archivo de ayuda.

Tomar notas Seleccione Anotar en el menú Editor de ayuda.

Copiar un tema Seleccione Copiar en el menú Editor de ayuda.

**Marcar un tema de ayuda**      **Seleccione el menú Marca-texto.**

## Funciones de lenguaje, comandos y variables del sistema

La lista presentada en esta página enumera las funciones, comandos y variables del sistema y está dividida por categorías de función. Si desea consultar esta lista en orden alfabético, seleccione la opción Funciones de la barra de botones sita en el extremo superior de la ventana de ayuda.

Las funciones de lenguaje, comandos y variables de ScriptTools no distinguen entre mayúsculas y minúsculas.

En las descripciones, se utilizan los siguientes signos:

- [] Los corchetes rectangulares encierran parámetros opcionales.
- <> Los corchetes angulares encierran parámetros obligatorios.
- | La raya vertical separa opciones que se excluyen entre sí.

Muchas de las descripciones de los comandos contienen ejemplos muy pormenorizados que el usuario puede copiar en el portapapeles y pegarlos en el script.

### Control del flujo del programa

[Call](#)  
[Cancel](#)  
[Define Event](#)  
[Exit](#)  
[GoTo](#)  
[If...Then...Else...EndIf](#)  
[Repeat](#)  
[Resume](#)  
[Run](#)  
[Stop](#)  
[Switch](#)  
[Suspend](#)  
[Wait](#)  
[Whenever](#)  
[While](#)

### Manipulación de series

[Chr\\$\(\)](#)  
[Instr\(\)](#)  
[Left\\$\(\)](#)  
[Length\(\)](#)  
[Lowercase\\$\(\)](#)  
[PadString\\$\(\)](#)  
[Right\\$\(\)](#)  
[SubStr\\$\(\)](#)  
[Uppercase\\$\(\)](#)

### Uso de programas

[CommandLine\\$\(\)](#)  
[DLLCall\(\)](#)  
[Exec](#)  
[ExitWindows](#)  
[Function](#)  
[GetEnvironment\\$\(\)](#)  
[Procedure](#)  
[Running\(\)](#)

### Empleo de archivos y directorios

[ChangeDir](#)  
[Chain](#)  
[Copy](#)  
[Create](#)  
[Delete](#)  
[\\_EOF](#)

[FileExists\(\)](#)  
[Fill](#)  
[Include](#)  
[MakeDir\(\)](#)  
[\\_Path\\$](#)  
[Read](#)  
[ReadIni\(\), ReadIni\\$\(\)](#)  
[\\_ReadFile\\$](#)  
[\\_ReadPos](#)  
[RemoveDir\(\)](#)  
[Rename](#)  
[Write](#)  
[\\_WriteFile\\$](#)  
[WriteIni\(\), WriteIni\\$\(\)](#)  
[\\_WritePos](#)

#### Utilización de cuadros de diálogo

[Button](#)  
[CheckBox](#)  
[ComboBox](#)  
[ComboText](#)  
[Define Dialog](#)  
[Dialog](#)  
[EditText](#)  
[ListBox](#)  
[MessageBox\(\)](#)  
[PromptBox\(\)](#)  
[RadioButton](#)  
[Remove](#)  
[ScrollBar](#)  
[TextPanel](#)

#### Uso de funciones de intercambio dinámico de datos (DDE)

[DDEExecute\(\)](#)  
[DDEInitiate\(\)](#)  
[DDEPoke\(\)](#)  
[DDERequest\\$\(\)](#)  
[DDETerminate\(\)](#)

#### Utilización de variables

[Array](#)  
[Long](#)

#### Empleo de la fecha y la hora

[AmPm\\$\(\)](#)  
[\\_DateMode](#)  
[Date\\$\(\)](#)  
[\\_DayMode](#)  
[Day\\$\(\)](#)  
[Hour\(\)](#)  
[Mins\(\)](#)  
[Month\\$\(\)](#)  
[Secs\(\)](#)  
[Time\(\)](#)  
[Time\\$\(\)](#)  
[Weekday\(\)](#)  
[WeekDay\\$\(\)](#)  
[Year\(\)](#)

#### Control de ventanas y menús

[ActiveWindow\\$\(\)](#)  
[\\_ActiveWindow](#)  
[Attach](#)

AttachWindow  
Capture\$()  
Focus()  
Focus\$()  
FocusWindow\$()  
GetText\$()  
Maximize  
Maximized()  
MenuDelete  
MenuSelect  
MenuSet  
Minimize  
Minimized()  
MouseWindow\$()  
Move  
Restore  
SetFocus  
Size  
SysMenuSelect  
TopInstance()  
WindowExists()

#### Uso del teclado y del Mouse

AttachMouseX, AttachMouseY  
CapsLock On/Off  
Keyboard On/Off  
Macro  
Mouse On/Off  
MouseMove  
MouseClicked  
MouseX, MouseY  
NumLock On/Off  
Permit  
Prevent  
Scrollock On/Off  
Type

#### Manipulación de las funciones de red

NetAttach()  
NetConnect()  
NetDetach()  
NetDisconnect()  
NetLoaded()  
NetLogin()  
NetLogout()  
NetSendMessage()  
NetUserName\$()

#### General

Beep  
Pause  
PauseMode  
Rem  
Version()  
Version\$()

## Funciones

Se las distingue por llevar paréntesis al final del nombre: Nombre().

Las funciones aceptan los valores paramétricos enunciados en la sección de sintaxis del apartado en el que aparece explicada cada función concreta. Hay que introducir dichos valores entre paréntesis. Si el parámetro aparece entre comillas dobles, hay que escribir el valor también entre comillas. Sin embargo, si sustituye una variable de serie por un parámetro literal, tiene que omitir las comillas.

Las funciones normalmente dan un valor. Dicho valor puede asignarse a una variable creada por el usuario, como muestra el ejemplo siguiente:

```
a=WindowExists("NOTEPAD_Edit1", 1)
```

Si el nombre de una función contiene el símbolo del dolar (\$), como en Date\$(), lo que da es una serie de texto. La variable en la que se almacena el valor dado debe ser una variable de serie.

Es posible crear funciones con nombres propios, valiéndose para ello del comando Function.

## Comandos

Los comandos de ScriptTools se sirven de una macro interna del mismo para ejecutar una acción o conjunto de acciones concretas. Hay más de 100 comandos ScriptTools, tales como Run, Beep, If...Then...Else...EndIf y Permit.

Para poder funcionar, muchos de los comandos necesitan además opciones o parámetros que especifiquen el objeto de la acción. Puede consultar estas opciones en la descripción del comando correspondiente.

Los comandos no dan un valor como resultado. No puede aparecer a la derecha de un enunciado de asignación, línea de código que utiliza el símbolo de igual (=) para asignar un valor a una variable.

## **Variables del sistema**

El primer carácter del nombre de una variable del sistema es un carácter de subrayado (\_): `_Nombre` de la variable.

Use las variables del sistema para averiguar o especificar el estado de los diversos elementos del programa como, por ejemplo, dónde se encuentra activado el Mouse, cuál es la ventana activa o la posición de lectura del archivo que está usando.

### **Variables de sólo lectura**

Ciertas variables son de sólo lectura y almacenan un valor asignado por el programa, según el estado del mismo. Asigne una variable previamente creada para recuperar dicho valor o, simplemente, examine el contenido de la variable del sistema.

Ejemplo de variables de sólo lectura:

```
OldPath$=_Path$  
If _EOF=1 Then.....
```

### **Otras variables del sistema**

Ciertas variables sirven para especificar cómo desea que actúe el programa. No está permitido asignar un valor arbitrario a una variable del sistema. Utilice los valores enumerados en la descripción de la variable en cuestión.

Ejemplos de configuración de variables:

```
_DayMode=1  
_PauseMode=0
```

## Array

Declara matrices.

### Sintaxis

Array <definiciones de matriz> [,<definiciones de matriz>,.....]

### Descripción

Declara matrices de serie o números enteros. El tamaño máximo de una matriz es de 10.000 elementos. Las matrices pueden tener solamente una dimensión, es decir, no pueden estar anidadas. El primer elemento de una matriz tiene asignado el número 0 en lugar del 1.

### Ejemplo

Este ejemplo crea una matriz de números enteros con 10 elementos. Dichos elementos están numerados del 0 al 9.

```
Array Numbers[ 10 ]
```

Este otro ejemplo crea una matriz de serie con 25 elementos, numerados del 0 al 24.

```
Array Alphas$[ 25 ]
```

El [ejemplo pormenorizado](#) muestra una mezcla varia de definiciones de matriz.

---

Consulte también

[Variables](#)

[Ejemplos de variables y series](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

```
;Ejemplo de Array
;Define y utiliza matrices de serie y numéricas
;
;
;matriz numérica de 10 elementos
Array alpha[ 10 ]

;matriz de serie de 10 elementos
Array beta$[ 10 ]

alpha[] = 1
;asigna el valor "1" a todos los elementos de "alpha"

alpha[0] = 10
;asigna el valor "10" al primer elemento de "alpha"

alpha[5] = alpha[0]
;copia el valor de "alpha[0]" en "alpha[5]"

textpanel 1 alpha[5]
Pause 2
;muestra un panel de texto con el valor de "alpha[5]" y se detiene
provisionalmente

beta$[] = "Sin definir"
;inserta la serie "sin definir" en cada uno de los elementos de "beta"

beta$[0] = "El perro de San Roque no tiene rabo"
;Inserta la serie "El perro de San ... rabo" en el primer elemento
; de la matriz de serie "beta".
beta$[5] = beta$[0]
;copia el valor de "beta$[0]" en "beta$[5]"

textpanel 2 beta$[5]
;muestra en pantalla un panel de texto con el contenido de "beta$[5]".
;se detiene provisionalmente para mirar el valor mostrado
Pause 2
;
;Fin del ejemplo de Array
;
```

## Attach

Anexa una ventana de aplicación.

### Sintaxis

Attach <nombre de la ventana> [, <número de instancia>]

### Descripción

Especifica la ventana en la que trabaja el usuario. Al anexar una ventana, queda activada. La mayoría de los comandos de ScriptTools se refieren a la ventana especificada en el enunciado Attach más reciente.

Attach utiliza el número de instancia para manipular los duplicados de un mismo programa. Por

ejemplo, instancia 1 es la primera copia del programa e instancia 2 la segunda copia. Si no está especificado el número de instancia o si está configurado en 0, se usa la primera ventana que case con el parámetro <nombre de la ventana>.

### **Ejemplo**

El primer [ejemplo pormenorizado](#) es un script de prueba que enseña cómo colocar un anexo en varios tipos de ventanas.

El segundo [ejemplo pormenorizado](#) es un script de prueba que muestra cómo cerciorarse de la presencia de una única copia de una aplicación antes de colocarle un anexo.

---

Consulte también

[Descubrimiento de un nombre de ventana](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

#### **;Ejemplo uno de Attach**

```
;Coloca un anexo en los diversos tipos de ventana.
;
;Ejecuta la aplicación Bloc de notas.
;
;Escribe "hola a todos" en el Bloc de notas.
;
exec "Notepad.EXE"
; Ejecuta la aplicación BLOC DE NOTAS.

; Utiliza la ventana de Bloc de notas.
    Attach "Notepad_Notepad", 1

Wk_AboutNotepad = 11
MenuSelect Wk_AboutNotepad
;Selecciona el ítem de menú "Acerca de"
;Utiliza el cuadro de diálogo "Acerca de".
Attach "Notepad_Notepad", 1
Wk_OK = 1
Button Wk_OK SingleClick
;Hace clic en el botón "Aceptar"

; Usa la ventana de edición del Bloc de notas.
Attach "BLOC DE NOTAS_Edit1", 1
;Utilice el área principal de edición del Bloc de notas y escriba
"hola a todos"
Escriba "hola a todos"
;
;Fin del ejemplo de Attach
;
```



Haga clic con el botón para copiar la muestra en el Portapapeles.

#### **;Ejemplo dos de Attach**

```
; Este ejemplo sirve para comprobar si la instancia 2
; del Bloc de notas está ejecutándose y
; si bien se conecta a un Bloc de notas ya existente e
; inicia otra o bien inicia una primera instancia.
; Tras ejecutar este script 3 veces, se conecta siempre a
; la instancia 2 a menos que borre la misma.
;
; prueba para ejecutar Bloc de notas e instancia
a=WindowExists("Notepad_Notepad",2)
b=running("notepad.exe")
;
; haga dos operaciones distintas, según los valores de resultado de
WindowExists()
; y Running().
; si existe la instancia 2, utilícela
if a=1
    attach "NOTEPAD_notepad",2
```

```

    a$="instance 2"
else
; si existe una instancia cualquiera, utilícela e inicie la instancia
2.
    if b=1
        attach "NOTEPAD_notepad"
        a$="generic instance"
        exec "notepad"
    else
; si no existe ninguna instancia, inicie una.
        if b=0
            exec "notepad"
            attach "NOTEPAD_notepad"
            a$="instance 1"
        endif
    endif
endif
; anexe la ventana de editar del Bloc de notas y escriba en ella.
attach "NOTEPAD_Edit1"
escriba "He anexo al Bloc de notas, " + a$ + ". {Return}"

```

## Beep

Emite por medio del altavoz interno la señal sonora especificada.

### Sintaxis

Beep <número>

Beep <serie>

### Descripción

Beep genera un pitido breve, con la frecuencia especificada por el parámetro <número>. El límite inferior de <número> es 1 y el superior 84. Los números más altos generan tonos más agudos, utilizando el módulo 84 de <número>. Cuanto más alto el número, más alta la frecuencia.

Para generar una serie de tonos, especifique <serie>. Las notas válidas están representadas por las letras "ABCDEFG" (correspondientes a las notas la, si, do, re, mi, fa y sol, respectivamente). Si utiliza una letra no incluida en la lista, no tiene ningún efecto.

Los sonidos se emiten mediante una escala de semitonos. Para indicar notas sostenidas, añada el signo (#) a la letra de la nota musical. Para indicar bemoles, añádale el signo (&).

Para especificar la duración del sonido, inserte un número antes de la letra de la nota. La duración de la misma viene dada por un número entero comprendido entre 1 y 8. Cuanto más bajo el número, mayor la duración de la nota.

Para especificar el tono, inserte un número tras la letra de la nota y un símbolo opcional de sostenido o bemoles. La octava de la nota viene especificada por un dígito sencillo comprendido entre 1 y 8. Cuanto mayor el número, mayor la octava.

No hace falta dejar espacios, a menos que prefiera insertarlos para evitar la ambigüedad al utilizar números enteros para indicar tanto la octava como el compás.

### Ejemplo

El ejemplo dado a continuación interpreta la canción "Cumpleaños feliz".

```
Beep "4a 6a 4b 4a 4d 4c#"
```

Consulte el [ejemplo pormenorizado](#), donde encontrará más canciones.



Haga clic con el botón para copiar la muestra en el Portapapeles.

#### **;Ejemplo de Beep**

```
;Produce distintos tonos de frecuencia, una serie de notas y una
escala
;
beep 10
;baja frecuencia
beep 50
;frecuencia media
beep 70
;alta frecuencia
beep "ABCD"
;interpreta una secuencia de notas en orden ascendente
beep "1a1 2a2 3a3 4a4 5a5 6a6 7a7 8a8"
;cada nota tiene una duración distinta (de corta a larga).
;y una octava también diferente
;a continuación, toque la escala de sol mayor (G).
beep "1G2 2A 2B 2C 2D 2E 2F 2F# 2G2"
;
;
;Fin del ejemplo de Beep
;
```

## **Button**

Selecciona un botón de un cuadro de diálogo o mensaje.

### **Sintaxis**

```
Button <identificación> Singleclick
```

### **Descripción**

Selecciona un botón del cuadro de diálogo o mensaje de la aplicación anexada actualmente. Los botones seleccionados con mayor frecuencia son Aceptar y Cancelar.

### **Ejemplo**

Este ejemplo envía un solo clic al botón Aceptar de un cuadro de diálogo si el nombre de la variable es WK\_OK.

```
Button Wk_OK SingleClick
```

Consulte el [ejemplo pormenorizado](#) , que facilita un script con el contexto del comando Button.

Consulte también

[Dialog](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

#### **;Ejemplo de Button**

```
;Selecciona el botón "Aceptar" del cuadro de diálogo "Acerca de" del
Bloc de notas;
exec "NOTEPAD.EXE"
; Ejecuta la aplicación Bloc de notas.
; Utiliza la ventana del Bloc de notas.
Attach "NOTEPAD_Notepad - (Untitled)1", 1
;Selecciona el ítem de menú "Acerca de"
Wk_AboutNotepad = 11
MenuSelect Wk_AboutNotepad
;A continuación, use el cuadro de diálogo "Acerca de".
Attach "NOTEPAD_About Notepad", 1
Wk_OK = 1
Button Wk_OK SingleClick
;Haga clic en el botón "Aceptar" para que desaparezca el cuadro de
diálogo.
;
;Fin del ejemplo de Button
;
```

## **Call**

Invoca el procedimiento de ScriptTools definido.

### **Sintaxis**

Call <procedimiento>

### **Descripción**

Ejecuta, a modo de subrutina, el procedimiento especificado. No es posible pasar parámetros a un procedimiento. Una vez finalizado dicho procedimiento, toma control el comando inmediatamente posterior al enunciado de Call.

Es factible definir procedimientos en cualquier punto del script, excepto en el interior de un bloque de código como, por ejemplo, un comando If...Then...Endif. Si lo desea, puede definir variables locales en los procedimientos, además, los procedimientos pueden invocar otros procedimientos. No obstante, no anide nunca más de 10 procedimientos.

### **Ejemplo**

Este ejemplo ejecuta un procedimiento denominado ExecNotePad.

```
Call ExecNotePad
```

Consulte el [ejemplo pormenorizado](#) que presenta el script entero que invoca al Bloque de notas.

---

Consulte también

[Procedimiento](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

#### **;Ejemplo de Call**

;Invoca un procedimiento que ocasiona una interrupción de 1 segundo  
;y ejecuta, acto seguido, una copia de Bloc de notas  
;Tras la finalización del procedimiento de llamada, aparece un mensaje.

```
call ExecNotePad
;lleva a cabo la llamada del procedimiento
;Una vez acabado dicho procedimiento, la línea ejecutada a
continuación es la siguiente.
textpanel 1 "Misión cumplida"
pause 1 second
; salga en este punto para demostrar que el procedimiento no tiene
porqué formar parte
; del programa principal
exit
```

;Inicie la definición del procedimiento.

```
Procedure ExecNotePad
```

;Los comandos que se realizan dentro del procedimiento:

```
exec "NOTEPAD.EXE"
```

```
pause 1 second
```

```
EndProc
```

```
;
```

```
;
```

```
;Fin del ejemplo de Call
```

```
;
```

## **ChangeDir**

Establece el directorio de trabajo activo.

### **Sintaxis**

```
ChangeDir <ruta>
```

```
CD <ruta>
```

### **Descripción**

Cambia el directorio activo para adoptar el indicado por la ruta especificada. Use este comando tanto para cambiar de directorio dentro de un mismo disco como para cambiar de unidad. Las formas "ChangeDir" y "CD" son intercambiables.

### **Ejemplo**

Este ejemplo establece como directorio activo C:\Windows.

```
CD "c:\windows"
```

## Cancel

Desactiva un comando Whenever.

### Sintaxis

Cancel <nombre del suceso>

### Descripción

No verifica la existencia de un suceso especificado por un comando Whenever. El suceso como tal sigue definido, de manera que puede comprobarlo en otro momento o en un contexto distinto.

### Ejemplo

Consulte el [ejemplo pormenorizado](#) que presenta un mensaje al hacer clic con el botón derecho del Mouse y cancela a continuación el comando Whenever.

---

Consulte también

[Whenever](#)

[Suspend](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

```
;Ejemplo de Cancel
;Define un suceso, establece un comando Whenever y lo cancela a
continuación
;
;Define un suceso "Rmouse" al hacer clic con el botón derecho del
Mouse
;
;Establece un comando "whenever" en el suceso "Rmouse".
;En cuanto se detecta el suceso "Rmouse", cancela el comando
"whenever".
;
;Define el suceso con el botón del Mouse presionado
Define Mouse Event Rmouse
    Right button Down Throwaway
    In Any Window
EndDef
;El comando "suspend" hace que el script
;reaccione exclusivamente ante sucesos.
suspend
;
;
;Establece el suceso con el comando "whenever".
whenever Rmouse
    Textpanel 1 "Suceso de Mouse"
    Pause 1 second
    Cancel Rmouse
    exit
endwhen
;
;Fin del ejemplo de Cancel
;
```

## CapsLock

Activa y desactiva la tecla de bloqueo de mayúsculas.

### Sintaxis

```
CapsLock <On>
```

```
CapsLock <Off>
```

### Descripción

CapsLock determina si la tecla de bloqueo de mayúsculas está activada o no. De esta manera el usuario sabe en qué estado se encuentra el teclado.

### Ejemplo

El ejemplo siguiente activa la tecla Bloq Mayús, espera 10 segundos y la desactiva.

```
CapsLock On
```

```
Pause 10 seconds
```

```
CapsLock Off
```

Consulte también

[NumLock](#)

Scrollock

## Chain

Inicia la ejecución de otro script de ScriptTools.

### Sintaxis

Chain <script>

### Descripción

Ejecuta el <script> especificado y continua la ejecución de la línea que sigue al comando Chain, de modo que ambos script se lleven a cabo. Sin embargo, no se da otra relación entre ambos programas de script.

Utilice el comando Chain para añadir módulos comunes a los script.

### Ejemplo

Consulte el [ejemplo pormenorizado](#) , en el que puede aprender a iniciar un segundo script y volver al script de llamada.

---

Consulte también

[Run](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

#### **;Ejemplo de Chain**

```
;Divida este ejemplo en dos archivos y compílelos
;por separado.
;MAIN.SCT
;
;Se encadena a un segundo script ("agent2.sct").
;
;Abre un panel de texto para mostrar la recuperación del mando.
;
;Realiza el comando "chain".
Chain "agent2"
;
textpanel 1 "Terminado comando Chain"
pause 5 seconds
;
; Fin del ejemplo de script: MAIN.SCT

;AGENT2.SCT
;Abre un panel de texto para mostrar la ejecución.
;
;Se detiene durante 10 segundos
;
textpanel 1 "Script AGENT2.SCT en ejecución"
pause 10 seconds
;
;Fin del ejemplo de script: AGENT2.SCT
;
```

## **CheckBox**

Selecciona una casilla de verificación.

### **Sintaxis**

```
CheckBox <identificación de control> SingleClick
CheckBox <identificación de control> On
CheckBox <identificación de control> Off
```

### **Descripción**

Selecciona la casilla de verificación identificada mediante <identificación de control>. Si especifica el parámetro SingleClick, cambia el estado de la casilla de verificación. Este comando se da con mayor frecuencia en aquellos script aprendidos con el comando Create Recording Session.

Los parámetros On y Off establecen explícitamente el estado de la casilla de verificación en On (con la equis) u Off (sin la equis), activada y desactivada respectivamente.

Al insertar el comando CheckBox en el script, la forma más fácil de determinar la <identificación de control> es mediante el empleo de la utilidad Create Attach Name. Con Create Attach Name aparece la información de estado de la ventana seleccionada.

El comando CheckBox no es lo mismo que la palabra clave CheckBox de la definición de cuadro de diálogo, que crea una casilla de verificación.

### **Ejemplo**

Estos ejemplos muestran las diversas formas del comando CheckBox.

CheckBox 12 SingleClick  
CheckBox 12 On  
CheckBox 12 Off

---

Consulte también

Dialog

Creación de una casilla de verificación

## ComboBox

Selecciona una serie de un cuadro combinado perteneciente a una aplicación anexa.

### Sintaxis

ComboBox <identificación de control> <texto> <SingleClick | DoubleClick>

### Descripción

Selecciona texto del cuadro combinado de una aplicación anexa. Se le reconoce por la <identificación de control>. El ítem que se va a seleccionar es <texto>. Este comando se da con mayor frecuencia en aquellos script que se aprenden con el comando Create Recording Session.

El parámetro <SingleClick> resalta el ítem <texto>. El parámetro <DoubleClick> selecciona el ítem <texto>.

El comando ComboBox es distinto de la palabra clave ComboBox de definición de cuadro de diálogo, que crea un cuadro combinado.

### Ejemplo

Este ejemplo no es un script ejecutable. Muestra las dos formas del comando ComboBox.

```
ComboBox 1232 "autoexec.bat" DoubleClick
```

```
ComboBox 1152 "All Files(*.*)" SingleClick
```

---

Consulte también

[Dialog](#)

[ComboText](#)

[Creating a Combo Box](#)

## ComboText

Emplee ComboText para introducir texto en un cuadro combinado.

### Sintaxis

ComboText <identificación de control> <texto\$>

### Descripción

Introduce el texto en el cuadro combinado de una aplicación anexa. El parámetro <identificación de control> especifica el cuadro combinado. Puede introducir el texto bien en forma de serie literal como, por ejemplo, "AUTOEXEC.BAT" o bien en forma de variable de serie como, por ejemplo, Texto\$.

Al introducir el comando ComboText puede utilizar bien Create Attach Name a fin de identificar la <identificación de control> del cuadro combinado o bien la opción de menú Create Recording Session.

### Ejemplo

Este ejemplo selecciona el segundo cuadro combinado de un cuadro de diálogo denominado STANDARD e introduce en él la serie de texto AUTOEXEC.BAT.

```
ComboText STANDARD_ComboBox2 "autoexec.bat"
```

---

Consulte también

[Dialog](#)

[ComboBox](#)

[Creación de un cuadro combinado](#)

## Copy

Copia un archivo.

### Sintaxis

Copy <nombre del archivo origen> To <nombre del archivo destino>

### Descripción

Copia un archivo o conjunto de archivos que corresponden con la especificación de archivo, del <nombre del archivo origen> al <nombre del archivo destino>. Puede utilizar los caracteres comodines ("?" y "\*"). El comando Copy es capaz de copiar archivos de un directorio o disco físico a otro.

### Ejemplo

Copia un archivo en otro directorio:

```
Copy "autoexec.sav" To "c:\windows\autoexec.sav"
```

Copia en la unidad B todos los archivos con la extensión .EXE:

```
Copy "*.exe" To "b:\*.**"
```

---

Consulte también

[Delete](#)

[Rename](#)

[Create](#)

## Create

Crea un archivo nuevo.

### Sintaxis

Create <ruta y nombre del archivo>

### Descripción

Crea un archivo con la ruta y el nombre especificados. El directorio tiene que existir. Si el archivo ya existe, se elimina su contenido actual. Si no especifica la ruta, se usa la ruta activa de Windows.

Use la variable de sistema `_Path$` para averiguar el nombre de ruta actual de Windows.

### Ejemplo

Crea un archivo en el directorio raíz (unidad C:):

```
Create "c:\test.dat"
```

---

Consulte también

[\\_Path](#)

[Read](#)

[Write](#)

## **DDEExecute()**

Pasa un comando a la aplicación servidora DDE.

### **Sintaxis**

DDEExecute(<identificador de conversación>, <comando>)

### **Resultado**

Da 1 si es satisfactorio, 0 en caso contrario.

### **Descripción**

Utiliza el identificador de conversación dado como resultado de DDEInitiate() para pasar el comando a la aplicación servidora DDE.

### **Ejemplo**

El script de muestra DDESHELL.SCT contiene una demostración de las funciones DDE.

El fragmento de código siguiente, que añade una carpeta nueva a la aplicación shell Windows, está tomado de DDESHELL.SCT:

```
a=DDEExecute(handle, "[CreateGroup('+a$+')]")
```

---

Consulte también

[DDEInitiate\(\)](#)

## **DDEInitiate()**

Obtiene un identificador para la aplicación que desea utilizar de servidora.

### **Sintaxis**

DDEInitiate(<nombre de la aplicación>,<tema>)

### **Resultado**

Da el identificador y, en caso de no resultar satisfactorio, da 0.

### **Descripción**

Da como resultado un identificador de conversación para la aplicación y tema especificados. Se pasa dicho identificador a otras funciones DDE con objeto de manipular la conversación.

### **Ejemplo**

Este ejemplo obtiene el identificador de conversación del Administrador de programas de Windows.

```
handle=DDEInitiate("PROGMAN", "PROGMAN")
```

El script de muestra DDESHELL.SCT contiene una demostración de las funciones DDE.

---

Consulte también

[DDEExecute\(\)](#)

[DDEInitiate\(\)](#)

[DDEPoke\(\)](#)

[DDERequest\(\)](#)

[DDETerminate\(\)](#)

## **DDEPoke()**

Pasa un valor a la aplicación servidora.

### **Sintaxis**

DDEPoke(<identificador de conversación>,<ítem>, <valor nuevo>)

### **Resultado**

Da 1 si es satisfactorio y, en caso contrario, 0.

### **Descripción**

DDEPoke() pasa un valor a la aplicación servidora.

### **Ejemplo**

Este ejemplo coloca 100 en la fila 10 columna 5 de una hoja de cálculo Excel.

```
handle=DDEInitiate("Excel"."BUDGET93.XLS")  
a=DDEPoke(handle,"R10C5","100")
```

---

Consulte también

[DDEInitiate\(\)](#)

## **DDERequest\$()**

Consigue una lista de ítems válidos para los datos de conversación DDE de un ítem específico.

### **Sintaxis**

DDERequest\$(<identificador de conversación>,<ítem>)

### **Resultado**

Da los datos de la petición en forma de serie de texto o bien una serie en blanco si el comando no resulta satisfactorio.

### **Descripción**

Emplee DDERequest\$() de dos formas. Si pasa una serie nula en el parámetro <ítem>, DDERequest() da como resultado la lista de todos los ítems válidos de la conversación. Si pasa un ítem válido, obtiene los datos del mismo.

### **Ejemplo**

El ejemplo dado a continuación contiene extractos de DDESHELL.SCT.

```
;Para obtener una lista con todos los ítems válidos de la conversación
array item$[100]
;declare una matriz para obtener resultados
array list1$[100]
; Cree una serie nula
item$=Chr$(0)
;obtenga una lista de los ítems válidos de $
a$=DDERequest$(handle,item$)
; ....
;verifique el resultado de $....
;ordene item$[] en list1$[]....
;.....
;Para obtener los datos de un ítem válido que el usuario seleccione desde la lista almacenada en
list1$[]
a$=DDERequest$(handle,list1$[0])
```

Consulte el script de muestra DDESHELL.SCT para ver una demostración exhaustiva de las funciones DDE.

Consulte también

[DDEInitiate\(\)](#)

## **DDETerminate()**

Cierra una conversación DDE.

### **Sintaxis**

DDETerminate(<identificador conversación>)

### **Resultado**

Da 1 si es satisfactorio y 0 de lo contrario.

### **Descripción**

Utiliza el identificador de conversación obtenido con DDEInitiate() para terminar una conexión DDE. Con frecuencia se usa DDETerminate() para terminar una conversación si se da un error buscado por el usuario.

Al salir del programa de script, se cierran automáticamente todas las conversaciones DDE.

### **Ejemplo**

Este ejemplo termina una conversación DDE si se detecta un error.

```
If Shell_Name$= "Progman" then
    MsgBox("Error", "Hay dos programas shell en ejecución.", 0)
    DDETerminate(handle)
    Exit
else
    Call good_shell
EndIf
```

Consulte el script de muestra DDESHELL.SCT que contiene una demostración exhaustiva de las funciones DDE.

---

Consulte también

[DDEInitiate\(\)](#)

## Define

Define un cuadro de diálogo o un suceso para su uso en un script.

### Sintaxis

```
Define <tipo de objeto> [<tipo genérico>] name  
    <parámetros de definición>  
EndDef
```

### Descripción

Crea un cuadro de diálogo o un suceso. El parámetro <tipo de objeto> es el nombre del objeto definido, bien Dialog o uno de los sucesos, como Screen o Keyboard. Si define un suceso, puede incluir el parámetro opcional <tipo genérico> Event.

Es muy poco corriente tener que escribir un comando Define entero, puesto que es posible servirse de las utilidades Create Event o Create Dialog para generar definiciones automáticamente.

Emplee Create Event para definir sucesos cronológicos, de pantalla, teclado, Mouse, menú y ventana. Utilice Create Dialog para definir cuadros de diálogo.

Una vez creado un cuadro de diálogo o suceso, puede modificar el código mediante la edición del archivo de script.

### Ejemplos:

Muchos de los ejemplos pormenorizados de script cuentan con comandos Define. A continuación se enuncian unos cuantos:

[Ejemplo de Fill](#)

[Ejemplo de MenuSet](#)

[Ejemplo de Resume/Suspend](#)

[Ejemplo de Whenever](#)

---

Consulte también

[Wait](#)

[Whenever](#)

[Definiciones de cuadro de diálogo](#)

[Definiciones de suceso](#)

## Delete

Borra un archivo.

### Sintaxis

Delete "<nombre del archivo>"

### Descripción

Borra un archivo o grupo de archivos que casan con la especificación de archivo del parámetro <nombre del archivo> . Puede utilizar los caracteres comodines ( "?" y "\*" ). El comando Delete es capaz de borrar archivos almacenados en distintos directorios o unidades.

### Ejemplo

Borra un archivo del directorio raíz (unidad C:\).

```
Delete "c:\autoexec.old"
```

Borra todos los archivos ".BAK" del directorio WINDOWS.

```
Delete "c:\windows\*.bak"
```

---

Consulte también

[Copy](#)

[Rename](#)

## Dialog

Abre el cuadro de diálogo definido por el script.

### Sintaxis

Dialog <cuadro de diálogo> <control 1,.....,control n>

### Descripción

Muestra el cuadro de diálogo especificado por el parámetro <cuadro de diálogo>.

El resultado de las opciones del usuario y del texto introducido viene especificado por las variables especificados en los parámetros <control 1, ....control n>. Por ejemplo, en el ejemplo pormenorizado, si el usuario hace clic en el botón Aceptar, la variable "okb" es 1. Si el usuario no selecciona Aceptar, "okb" es 0. El resultado del ítem seleccionado en la lista aparece en el primer elemento de la matriz pasado a la lista, "list\$[0]."

### Ejemplo

Consulte el [ejemplo pormenorizado](#), en el que un script de muestra define un cuadro de diálogo con todos sus controles, le pasa todas las variables y, por último, lo muestra.

---

Consulte también

[Define Dialog](#)



Haga clic con el botón para copiar la muestra en el Portapapeles.

```
;Ejemplo de Dialog
;
;
;Define un cuadro de diálogo con todos sus controles.
;
;Pasa las variables al cuadro de diálogo.
;
;Presenta el diálogo.
;
;Define el diálogo
  Define Dialog AllDiags 150 116
  Title "All ScriptTools controls"
  TopLeft 75 75
  PushButton 13 80 27 23 "OK" Default
  PushButton 50 80 27 23 "Cancel"
  Text 5 5 "Text"
  EditControl 23 3 60 12
  CheckBox 7 22 "Check Box"
  RadioButton 12 50 "Button1"
  RadioButton 12 62 "Button2"
  ListBox 97 5 32 30 Sorted
  ComboBox 87 42 32 32 Sorted
  Icon 120 80 Information
  GroupBox 5 37 57 37 "Group Box"
EndDef
;
;Define las matrices de la lista y del cuadro combinado
Array list$[10], combo$[10]
;
;Rellena las matrices de números
i = 0
repeat
  combo$[i] = i
  list$[i] = i
  i = i + 1
until i>9
;
;Inicia el control de edición
a$ = "No text"
;
;Establece como valor predeterminado la casilla de verificación "On"
check = 1
;
;Resalta "Button 1"
rb1 = 1
;
;Muestra el diálogo.
dialog AllDiags okb, canb, a$, check, rb1, rb2, list$[], combo$[]
;
;
;
;Fin del ejemplo de Dialog
;
```



## EditText

Introduce texto en un control de edición.

### Sintaxis

EditText <identificación de control> <texto>

### Descripción

Coloca <texto> en el control de edición de una aplicación anexa, identificada por <identificación de control>.

Determina la <identificación de control> por medio de la utilidad Create Attach Name.

### Ejemplo

Consulte el [ejemplo pormenorizado](#), en el que un script de muestra, que anexa el Administrador de programas, selecciona el comando Run del menú Archivo y utiliza EditText para introducir el Bloc de notas como el programa que ha de ejecutarse.

---

Consulte también

[Dialog](#)

[Creación de un cuadro de edición de texto](#) tras seleccionar Make Dialog.



Haga clic con el botón para copiar la muestra en el Portapapeles.

```
;Ejemplo de EditText
;Ejecuta la aplicación Bloc de notas desde el Administrador de
programas.
;
;Usa la ventana principal del Administrador de programas.
Attach "PROGMAN_Program Manager1", 1
;
;Selecciona "Ejecutar" desde el "menú Archivo".
Wk_Run = 107
MenuSelect Wk_Run
;
;Usa el cuadro de diálogo Ejecutar.
Attach "PROGMAN_Run", 1
;
;Inserta "notepad" (Bloc de notas) en el control de edición
Wk_notepad = 119
EditText Wk_notepad "notepad"
;
;Hace clic en el botón "Aceptar".
Wk_OK = 1
Button Wk_OK SingleClick
;Fin del ejemplo de EditText
;
```

## Exec

Inicia una aplicación Windows o DOS.

### Sintaxis

```
Exec "<nombre de la aplicación> [<línea de comando>]"
```

### Descripción

Ejecuta el programa especificado. No hace falta facilitar la extensión de archivo. Si la aplicación se encuentra en el directorio Windows o en la ruta del DOS, no se exige facilitar la ruta. Es posible pasar a la aplicación parámetros de línea de comando legítimos.

Tenga en cuenta que no puede usar la utilidad Create Recording Session para aprender pulsaciones de teclado en una aplicación Dos una vez iniciada desde ScriptTools. Al salir del programa DOS, se le devuelve automáticamente a Windows.

### Ejemplo

Este ejemplo ejecuta la aplicación Bloc de notas y carga "WIN.INI".

```
Exec "NOTEPAD WIN.INI"
```

Consulte también

[Aplicaciones DOS](#)

## Exit

Termina el script y devuelve el mando al usuario.

### Sintaxis

Exit

### Descripción

Interrumpe el script y cierra todos los sucesos, diálogos, paneles de texto y macros creados por el script.

Si se ha ejecutado el script por medio del comando Run, se devuelve el mando al script de llamada. En todos los casos la aplicación de primer plano recupera el control.

### Ejemplo

Consulte el [ejemplo pormenorizado](#) para ver un breve script de prueba.



Haga clic con el botón para copiar la muestra en el Portapapeles.

```
;Ejemplo de Exit
;Muestra un mensaje en pantalla.
;Se detiene durante 5 segundos
;Sale
;
;Muestra un mensaje
textpanel 1 "Mensaje breve."
;
;Espera durante 5 segundos
pause 5 seconds
;
;Termina el script.
Exit
;
;Este enunciado no se ejecuta NUNCA.
beep 10
;
;Fin del ejemplo de Exit
```

## ExitWindows

Termina la sesión Windows.

### Sintaxis

ExitWindows [<valor de salida>]

### Descripción

Intenta dar por finalizada la sesión Windows. El nivel de error DOS está establecido por <valor de salida> si se especifica.

Los valores legítimos de salida son los siguientes:

67 Rearranca el sistema.

68 Reinicia Windows

0 Sale a DOS

ExitWindows no funciona si una aplicación abierta se niega a finalizar. Por ejemplo, si el usuario selecciona Cancelar como respuesta a un mensaje de salida como "¿Desea cambiar los cambios actuales?" del Bloc de notas o si hay un subproceso DOS activo, Windows no consigue salir.

Si hay una aplicación activa, aparece un cuadro de mensaje. Una vez ha seleccionado el usuario el botón Aceptar, el script sigue desde el punto en que se ha dado el comando ExitWindows.

### Nota

Con ciertos ordenadores, si usa el comando ExitWindows para reorganizar el sistema, cabe la posibilidad de que sufra una detención inesperada. Si se da el caso, pulse el botón de restaurar o apague el ordenador y vuelva a encenderlo transcurridos unos breves instantes. El método de pulsar las teclas Ctrl + Alt + Supr no suele funcionar en estos casos.

### Ejemplo

Las siguientes dos líneas de código intentan salir y reiniciar Windows y, acto seguido, terminar el script si el comando ExitWindows no da resultado.

```
ExitWindows 68
Exit
```

---

Consulte también

Exit

Stop

## Fill

Rellena una matriz con una lista de archivos.

### Sintaxis

Fill <nombre de la matriz> With <especificación del archivo>

### Descripción

La matriz especificada se rellena con todos los archivos que coinciden con la especificación del archivo. Si el número de archivos que coinciden es demasiado grande para la matriz, no se utilizan los archivos adicionales. Si el nombre de la matriz no es válido, no se ejecuta el comando.

Use Fill para crear una lista de nombres de archivo antes de presentarla en un control de cuadro de diálogo, como una lista.

### Ejemplo

El [ejemplo pormenorizado](#) muestra cómo rellenar una matriz y cómo presentarla en un cuadro de diálogo.

```

▶
;Ejemplo de Fill
;
;Rellena una matriz con archivos .SCT.
;
;Presenta un cuadro de diálogo con la matriz.
;
;Muestra el archivo seleccionado.
;
;Define una matriz para los nombres de archivo.
array files${50}
;
;Rellena la matriz con los archivos .SCT
fill files${} with "*.sct"
;
;Define el diálogo a fin de presentar la matriz.
  Define Dialog Files 105 80
  Title "Mostrar comando Fill"
  TopLeft 75 75
  PushButton 65 25 23 13 "OK" Default
  ListBox 5 5 55 55 Sorted
EndDef
;
;Muestra el diálogo con la matriz
dialog files okb, files${}
;
;Muestra el archivo seleccionado
textpanel 1 "El archivo seleccionado es : " + files${0}
;
;Deja transcurrir el tiempo necesario para ver el resultado
pause 3 seconds
;
;Fin del ejemplo de Fill .
;

```

## Function

Define una función de ScriptTools.

### Sintaxis

```

Function <nombre de la función> ( [<argumento>,<argumento>.....] )
  [Local <variable local > [,<variable local>.....] ]
  [Return <expresión>]
EndFunction

```

### Descripción

Utiliza las funciones definidas por Function aproximadamente de la misma manera que se usan las funciones en un lenguaje como el C.

Es posible pasar argumentos a una función definida y conseguir un valor de resultado. Ésta es la principal diferencia entre una función y un procedimiento de ScriptTools. Los argumentos de la función definida se ejecutan siempre que aparece en una expresión el nombre de la función. Si la función definida contiene un enunciado Return, la expresión se evalúa y el resultado se da como el valor actual de la función. Puede bien almacenar dicho valor en una variable o bien analizarlo en el script. Sin embargo, una función no tiene porqué definir un valor de resultado.

El enunciado Local define variables locales de la función. Sírvese de comandos para separar los

nombres de las variables.

Los parámetros utilizados como argumentos en las líneas de enunciado Function o Local pueden ser números enteros cortos o series entre comillas. No pueden ser matrices ni números enteros largos. No puede usarse una variable Local fuera de su función definidora. Los números enteros largos y matrices pueden manipularse dentro de una función si se los ha declarado de forma global.

No tiene porqué asignarse a una variable el valor de resultado de una función. Cuando no se almacena el valor de resultado, la función se comporta como un procedimiento, pero se le permiten argumentos.

### **Ejemplos**

Se presentan dos ejemplos pormenorizados.

Empleo de una función en una expresión

Empleo de una función fuera de una expresión

```
▶
;Ejemplo 1 de Function
;
;Define una función para añadir dos números.
;
;Invoca la función "Add2" para llevar a cabo la adición.
a=Add2( 4, 5 )
;
;Define la función Add2().
Function Add2( a, b )
return a+b
EndFunction
;
;
;Fin del ejemplo 1 de Function
```

```
▶
;Ejemplo 2 de Function2
;
;Ejecuta el programa según el número.
;
;Invoca la función con el parámetro (fijado en 1 en el ejemplo)
RunProgram( 1 )
;
;Define la función "RunProgram()"
Function RunProgram( ProgramID )
if ProgramID = 1 then
    exec "NOTEPAD win.ini"
else
    exec "NOTEPAD 3270.txt"
endif
EndFunction
;
;Fin del ejemplo 2 de Function
;
```

## GoTo, Go

Lleva hasta una etiqueta del script.

### Sintaxis

GoTo <etiqueta>

Go <etiqueta>

### Descripción

Pasa a la primera instrucción que sigue a la etiqueta especificada.

Una etiqueta es una serie de caracteres alfanuméricos que terminan en dos puntos (:), como StopHere:. No se usan los dos puntos al utilizar el nombre de la etiqueta a modo de argumento del comando GoTo.

GoTo no distingue entre mayúsculas y minúsculas.

### Ejemplo

Consulte el [ejemplo pormenorizado](#) , que expone cómo utilizar GoTo de forma condicional.

```

▶
;Ejemplo de GoTo
;
;Pasa por un script saltándose ciertas instrucciones.
;
;Muestra un mensaje.
textpanel 1 "Empezar aquí"
;Se detiene durante un segundo
Pause 1 second
;
;Ejecuta un comando GoTo para impedir el comando de salida.
GoTo Skip
;
;Este comando no debe ejecutarse nunca.
exit
;
;Etiqueta a la que hay que saltar (rematada por dos puntos).
Skip:
;Muestra un mensaje
textpanel 1 "Fin del script"
pause 2 seconds
;
;Fin del ejemplo de GoTo
;

```

## If...Else...EndIf

Comprueba las condiciones y puede especificar acciones alternativas.

### Sintaxis

```

If <condición> [Then]
<instrucciones>
[Else <instrucción>]
EndIf

```

### Descripción

Ejecuta un conjunto de comandos cuando se evalúa un enunciado como verdadero.

Si la condición inicial es falsa pero las condiciones Else son opcionales, puede especificar otro conjunto de comandos que desea ejecutar. Si omite el enunciado Else y la condición es falsa, el script salta al siguiente comando del mismo.

No hace falta especificar la palabra clave Then, pero facilita la lectura del código.

### Ejemplo

El [ejemplo pormenorizado](#) muestra cómo obtener un número aleatorio y una señal sonora si éste es inferior a 100.

```

▶
;Ejemplo de If...Else...EndIf
;
;Obtiene un número al azar.
;
;Si el número es inferior a 100, emite una señal sonora.
;
;Si el número es superior a 100, aparece un panel.
;
;Obtiene el número al azar
a=random()
;
;Ejecuta If..EndIf.
If a < 100 Then
    beep 50
Else
    textpanel 1 "El número es superior a 100"
EndIf
;
;Deja transcurrir el tiempo para leer el panel de texto.
pause 1 second
;
;
;Fin del ejemplo de If...Else...EndIf
;

```

## Include

Incluye otros archivos de origen ScriptTools al realizar una compilación.

### Sintaxis

```
Include <archivo .SCT>
```

### Descripción

Ordena al compilador de ScriptTools que compile el archivo especificado, como parte del programa principal. El archivo .SCT puede constar de cualquier enunciado válido ScriptTools.

El compilador primero busca los archivos sitios en el directorio actual y, a continuación, en el directorio predeterminado de ScriptTools. Utiliza el primer archivo que encuentra.

Una de las funciones más corrientes del comando Include es la de utilizar definiciones de diálogo comunes en más de un script.

El tamaño máximo de un único archivo .SCT es de 32 K. El comando Include sirve también para crear script más grandes.

### Ejemplo

Este ejemplo presenta cómo incluir un archivo denominado DIALOGS.SCT.

```
Include "dialogs.sct"
```

## Keyboard

Activa o desactiva el teclado.

### Sintaxis

Keyboard On

Keyboard Off

### Descripción

Activa y desactiva la entrada de datos por medio del teclado. El sistema no acepta la introducción de datos por medio del teclado mientras éste se halla desactivado.

Solo la introducción de datos por parte del usuario queda afectada. El script sigue funcionando y es posible utilizar el comando Type para introducir caracteres en un programa de aplicación.

**Cuidado:** Utilice Keyboard Off sólo cuando sea estrictamente necesario, ya que quedan desactivadas incluso las secuencias especiales de teclado de método abreviado de Windows.

### Ejemplo

El [ejemplo pormenorizado](#) muestra el empleo de Keyboard Off.

---

Consulte también

[Mouse On/Off](#)

```
▶  
;Ejemplo de Keyboard  
;  
;Desactiva el teclado durante 30 segundos y, a continuación, vuelve a  
activarlo.  
;  
;Desactiva el teclado.  
Keyboard Off  
;Espera durante 30 segundos (pruebe a escribir en el Bloc de notas).  
Pause 30 seconds  
;Vuelve a activar el teclado.  
Keyboard On  
;  
;  
;Fin del ejemplo de Keyboard.  
;
```

## ListBox

Selecciona una serie de una lista de una aplicación anexa.

### Sintaxis

```
ListBox <identificación de control> <texto> <SingleClick>  
ListBox <identificación de control> <texto> <DoubleClick>
```

### Descripción

Selecciona el texto de la lista de una aplicación anexa. El parámetro <identificación de control> especifica la lista. La serie del parámetro <texto> especifica el ítem que ha de seleccionarse.

En la mayoría de las listas, con la opción <SingleClick> de un solo clic, simplemente se resalta el ítem. Con la opción <DoubleClick> de doble clic, se resalta y selecciona el ítem.

Use Create Attach Name para identificar <identificación de control>.

El comando ListBox no es lo mismo que la palabra clave ListBox, empleada en la definición de un cuadro de diálogo, que crea una lista.

### Ejemplo

El [ejemplo pormenorizado](#) expone cómo anexar el Bloc de notas y seleccionar un archivo específico desde la lista Abrir del menú Archivo.

---

Consulte también

[Creación de una lista](#)

[Dialog](#)

```

▶
;Ejemplo de ListBox
;
;Ejecuta la aplicación Bloc de notas.
;
;Selecciona "Abrir" del menú Archivo.
;
;Usa ListBox para seleccionar el archivo "setup.txt"
;

;Ejecuta el Bloc de notas
exec "Notepad.exe"
;Escoge el comando Abrir del menú Archivo
Attach "NOTEPAD_Notepad", 1

Wk_Open = 10
MenuSelect Wk_Open
;Anexa al diálogo "Abrir".
Attach "NOTEPAD_Open", 1
Wk_txt = 1152
EditText Wk_txt "*.txt"
;Resalta el archivo "setup.txt".
Wk_SetUPTXT = 1120
ListBox Wk_SetUPTXT "SETUP.TXT" SingleClick
;Selecciona el archivo "setup.txt".
Wk_SetUPTXT = 1120
ListBox Wk_SetUPTXT "SETUP.TXT" DoubleClick
;
; Fin del ejemplo de ListBox
;

```

## Long

Declara una variable como un número entero largo.

### Sintaxis

Long <a>[, <b>.....]

### Descripción

Define una variable como un número entero largo. Hay que declarar todas las variables largas antes de utilizarlas. Una variable larga puede tener aquellos valores que son demasiado largos para los números enteros cortos, cuyos límites son -32.768 a +32.767. Los límites de valores largos son de -2.147.483.648 a +2.147.483.647.

Al introducir un número entero largo como valor literal, escriba el número con una L de sufijo, a=334578L, por ejemplo.

Si no está seguro del número que tiene que manipular, es mejor declararlo como variable Long.

Al comparar un número entero Long con uno corto, siga las dos normas siguientes:

- ▶ En un enunciado, el número entero Long tiene que estar a la izquierda del signo de igual.
- ▶ En una construcción Switch, el número entero Long tiene que estar en el enunciado Switch en lugar de en el enunciado Case.

### Ejemplo

Puede declarar una serie de variables Long en una línea.

```
Long anum, bnum, cnum, dnum
```

El [ejemplo pormenorizado](#) muestra cómo definir y utilizar una variable larga.

---

Consulte también  
[Variables](#)

```

▶
;Ejemplo de Long
;
;Usa una variable larga para llevar a cabo un bucle 100.000 veces.
;
;Declara la variable como Long (antes de usarla).
Long loopcount
;Ejecuta un bucle 100.000 veces.
loopcount = 0
repeat
    loopcount = loopcount + 1
;las operaciones que realiza durante la ejecución del bucle aparecen
aquí.....
;
until loopcount > 100000
;El bucle ha terminado.
textpanel 1 "La cuenta actual de bucles es de " + loopcount
;Deja transcurrir el tiempo para ver el panel de texto.
pause 1 second
;
;
;Fin del ejemplo de Long
;

```

## Macro

Define macros de teclado.

### Sintaxis

Macro <descripción clave> To <acción clave>

### Descripción

Configura macros de teclado. Siempre que se da la pulsación de teclado especificada en <descripción clave>, se lleva a cabo en la aplicación activa la acción especificada en <acción clave>.

El parámetro <descripción clave> puede constar de una sola tecla más las teclas modificadoras Alt y Ctrl. Deje un espacio tras el nombre de la tecla modificadora y encierre la expresión y nombre de la tecla en el tipo de corchetes utilizados en el ejemplo siguiente.

Un programa de script puede contener un gran número de macros.

### Ejemplo

Este breve ejemplo muestra cómo hacer que Alt + Mayúsculas + F1 impriman "Hola a todos" en una ventana.

Macro "{Alt {Shift {F1}}}" To "Hola a todos"

El [ejemplo pormenorizado](#) configura y comprueba tres teclas macro.

### **▶** ;Ejemplo de Macro

```
;Macros que redibujan el mapa de las teclas "a", "b" y "c".  
;  
;Ejecuta "Bloc de notas" para comprobar las macros.  
;Permite la prueba hasta Ctrl+x  
;Quita el script para restaurar las teclas.  
;  
;Ejecuta el Bloc de notas.  
exec "notepad"  
;define el suceso de teclado para salir  
Define Keyboard Event Quit  
    "{Ctrl x}"  
    In Any Window  
EndDef  
;  
;añade un par de paneles de texto con instrucciones  
textpanel 1 "Escriba a, b o c para ver las macros."  
pause 2 seconds  
remove 1  
textpanel 2 "Pulse Ctrl + x para restaurar el teclado normal."  
pause 2 seconds  
;  
;Configura las macros.  
macro "a" to "Esta es la tecla A {Retorno}"  
macro "b" to "Esta es la tecla B {Retorno}"  
macro "c" to "Esta es la tecla C {Retorno}"  
;Especifica la condición final  
Wait until Quit  
    Exit  
;Fin del ejemplo de Macro  
;
```

### **Maximize**

Agranda una ventana de manera que ocupa toda la pantalla.

#### **Sintaxis**

Maximize [<nombre de la ventana> [,<número de instancia>]]

#### **Descripción**

Hace que la ventana especificada ocupe toda la pantalla. Si no especifica el parámetro <nombre de la ventana>, se maximiza la que se encuentre activa en ese momento.

Si especifica el <nombre de la ventana>, puede especificar también el <número de instancia> para maximizar una instancia específica de la ventana en cuestión. El valor predeterminado es que se maximice la primera ventana cuyo nombre coincida con el parámetro nombre de la ventana.

#### **Ejemplo**

El primer ejemplo expone cómo maximizar la ventana actual. El segundo muestra cómo maximizar la ventana PC Tools para Windows del Administrador de archivos.

```
Maximize  
Maximize "WNFM_File Manager1"
```

## MenuDelete

Borra un ítem de un menú de aplicación.

### Sintaxis

MenuDelete <ítem del menú> On <menú\$> In <nombre de la ventana\$> [, <instancia>]

### Descripción

Elimina un ítem de un menú de aplicación de manera que el usuario no pueda seleccionarlo. Esta operación sólo funciona con el caso específico del programa activo.

El parámetro <ítem del menú> puede ser una serie o un valor numérico.

Si especifica una serie, se compara la longitud especificada en el <ítem del menú> con los ítems del menú. Basta con escribir el suficiente número de caracteres que lo distinguen de los demás ítems del menú. Un carácter subrayado de un nombre de menú o ítem va precedido del signo &. El valor que casa con <ítem del menú> se borra.

Si especifica un número, se borra el ítem de menú en dicha posición (contando desde 0). De esta manera puede borrar separadores horizontales de barras, que son ítems numerados de menú.

Cuando se ha borrado un ítem de menú, los demás ítems se vuelven a numerar. Es decir, para borrar los ítems 2 y 3 de un menú, primero hay que borrar el ítem 2, lo cual hace que el antiguo ítem número 3 se convierte en el ítem número 2 y, a continuación, volver a borrar el ítem 2.

Si deja el parámetro <ítem del menú> en blanco, se borra el menú entero.

### Ejemplo

Los ejemplos presentados a continuación utilizan el Bloc de notas como programa y muestran cómo borrar ítems del menú Archivo.

Borra el comando Imprimir por su nombre:

```
MenuDelete "&Imprimir" On "&Archivo" In "Bloc de notas"
```

Borra el comando Nuevo por posición

```
MenuDelete 0 On "&Archivo" In "Bloc de notas"
```

Borra el comando Abrir por posición (se ha vuelto a numerar tras borrar Nuevo).

```
MenuDelete 0 On "&Archivo" In "Bloc de notas"
```

Borra todo el menú Archivo:

```
MenuDelete "" On "&Archivo" In "Bloc de notas"
```

---

Consulte también

[MenuSet](#)

## MenuSelect

Elige un ítem de un menú.

### Sintaxis

MenuSelect <identificación del menú>

### Descripción

Selecciona, desde la barra de menú del programa Windows activo, el ítem de menú identificado mediante el parámetro <identificación del menú>.

El comando MenuSelect se encuentra con más frecuencia en script generados con el comando Create Recording Session. Si crea el script de forma manual, utilice Create Recording Session para recuperar el parámetro <identificación del menú>.

### Ejemplo

El ejemplo presenta cómo elegir Abrir desde el menú Archivo del Bloc de notas.

```
;Ejecuta el Bloc de notas  
exec "Notepad.exe"  
;Elige el comando Abrir del menú Archivo  
Attach "NOTEPAD_Notepad", 1  
Wk_Open = 10  
MenuSelect Wk_Open
```

---

Consulte también

[SysMenuSelect](#)

## MenuSet

Añade un ítem a un menú de aplicación.

### Sintaxis

MenuSet <ítem\$> On <menú\$> In <nombre de la ventana\$> [, <instancia>] [ ID <identificación del menú>]

### Descripción

Añade un ítem de menú más en una aplicación. No es factible crear un menú principal. Todos los ítem de menú añadidos con MenuSet tienen que estar asociados con un suceso de menú creado con el comando Define de ScriptTools.

Ponga el signo (&) a modo de prefijo a los caracteres subrayados de un nombre de ítem o menú utilizados como teclas de método abreviado

El parámetro opcional <identificación del menú> contiene el valor numérico que Windows asocia con esta selección de menú. Utilice <identificación del menú> si el valor predeterminado de 1 ocasiona algún problema. Ha de tener en cuenta, no obstante, que si la identificación elegida coincide con un ítem de menú existente (incluso si pertenece a un menú distinto de la misma aplicación), el ítem recién añadido actúa como el ya existente.

Para insertar un serparador horizontal en el menú, especifique <ítem\$> como un guión "-". Para insertar un separador vertical, introduzca una barra vertical (|) como el primer carácter de <ítem\$>. Windows trata una barra horizontal como si fuera un ítem de menú de todo derecho, pero las barras veticales existen solamente como separadores de columnas.

### Ejemplo

El [ejemplo pormenorizado](#) muestra cómo utilizar los comandos Define y MenuSet para crear un ítem de menú.

---

Consulte también

[MenuDelete](#)

[Define](#)

### **▶** ;Ejemplo de MenuSet

```
;
;Añade un ítem de menú "&WIN.INI" al Administrador de programas.
;Define un suceso de menú para permitir la detección.
; Añade al menú un separador horizontal
;(para dar mayor efecto).
MenuSet "-" On "&File" In "PROGMAN_Program Manager1", 1
; Añade la entrada nueva al menú "&Archivo" del Administrador de
programas.
MenuSet "&WIN.INI" On "&File" In "PROGMAN_Program Manager1", 1
;Define un suceso de menú para detectar el ítem de menú nuevo.
Define Menu Event winini
    Item "&WIN.INI"
    In Module "PROGMAN.EXE"
EndDef
;Establece un "whenever" en el suceso "winini".
whenever winini

;Ejecuta el Bloc de notas y carga "win.ini" cuando el menú está
seleccionado.
    exec "notepad win.ini"
endwhen
;Permite al script esperar al comando "whenever".
suspend
;
;Fin del ejemplo de MenuSet
;
```

### **Minimize**

Reduce una ventana al tamaño de su icono.

#### **Sintaxis**

Minimize ["<nombre de la ventana> "[,<número de instancia>]]

#### **Descripción**

Reduce la ventana especificada al tamaño de su icono. Si no especifica ninguna ventana, se minimiza la activa.

Si especifica una ventana, también puede especificar un número de instancia. El valor predeterminado es que se minimice la primera ventana cuyo nombre coincida con el parámetro nombre de la ventana.

#### **Ejemplo**

Los ejemplos siguientes muestran cómo minimizar la ventana activa y la ventana principal del Administrador de programas.

```
Minimize " PROGMAN_Program Manager1"
```

## Mouse

Activa o desactiva el ratón.

### Sintaxis

```
Mouse On  
Mouse Off
```

### Descripción

Mouse Off desactiva el ratón de manera que los sucesos pertinentes no se lleven a cabo hasta que se ejecute el enunciado Mouse On.

Debido a que el script como tal es capaz de manipular el ratón mientras está vetado al usuario, los comandos MouseClick y MouseMove funcionan.

### Ejemplo

El ejemplo siguiente muestra cómo desactivar el ratón durante 30 segundos.

```
Mouse Off  
;Espera durante 30 .  
Pause 30 seconds  
;Vuelve a activar el ratón.  
Mouse On
```

---

Consulte también

[MouseClick](#)

[MouseMove](#)

[Keyboard On/Off](#)

## MouseClicked

Especifica un clic de ratón.

### Sintaxis

MouseClicked Left | Right | Middle [Button] [Up | Down| Double ] At x, y

### Descripción

Especifica las operaciones de ratón, incluyendo hacer clic con el botón izquierdo, derecho o central (left, right y middle respectivamente). La palabra clave Button es opcional.

Utilice los parámetros Up y Down para arrastrar y soltar un ítem, especificando Down en un punto de la pantalla y Up para soltar el ítem en otro punto. Esta secuencia aparece en el ejemplo. Si no especifica Up o Down, el valor predeterminado es un clic normal.

El parámetro opcional Double sirve para especificar un doble clic.

Las coordenadas "AT x,y" son relativas a la ventana anexa activa.

Los comandos MouseClick suelen crearse en la mayoría de los casos con el comando Create Recording Session. Dicho comando genera automáticamente la sintaxis y coordenadas de ventana correctas. Es posible, si bien difícil, utilizar la función de pegado de lenguaje para un clic de ratón o introducirlo a mano.

### Ejemplo

Este ejemplo muestra la sintaxis correcta para hacer clic con el botón izquierdo, arrastrar y soltar un ítem.

```
MouseClicked Left Button Down At 36, 14  
MouseClicked Left Button Up At 42, 12
```

El ejemplo muestra cómo hacer doble clic con el botón derecho, a fin de seleccionar un ítem.

```
MouseClicked Right Button Double At 25, 34
```

[Consulte también](#)

[MouseMove](#) contiene un ejemplo pormenorizado de gran utilidad.

## MouseMove

Especifica los desplazamientos del ratón.

### Sintaxis

```
MouseMove <x> <y>
```

### Descripción

Desplaza el ratón hasta las coordenadas <x> e <y>, definidas en píxeles, relativas a la ventana activa.

El comando MouseMove proporciona un control preciso del desplazamiento del ratón. Puede crear script que dibujen círculos u otras formas, gracias a la edición de este comando, tal y como se explica en el ejemplo pormenorizado.

El comando MouseMove se crea casi siempre con el comando Create Recording Session. Dicho comando genera automáticamente la sintaxis y coordenadas de ventana correctas. Es posible, si bien difícil, utilizar la función de pegado de lenguaje para las acciones de ratón o introducirlas a mano con precisión.

### Ejemplo

El [ejemplo pormenorizado](#) muestra un script editado que dibuja un círculo en Paintbrush.

```

▶
;Ejemplo de MouseMove
;
;Ejecuta el programa Paintbrush.
;Dibuja un círculo.
;
Exec "pbrush"
Attach "PBRUSH_pbPaint1", 1
MouseClicked Left Button Down At 89, 35
  MouseMove 88 35
  MouseMove 87 35
  MouseMove 85 36
  MouseMove 84 37
  MouseMove 82 39
  MouseMove 81 42
  MouseMove 81 45
  MouseMove 81 47
  MouseMove 80 52
  MouseMove 81 58
  MouseMove 81 61
  MouseMove 82 64
  MouseMove 85 69
  MouseMove 88 72
  MouseMove 89 72
  MouseMove 93 73
  MouseMove 96 73
  MouseMove 102 72
  MouseMove 106 71
  MouseMove 118 67
  MouseMove 122 58
  MouseMove 123 50
  MouseMove 123 47
  MouseMove 119 40
  MouseMove 114 37
  MouseMove 98 36
  MouseMove 94 36
MouseClicked Left Button Up At 89, 37
;
;
;Fin del ejemplo de MouseMove
;

```

## Move

Lleva una ventana a un lugar distinto de la pantalla, pero sin cambiarla de tamaño.

### Sintaxis

Move To <x> <y>

### Descripción

Lleva el ángulo superior izquierdo de la ventana activa a la ubicación especificada por <x> <y> en píxeles. En una pantalla VGA , 0,0 es la esquina superior izquierda y 639, 479 es la esquina superior izquierda.

### Ejemplo

El [ejemplo pormenorizado](#) expone cómo mover la ventana del Bloc de notas por la pantalla.



```
▶
;Ejemplo de Move
;Lleva la ventana del Bloc de notas a la parte superior izquierda de
la pantalla.
;La desplaza hacia abajo 25 píxeles cada vez.
;La desplaza horizontalmente 50 píxeles cada vez.
;Inicia y anexa el Bloc de notas.
exec "notepad.exe"
attach "Notepad_notepad"
;Lleva la ventana al extremo superior izquierdo de la ventana.
move to 0, 0
;La desplaza por la pantalla hacia abajo y en sentido horizontal.
x = 0
y = 0
repeat
    move to x, x
    x = x + 50
    y = y + 50
until x > 300
;
;Fin del ejemplo de script.
;
```

## NetAttach()

Anexa un servidor Novell.

### Sintaxis

NetAttach(<nombre del servidor\$>)

### Resultado

Si NetAttach() resulta satisfactorio, da como resultado 1; de lo contrario, devuelve un 0.

### Descripción

Anexa un servidor Novell.

Para anexarse a otro tipo de servidores, sírvase de la función NetLogin(). Tiene que estar conectado a un servidor antes de poder utilizar NetConnect() para acoplarlo a una unidad local.

Es posible anexarse a cualquier servidor NetWare en el cual pueda utilizar el mismo nombre de usuario que en el servidor primario.

### Ejemplo

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

Consulte también

[NetDetach\(\)](#)

[NetLogin\(\)](#)

[NetConnect\(\)](#)

## NetConnect()

Conecta una unidad logica con un servidor.

### Sintaxis

NetConnect(<ruta de la red\$>, <nombre local\$>, <contraseña\$>)

### Resultado

Si NetConnect() resulta satisfactorio, da como resultado un 1; de lo contrario, da un 0.

### Descripción

Conecta nombres locales de dispositivos al sistema de archivo o dispositivos de un servidor de red tras haberse conectado al servidor y haber accedido a él.

El parámetro ruta de la red\$ contiene una serie con la forma "servidor/volumen:"; el parámetro nombre local\$ contiene la letra de una unidad local a la que se ha de acoplar el volumen del servidor, como "N:". El parámetro contraseña\$ lo utiliza exclusivamente el servidor LAN Manager. Si está conectado a un servidor NetWare, deje ese parámetro en blanco.

Utilice NetDisconnect() para desconectar el servidor de la unidad.

### Ejemplo

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

---

Consulte también

[NetDisconnect\(\)](#)

[NetLogin\(\)](#)

## **NetDetach()**

Desconecta al usuario del servidor especificado.

### **Sintaxis**

NetDetach(nombre del servidor\$)

### **Resultado**

Si resulta satisfactorio, da como resultado un 1; de lo contrario, da un 0.

### **Descripción**

Desconecta al usuario del servidor especificado en el parámetro nombre del servidor\$.

### **Ejemplo**

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

---

[Consulte también](#)

[NetAttach\(\)](#)

## **NetDisconnect()**

### **Sintaxis**

NetDisconnect(nombre local\$)

### **Resultado**

Si resulta satisfactorio, da como resultado un 1; de lo contrario, da un 0.

### **Descripción**

Desconecta el archivo o dispositivo de red designado en el parámetro nombre local\$.

### **Ejemplo**

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

---

Consulte también

[NetConnect\(\)](#)

## **NetLoaded()**

Comprueba la presencia de una red.

### **Sintaxis**

NetLoaded()

### **Resultado**

Si la red está cargada, NetLoaded() devuelve un 1. Si no lo está, da como resultado un 0.

### **Descripción**

Comprueba la presencia de un sistema de red.

### **Ejemplo**

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

## NetLogin()

Permite al usuario acceder y conectarse al servidor de red.

### Sintaxis

NetLogin(nombre del usuario\$, contraseña\$, nombre del servidor\$)

### Resultado

NetLogin() da como resultado un 1 si la conexión es satisfactoria, y un 0 en caso contrario.

### Descripción

Es el protocolo de entrada a una red. Ciertas redes no utilizan el parámetro nombre del servidor\$, pero todas exigen los parámetros nombre del usuario\$ y contraseña\$.

NetLogin() además anexa el servidor especificado si no estuviera ya anexado. Si bien puede utilizar NetAttach() para anexar un servidor Novell NetWare, primero tiene que acceder al servidor NetWare por medio de NetLogin() antes de poder conectarse.

### Ejemplo

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

[Consulte también](#)

[NetLogout\(\)](#)

## NetLogout()

Sale del servidor de red especificado.

### Sintaxis

NetLogout(nombre del servidor\$)

### Resultado

NetLogout da como resultado un 1 cuando resulta satisfactorio y un 0 en caso contrario.

### Descripción

Sale de la red. Ciertas redes no se sirven del parámetro nombre del servidor\$.

### Ejemplo

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

---

Consulte también

[NetLogin](#)

## NetSendMessage()

Envía a los usuarios de la red el mensaje especificado.

### Sintaxis

```
NetSendMessage(destinatario$, nombre del servidor$, mensaje$)
```

### Descripción

Envía mensajes textuales a otros usuarios de la red.

destinatario\$	Variable de serie que contiene el nombre de un usuario particular o de un grupo de usuarios. Si deja en blanco el parámetro destinatario\$, el mensaje llega a todos los usuarios.
nombre del servidor\$	Variable de serie que contiene el nombre del servidor. Si la red no utiliza este parámetro, déjelo en blanco. Si utiliza el parámetro nombre del servidor\$, éste califica el parámetro destinatario\$. Si deja dicho parámetro en blanco, el mensaje llega a los usuarios especificados en todos los servidores a los que esté conectado.
mensaje\$	Variable de serie que contiene el mensaje que desea enviar. Dicho texto puede tener una longitud máxima de 132 caracteres, incluyendo espacios y signos de puntuación.

Si NetSendMessage resulta satisfactorio, devuelve un 1 y, en caso contrario, da como resultado un 0.

### Ejemplo

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

## **NetUserName\$()**

Facilita el nombre de usuario de la red.

### **Sintaxis**

NetUserName\$()

### **Descripción**

Muestra el nombre del usuario conectado a red en ese momento.

Si resulta satisfactorio, da como resultado el nombre del usuario, presentado en forma de variable de serie. En caso contrario, no da ningún nombre.

### **Ejemplo**

Consulte NETTEST.SCT para una demostración de los comandos de red en un contexto más amplio.

## NumLock

Activa y desactiva la tecla de bloqueo del teclado numérico.

### Sintaxis

```
NumLock On  
NumLock Off
```

### Descripción

NumLock determina si la tecla de bloqueo del teclado numérico está activada o no. De esta manera el usuario sabe en qué estado se encuentra el teclado numérico.

### Ejemplo

El ejemplo siguiente activa la tecla Bloq Núm, espera 30 segundos y la desactiva.

```
Numlock On  
Pause 30 seconds  
Numlock Off
```

---

Consulte también

[CapsLock](#)

[Scrolllock](#)

## Pause

Interrumpe un script durante el intervalo especificado.

### Sintaxis

Pause <intervalo> [Sec | Secs| Second| Seconds | Ticks | Tick]

### Descripción

Detiene el script durante el intervalo especificado. Puede especificar el intervalo en segundos o en ticks; si lo especifica en ticks, un tick es 1/10 de segundo (10 ticks = 1 segundo). Si no especifica la unidad cronométrica, el valor predeterminado es segundos.

### Ejemplo

El ejemplo siguiente muestra cómo introducir, en ticks, una pausa de 2 segundos:

```
Pause 20 Ticks
```

## Permit

Permite al usuario utilizar exclusivamente las teclas enumeradas.

### Sintaxis

Permit "<lista de teclas>"

### Descripción

Recupera el uso de las teclas desactivadas mediante el comando Prevent. Sólo se encuentran activadas las teclas que aparecen en la lista.

Permit distingue entre mayúsculas y minúsculas. Hay que enumerar de forma explícita todas las teclas modificadoras que desee activar. Por ejemplo, si desea que el usuario pueda utilizar la a, Ctrl + a y Alt + a, tiene que incluir en la lista las tres combinaciones de teclas.

### Ejemplo

El ejemplo siguiente muestra cómo permitir al usuario utilizar sólo las letras minúsculas "abcdef" tras haber quedado desactivado todo el alfabeto mediante el comando Prevent.

Permit "abcdef"

---

Consulte también

[Prevent](#)

## Prevent

Desactiva las teclas enumeradas.

### Sintaxis

Prevent "<lista de teclas>"

### Descripción

Impide al usuario utilizar las teclas especificadas en la lista. Prevent distingue entre mayúsculas y minúsculas.

Prevent no bloquea aquellas teclas que son detectadas por el hardware o que generan interrupciones, como la secuencia de re arranque: Ctrl + Alt + Supr.

Los comandos Prevent pueden añadirse unos tras otros sin excluirse. Cada comando Prevent añade a la lista más caracteres que no puede utilizar el usuario. No sustituye a la lista anterior. Para volver a activar las teclas especificadas en la lista, hay que hacerlo de forma explícita mediante el comando Permit.

### Ejemplo

El ejemplo siguiente expone cómo impedir que el usuario utilice las letras minúsculas a, b, c, d y e. Las letras mayúsculas A, B, C, D y E se pueden utilizar sin ningún problema.

Prevent "abcde"

---

Consulte también

[Permit](#)

## Procedure

Define la subrutina especificada.

### Sintaxis

```
Procedure <nombre del procedimiento>  
[local <variable local> [, <variable local>, .....]]  
<lista de instrucciones>  
EndProc
```

### Descripción

Usa los procedimientos definidos para que el código sea más modular y para volver a utilizar código normal. Es posible definir un procedimiento cualquiera del código de script. Si utilizan los procedimientos varios programas, defínalos en un archivo .SCT aparte que pueda añadir con el comando Include.

Si lo desea, puede definir las variables de números enteros y de serie locales que se utilizan sólo dentro del procedimiento. Sírvase de comas para separar los nombres de las variables.

Los procedimientos pueden anidarse, pero no anide nunca más de 10.

No es posible pasar parámetros a un procedimiento ni conseguir un resultado a partir de él. Si necesita un módulo al cual poder pasar un parámetro y del cual conseguir un valor de resultado, utilice el comando Function .

No se permite definir un procedimiento dentro de un bloque de código como la construcción Wait or If...Then...EndIf.

### Ejemplo

El [ejemplo pormenorizado](#) invoca un procedimiento que ejecuta una copia del Bloc de notas.

---

Consulte también

[Call](#)

```

▶
;Ejemplo de Procedure
;
;
;Invoca un procedimiento que se detiene durante 1 segundo y
;pasa a continuación a ejecutar una copia de la aplicación Bloc de
notas.
;Tras finalizar el procedimiento, aparece
;un mensaje.
;
call ExecNotePad
;ejecuta la llamada del procedimiento
;cuando termina dicho procedimiento, se ejecuta la línea siguiente.
textpanel 1 "Misión cumplida"
pause 1 second
;Inicia la definición del procedimiento.
Procedure ExecNotePad
;Comandos que se ejecutan dentro del procedimiento:
    exec "NOTEPAD.EXE"
    pause 1 second
EndProc
;
;Fin del ejemplo de Procedure
;

```

## RadioButton

Selecciona un botón de opción.

### Sintaxis

```

RadioButton <identificación de control> <SingleClick>
RadioButton <identificación de control> On

```

### Descripción

Selecciona el botón de opción especificado por el parámetro <identificación de control>. El parámetro SingleClick cambia el estado del botón, es decir, lo activa si estaba desactivado y viceversa. Normalmente se ve esta modalidad en los script creados con Create Session.

Los parámetros On y Off establecen explícitamente el estado del botón de opción en On (con la equis) o Off (sin la equis), activada y desactivada respectivamente.

Utilice Create Attach Name cuando inserte el comando RadioButton de forma manual. Esta utilidad recupera el código de identificación de control del botón.

El comando RadioButton no es lo mismo que la palabra clave RadioButton empleada en los enunciados de definición de cuadro de diálogo para crear un botón de opción.

### Ejemplo

Los ejemplos siguientes muestran distintas versiones del comando RadioButton.

```

RadioButton 12 SingleClick
RadioButton 12 On

```

Consulte también

[Dialog](#)

[Creación de un botón de opción](#)

## Read

Lee los datos de un archivo y los almacena en una variable.

### Sintaxis

Read [FROM] <nombre del archivo> [Into] <variable>

Read [FROM] <nombre del archivo> [Into] <variable> Length <longitud>

Read [FROM] <nombre del archivo> [Into] <variable> [,<variable>,...] Delimited [By] { CRLF | Comma }

### Descripción

Lee el contenido del archivo especificado y almacena los datos en una o más variables.

El caso más simple del comando Read lee datos en bruto y los almacena en una variable. El tipo de la variable determina la longitud del registro.

▶ Si la variable es una serie, el archivo se lee hasta el primer delimitador o un mínimo de 32766 caracteres.

▶ Si la variable es un número entero, se leen dos bytes.

▶ Si la variable es Long, se leen cuatro bytes.

El parámetro Length modifica la longitud del registro. Sin embargo, dicha longitud no puede sobrepasar el tamaño de la variable. En el caso de una variable de serie, se leen los datos hasta que se llega a la longitud especificada por el parámetro Length, se encuentra un delimitador o se hayan leído 32766 caracteres.

Si especifica el parámetro Delimited, se leen los datos hasta que se encuentre el delimitador especificado o se alcance el tamaño de la variable. Se permiten dos delimitadores: CRLF y Comma. Durante la lectura de datos de un archivo, una serie nula se considera también como delimitador.

### Ejemplo

Se facilitan dos ejemplos pormenorizados.

Ejemplo uno escribe datos en un archivo y los lee en una variable Long y de serie.

Ejemplo dos lee AUTOEXEC.BAT en forma de variable de serie, línea por línea.

Consulte también

Write

```

▶
;Ejemplo de Read/Write
;
;
;Escribe datos en un archivo.
;Lee los datos del archivo.
;Presenta los datos.
;
;Define las variables largas.
Long c, e
;Comprueba que el archivo "temp.txt" está vacío.
create "temp.txt"
;Escribe los datos.
a$ = "Hola a todos"
b = 50
c = 100000L
write to "temp.txt" from a$, b, c
;Vuelve a leer los datos.
read from "temp.txt" into b$, d, e
;Presenta los resultados.
textpanel 1 b$ + ", " + d + ", " + e
;Deja transcurrir cierto tiempo para ver los resultados.
pause 2 seconds
;
;
;Fin del ejemplo de Read/Write
;

```

```

▶
; Ejemplo 2 de Read
;
;
;Lee el archivo "autoexec.bat" y lo presenta línea por línea.
;
;
;NOTA:
;Asume que "autoexec.bat" está en la unidad "C:\".
;
repeat
read from "c:\autoexec.bat" into a$ delimited by CRLF
textpanel 1 a$
pause 1 second
until _eof = 1
;
;
;Fin del ejemplo de Read
;

```

## Rem

Agrega comentarios en su script.

## Sintaxis

```

Rem <comentario>
Remark <comentario>
; <comentario>

```

**Descripción**

Precede a los comentarios que describen los enunciados del programa de script. Si los comentarios se dividen en varios renglones, cada renglón tiene que empezar con un comando Rem o un punto y coma (;).

**Ejemplo**

; Todo renglón que empiece con un punto y coma es un comentario.

Rem      Éste es otro comentario.

Remark   Y éste otro más.

## Remove

Elimina de pantalla un mensaje de panel de texto.

### Sintaxis

Remove <número del panel>

### Descripción

Borra el mensaje presentado por un enunciado de TextPanel. El parámetro <número del panel> es un número entero comprendido entre 0 - 24. Si el <número del panel> no es válido, no ocurre nada.

### Ejemplo

El [ejemplo pormenorizado](#) muestra tres paneles de texto y los borra en el orden inverso de presentación.

---

Consulte también

[TextPanel](#)

```

▶
;Ejemplo de Remove/Repeat/While
;
;Presenta una serie de paneles de texto mediante While...EndWhile.
;Elimina los paneles en el orden inverso de presentación, utilizando
Repeat...until.
;determina el valor inicial de i
i=0
;A continuación aparece la construcción While
While i<5 do
    textpanel i "Éste es el panel " + i
    pause 1 second
    i = i + 1
EndWhile

;Deja transcurrir un intervalo de tiempo para que se vea.
pause 3 seconds
;
;Elimina los paneles en el orden inverso mediante Repeat...until.
repeat
    remove i
    pause 1 second
    i = i - 1
until i=0
;
;
;Fin del ejemplo de Remove/Repeat/While
;

```

## Rename

Cambia un archivo de nombre.

### Sintaxis

Rename "<especificación del archivo>" To "<especificación nueva del archivo>"

### Descripción

Cambia el nombre del archivo o serie de archivos que casan con la especificación del archivo. Los caracteres comodines están admitidos. Si especifica un nombre de archivo inválido, se acaba el script.

Con Rename, puede trasladar a distintos directorios aquellos archivos cuyas rutas especifique, pero no los puede trasladar a una unidad distinta.

### Ejemplo

Rename "c:\windows\3270.txt" To "c:\3270.txt"

Consulte también

Copy

Delete

## Repeat

Repite una lista de instrucciones hasta que se evalúa como verdadera la expresión especificada.

### Sintaxis

```
Repeat  
<instrucciones>  
Until <expresión>
```

### Descripción

Ejecuta una función de bucle con la expresión evaluada al final del bucle, lo cual implica que el bucle se ejecuta siempre una vez por lo menos.

Las expresiones múltiples pueden evaluarse con el enunciado Until.

### Ejemplo

Consulte el [ejemplo pormenorizado](#) del comando Remove.

## Resume

Termina un enunciado Suspend.

## Sintaxis

Resume

## Descripción

Cancela un enunciado Suspend.

Resume se ejecuta dentro de una construcción Whenever para reanudar la ejecución del script después de la realización de un suceso especificado.

## Ejemplo

El [ejemplo pormenorizado](#) muestra cómo suspender un programa hasta que el usuario presiona las teclas Ctrl + Retroceso.

---

Consulte también

[Suspend](#)

[Whenever](#)

```

▶
;Ejemplo de Resume
;
;Suspende un script de forma que se quede a la espera de
;la secuencia {Ctrl {Retroceso}}.
;Reanuda el script al detectar la secuencia del teclado.
;Muestra el mensaje antes de acabar.
;Define el suceso de teclado al que hay que esperar.
Define Keyboard Event Eventkey
    "{Ctrl {Backspace}}" Throwaway
    In Any Window
EndDef
;Establece whenever respecto al suceso de teclado.
whenever Eventkey
    resume
endwhen
;Solicita la operación exigida
textpanel 1 "Pulse las teclas Ctrl + Retroceso para reanudar"
;Hace esperar al script
suspend
;Enseña que se ha reanudado el script.
textpanel 1 "Reanudación en proceso"
;Da tiempo para leer el mensaje.
pause 2 seconds
;
;
;Fin del ejemplo Resume/Suspend
;

```

## Restore

Restaura la ventana a su tamaño original.

### Sintaxis

```

Restore
Restore "<nombre de la ventana>" [, <número de instancia>]

```

### Descripción

Devuelve el tamaño original de la ventana. No tiene porqué ser el tamaño máximo. Si no especifica el <nombre de la ventana>, se restaura la ventana activa.

Si especifica el <nombre de la ventana>, puede especificar también el <número de instancia> para maximizar una instancia específica de la ventana en cuestión. El valor predeterminado es que se restaure el tamaño de la primera ventana cuyo nombre coincida con el parámetro <nombre de la ventana>.

### Ejemplo

Los ejemplos siguientes muestran cómo restaurar la ventana actual y la del Bloc de notas.

```

Restore
Restore "NOTEPAD_notepad"

```

## Run

Ejecuta otro script en forma de subrutina.

### Sintaxis

Run "<nombre del script > <lista de parámetros>"

### Descripción

Carga en la memoria el script especificado, lo ejecuta y devuelve el control al script original una vez completada la ejecución o al encontrarse un comando Exit command.

### Ejemplo

El ejemplo muestra dos script breves. El primero utiliza el comando Run para ejecutar el segundo script, que ejecuta el Bloc de notas. El primer script escribe en el Bloc de notas "Hola a todos".

```
MAIN.SCT
;Ejecuta el segundo script compilado.
run "agent2.agt"
;Utiliza la copia del Bloc de notas.
Attach "NOTEPAD_Edit1", 1
type "Hola a todos"
;Fin del ejemplo script : MAIN.SCT

AGENT2.SCT
exec "notepad"
;Fin del ejemplo script : AGENT2.SCT
```

---

Consulte también

[Chain](#)

## ScrollBar

Mueve las barras de desplazamiento.

### Sintaxis

```
ScrollBar {<identificación de control> | [Window] {Horz | Vert}} <unidades> [Set | Page | Line]
```

### Descripción

Mueve la barra de desplazamiento de la ventana anexa o especificada por <identificación de control>. Con Set se desplaza media página. Line y Page tienen el mismo efecto que hacer clic en las flechas y en las áreas de página de las barras de desplazamiento. El parámetro <unidades> de Line o Page se interpreta tomando la posición actual como punto de referencia.

La única unidad válida para Line o Page es 1 (abajo) ó -1 (arriba).

### Ejemplo

El [ejemplo pormenorizado](#) muestra la manipulación de las barras de desplazamiento en el Bloc de notas.

```
▶  
;Ejemplo de ScrollBar  
;  
;Ejecuta el Bloc de notas y carga el archivo "win.ini" file.  
;  
;Mueve la barra de desplazamiento mediante el comando ScrollBar.  
;  
;Ejecuta el Bloc de notas (cargando "win.ini").  
exec "notepad win.ini"  
;Usa la ventana principal del Bloc de notas.  
Attach "NOTEPAD_Notepad"  
;Baja la barra 1 línea  
ScrollBar Window Vert 1 Line  
;Sube la barra 1 línea  
ScrollBar Window Vert -1 Line  
;Baja la barra media página  
ScrollBar Window Vert 1 Set  
;Sube la barra 1 página  
ScrollBar Window Vert -1 Page  
;  
;  
;Fin del ejemplo de ScrollBar  
;
```

## Scrolllock

Activa y desactiva la tecla de bloqueo de desplazamiento.

### Sintaxis

```
Scrolllock On  
Scrolllock Off
```

### Descripción

Determina si la tecla de bloqueo de desplazamiento está activada o no. De esta manera el usuario sabe en qué estado se encuentra el teclado.

### Ejemplo

Los ejemplos siguientes muestran las dos posibilidades de este comando.

```
Scrolllock On  
Scrolllock Off
```

---

Consulte también

[NumLock](#)

[CapsLock](#)

## **SetFocus**

Activa una ventana de aplicación en la que poder realizar operaciones.

### **Sintaxis**

SetFocus <nombre de la ventana> [,<número de instancia>]

### **Descripción**

Activa la ventana especificada a fin de poder realizar operaciones. Si no especifica ninguna, no ocurre nada. Si no especifica un <número de instancia>, se activa la primera ventana cuyo nombre coincida con el parámetro <nombre de la ventana>. Si especifica el <número de instancia>, tiene que casar tanto el <nombre de la ventana> como el <número de instancia>.

### **Ejemplo**

El ejemplo siguiente activa la ventana número 1 del Bloc de notas.

```
SetFocus "NOTEPAD_notepad", 1
```

## Size

Hace que la ventana de aplicación especificada adopte el tamaño definido.

### Sintaxis

Size To <anchura> <altura>

### Descripción

Modifica el tamaño de la ventana según la altura y anchura especificadas. Los parámetros <anchura> y <altura> se expresan en píxeles. En una pantalla VGA, 0,0 es la esquina superior izquierda, mientras que 639, 479 es el extremo inferior izquierdo.

### Ejemplo

El [ejemplo pormenorizado](#) cambia el tamaño de la ventana del Administrador de programas.

```
▶  
;Ejemplo de Size  
;Este ejemplo ejecuta el Bloc de notas  
; y cambia el tamaño de la ventana  
exec "notepad.exe"  
attach "NOTEPAD_notepad"  
;Reduce la ventana a 0,0 para observar mejor los cambios de tamaño.  
move to 0, 0  
;Adopta varios tamaños.  
w=0  
h = 0  
repeat  
    size to w,h  
    w = w + 50  
    h = h + 40  
until w > 400  
;  
;  
;Fin del ejemplo de Size
```

## Stop

Termina todos los script.

### Sintaxis

Stop

### Descripción

Detiene todos los script en ejecución, incluyendo los script padres. El control es devuelto o bien a la aplicación o a Windows. Stop es como Exit, con la diferencia que terminan los script padres, así como el programa que invoca Stop.

### Secuencia de teclado de método corto

Existe una secuencia de teclas especial que termina todos los programas en ejecución. Pulse Mayúsculas + Ctrl + Pausa. Presione las teclas una tras la otra, no todas a la vez (la tecla Pausa suele ser F15 en la mayoría de los teclados extendidos).

## Suspend

Suspende la ejecución de un script.

### Sintaxis

Suspend

### Descripción

Suspende el programa hasta que se da el suceso especificado por un enunciado Whenever. A continuación ejecuta los comandos definidos y espera al siguiente suceso Whenever.

Si ejecuta un enunciado Resume en una subrutina de enunciado Whenever, Suspend queda cancelado y se procede a ejecutar el siguiente enunciado.

### Ejemplo

El [ejemplo pormenorizado](#) de Resume sirve también para Suspend.

---

Consulte también

[Whenever](#)

[Resume](#)

## Switch

Ejecuta un conjunto de instrucciones de acuerdo con el valor de la variable especificada.

### Sintaxis

```
Switch <variable> [Of | From| On]
  Case<valor x de la variable>
  <instrucciones>
  EndCase
  Case....EndCase
Default
  <instrucciones>
EndSwitch
```

### Descripción

Selecciona una de varias opciones mediante la evaluación y comparación del valor de <variable> con <valor n de la variable> de los enunciados Case particulares.

Si bien puede tener muchos enunciados Case particulares, tiene que introducir una lista de instrucciones Default, que se usa cuando no se evalúa ninguno de los enunciados Case como verdadero.

Los valores de las variables pueden ser números enteros o series. La conversión de tipos de letra se realiza automáticamente antes de la comparación.

### Ejemplo

El [ejemplo pormenorizado](#) muestra cómo ejecutar varios programas accesorios con un enunciado Switch que utiliza la entrada de la solicitud como la variable.

```

▶
;Ejemplo de Switch
;
;Introduce una serie por medio de la función Promptbox().
;
;Conmuta de acuerdo con el valor de la serie.
;
;Introduce el nombre (NOTEPAD, CALC o Write).
a = promptbox( "Prueba", "¿Qué accesorio desea ejecutar?", a$ )
;Convierte todo en mayúsculas para evitar problemas.
a$ = Uppercase$( a$ )
;Observa si tiene que realizar alguna operación.
switch a$ on
  case "NOTEPAD"
    exec "Notepad"
  endcase
  case "CALCULATOR"
    exec "Calc"
  endcase
  case "Write"
    exec "Write"
  endcase
  default
    textpanel 1 "El nombre " + a$ + " no corresponde a ningún
accesorio conocido"
    pause 2 seconds
endswitch
;
;
;Fin del ejemplo de script.
;

```

## SysMenuSelect

Realiza selecciones desde el menú Control de la aplicación.

### Sintaxis

```
SysMenuSelect <ítem>
```

### Descripción

Selecciona una opción del menú del sistema de la ventana activa, que es el que aparece al hacer clic en la esquina superior izquierda de la ventana. Este comando se ve con mucha frecuencia en los scripts creados con Create Recording Session.

Si introduce a mano un comando SysMenuSelect en un script , tiene que utilizar Recording Session para conseguir el valor de <ítem>.

### Ejemplo

El ejemplo ejecuta el Bloc de notas y lo cierra mediante el cuadro de cierre del menu de la ventana.

```

;Ejecuta el Bloc de notas.
exec "Notepad"
;Lo cierra mediante el menú del sistema.
Attach "NOTEPAD_Notepad - (Untitled)1", 1
Wk_CloseAltF4 = -4000
SysMenuSelect Wk_CloseAltF4

```

---

Consulte también  
[MenuSelect](#)

## TextPanel

Presenta en pantalla un mensaje de texto.

### Sintaxis

TextPanel <número de panel> <mensaje> [At <x>, <y>]

### Descripción

Muestra un mensaje de texto. El parámetro <número de panel> es un número comprendido entre 0 y 24, lo cual quiere decir que pueden verse hasta 25 mensajes a la vez. Si utiliza el mismo número de panel más de una vez, el texto más reciente sustituye al primitivo.

Emplee los parámetros opcionales de ubicación At para colocar el panel en pantalla en el punto indicado por las coordenadas de píxel x,y. Sírvese de Remove para borrar el cuadro y mensaje de un panel de texto.

El parámetro <mensaje> enumera las variables o texto que ha de aparecer en pantalla. Escriba el texto entre comillas dobles. Inserte un signo de más entre los elementos del mensaje, como puede apreciar en el ejemplo.

El script borra todos los paneles de texto al salir.

El comando TextPanel no crea una pausa en el programa. Si desea que se produzca una interrupción para poder leer el panel de texto, introduzca un enunciado Pause o Wait.

### Ejemplo

Este ejemplo muestra cómo presentar un mensaje de advertencia en TextPanel 1.

```
textpanel 1 "El nombre " + a$ + " no corresponde a ningún accesorio conocido"
```

El [ejemplo pormenorizado](#) facilita varios enunciados TextPanel.

---

Consulte también

[Remove](#)

```

▶
;Ejemplo de TextPanel
;
;Muestra un panel de texto durante 10 segundos y lo borra a
continuación.
;
;Presenta varios tipos de variables en un segundo panel.
Long d
d = 100000L
a$ = "Un mensaje"
b = 50
;Muestra el mensaje durante 10 segundos y lo borra a continuación.
textpanel 3 "Espere durante 10 segundos..." AT 300,300
;Espera durante 10 segundos
pause 10 seconds
;Borra el panel de la pantalla.
remove 3
;Presenta un panel con varios tipos de datos.
textpanel 1 a$ + "," + b + "," + d
;Deja transcurrir el tiempo necesario para ver el panel.
Pause 1 second
;
;
;Fin del ejemplo de TextPanel
;

```

## Type

Simula pulsaciones de teclado.

### Sintaxis

Type <texto, incluyendo teclas especiales>

### Descripción

Introduce caracteres y teclas especiales en la ventana activa de la aplicación. Es posible enviar un máximo de 255 caracteres con un único comando Type. Para introducir más caracteres, utilice más de un comando Type. El carácter de retorno de carro/salto de línea tiene que introducirse como un literal, utilizando {Return}, como aparece en la lista inferior.

Type matches the case of alphabetic characters in the script. Admite las teclas especiales siguientes:

Teclas de función {F1} ...{F10}

Barra espaciadora {Space}

Teclado numérico {K/} {K\*} {K-} {K+} {K0}...{K9} {Return}

Varios {Tab} {Escape} {Return} {Numlock} {Scrollock} {Insert} {Delete} {Up} {Down} {Left} {Right} {PageUp} {PageDown} {Home} {End} {Rightbrace} {Leftbrace} {Dquote}. (Tabulación, Escape, Retorno, Bloq Num, Bloq Despl, Ins, Supr, flecha arriba, flecha abajo, Inicio, Fin, corchete inicial, corchete final y comillas dobles, respectivamente.)

Además, pueden usarse las teclas modificadoras {Ctrl}, {Alt} y {Shift} (Mayúsculas) en conjunto con cualquier otra tecla.

### Ejemplo

Estos ejemplos de Type muestran cómo introducir las pulsaciones de teclado.

```
Type "El perro de San Roque no tiene rabo."
```

```
Type "{Return}"
```



## Wait

Interrumpe el script hasta que se da un suceso específico.

### Sintaxis

```
Wait [<intervalo> Seconds | Ticks] Until <condición>
```

```
[ ElseWait  
  <instrucciones>  
EndWait ]
```

```
Wait [<intervalo> Seconds | Ticks ] Until Any of <condición1>, <condición2>,...
```

```
Wait [<intervalo> Seconds | Ticks ] Until All of <condición1>, <condición2>,...
```

### Descripción

Wait...ElseWait...EndWait interrumpe el script hasta que se da el suceso definido o transcurre el intervalo. Si se acaba el intervalo y ha especificado la condición opcional ElseWait, se llevan a cabo las instrucciones de la misma.

Wait...Until Any of... sigue las instrucciones cuando transcurre el intervalo o se evalúa una instrucción cualquiera de un conjunto de instrucciones como verdadera. Wait...Until All of... lleva a cabo las instrucciones cuando el intervalo se acaba o cuando todas las instrucciones de un conjunto se evalúan como verdaderas. La opción ElseWait no es válida con las formas Any of y All of del enunciado Wait.

Hay 10 ticks en un segundo.

### Ejemplo

El ejemplo siguiente muestra cómo definir un suceso y esperar hasta que ocurra antes de salir del programa.

```
Define Mouse Event TheEnd  
  Right button Double Throwaway  
  In Any Window  
EndDef  
Wait until TheEnd  
Exit
```

Consulte también

[Define](#)

## Whenever

Define los enunciados que tienen que siempre que se dé un suceso definido.

### Sintaxis

```
Whenever <nombre del suceso>  
    <lista de enunciados>  
EndWhen
```

### Descripción

Define la secuencia de enunciados que ha de llevarse a cabo cada vez que se dé el suceso especificado. El script busca sucesos Whenever por resolver mientras procesa un enunciado Pause, Wait o Suspend.

Tras establecer una construcción Whenever, el script sigue la operación. La lista de enunciados Whenever se ejecuta sólo cuando se da el suceso o sucesos especificados.

### Ejemplo

El [ejemplo pormenorizado](#) muestra cómo hacer que suene una melodía cuando se pulsa una tecla.

---

Consulte también

[Pause](#)

[Wait](#)

[Suspend](#)

[Cancel](#)

[Resume](#)

```

▶
;Ejemplo de Whenever
;
;Hace sonar una melodía, valiéndose del comando "beep", siempre que se
pulsen las teclas {Ctrl a}.
;
;Define el suceso clave {Ctrl a}
; teniendo en cuenta que los sucesos de teclado distinguen entre
mayúsculas y minúsculas
define Keyboard Event keyev1
    "{Ctrl a}" throwaway
    in any window
enddef
;A continuación define el comando Whenever.
whenever keyev1
    beep "abcd"
    resume
endwhen
;Permite que el script siga activo
suspend
;
;
;Fin del ejemplo de Whenever.
;

```

## While

Lleva a cabo una lista de instrucciones siempre que se dé una condición.

### Sintaxis

```

While <condición> [Do]
    <instrucciones>
EndWhile

```

### Descripción

Permite la ejecución de una lista de instrucciones durante el tiempo en el que se cumpla la condición. Debido a que la condición se evalúa antes de llevar a cabo las instrucciones, es posible que no se lleguen a realizar.

Los enunciados While admiten más de una condición.

### Ejemplo

El [ejemplo pormenorizado](#) de Remove contiene tanto el bucle While...Do..EndWhile como el Repeat...Until.

```

i=0
While i<5 do
    textpanel i "This is panel " + i
    pause 1 second
    i = i + 1
EndWhile

```

## Write

Escribe en un archivo el contenido de una variable.

### Sintaxis

Write [To] <nombre del archivo> [From] <variable>

Write [To] <nombre del archivo> [From] <variable> Length <longitud>

Write [To] <nombre del archivo> [From] <variable> [,<variable>,.....] Delimited [By] { CRLF | Comma }

### Descripción

Escribe el contenido de una variable, con frecuencia una matriz, en un archivo especificado.

El caso más sencillo del comando Write escribe los datos de una variable y los almacena en un archivo. El tipo de variable determina la longitud del registro.

▶ Si la variable es una serie, el registro termina tras 32766 caracteres o en el primer carácter delimitador.

▶ Si la variable es un número entero, el registro tiene una longitud de dos bytes.

▶ Si la variable es Long, el registro tiene una longitud de cuatro bytes.

El parámetro Length modifica la longitud del registro. Sin embargo, dicha longitud no puede sobrepasar el tamaño de la variable. En el caso de una variable de serie, se escriben los datos hasta que se llega a la longitud especificada por el parámetro Length o se encuentra un delimitador.

Si especifica el parámetro Delimited, se escribe y delimita el valor de la variable tal y como está especificado. Se permiten dos delimitadores: CRLF y Comma.

Utilice PadString\$() para crear series de una longitud especificada de forma que pueda escribir registros de longitud fija en un archivo. La longitud máxima de una serie es de 32766 bytes.

### Ejemplo

El [ejemplo pormenorizado](#) muestra cómo escribir datos en un archivo y leerlos a continuación.

---

Consulte también

[Read](#)

[PadString\\$\(\)](#)

## Funciones de sistema Windows

A continuación se enumeran las funciones de sistema Windows disponibles:

<u>ActiveWindow\$()</u>	<u>NetDetach()</u>
<u>AmPm\$()</u>	<u>NetDisconnect()</u>
<u>Capture\$()</u>	<u>NetLoaded()</u>
<u>Chr\$()</u>	<u>NetLogin()</u>
<u>CommandLine\$()</u>	<u>NetLogout()</u>
<u>Date\$()</u>	<u>NetSendMessage()</u>
<u>Day\$()</u>	<u>NetUserName\$()</u>
<u>DDEExecute()</u>	<u>PadString\$()</u>
<u>DDEInitiate()</u>	<u>PromptBox()</u>
<u>DDEPoke()</u>	<u>Random()</u>
<u>DDERequest\$()</u>	<u>ReadIni\$()</u>
<u>DDETerminate()</u>	<u>ReadIni()</u>
<u>DLLCall()</u>	<u>RemoveDir()</u>
<u>FileExists()</u>	<u>Right\$()</u>
<u>Focus\$()</u>	<u>Running()</u>
<u>Focus()</u>	<u>Secs()</u>
<u>FocusWindow\$()</u>	<u>SubStr\$()</u>
<u>Hours()</u>	<u>Time\$()</u>
<u>Instr()</u>	<u>Time()</u>
<u>Left\$()</u>	<u>TopInstance()</u>
<u>Length()</u>	<u>TopWindow\$()</u>
<u>Lowercase\$()</u>	<u>Uppercase\$()</u>
<u>MakeDir()</u>	<u>Version\$()</u>
<u>Maximized()</u>	<u>Version()</u>
<u>MessageBox()</u>	<u>WeekDay\$()</u>
<u>Minimized()</u>	<u>Weekday()</u>
<u>Mins()</u>	<u>WindowExists()</u>
<u>Month\$()</u>	<u>WriteIni\$()</u>
<u>MouseWindow\$()</u>	<u>WriteIni()</u>
<u>NetAttach()</u>	<u>Year()</u>
<u>NetConnect()</u>	

### ActiveWindow\$()

Da el nombre de la ventana activa.

#### Sintaxis

ActiveWindow\$()

#### Descripción

Da como resultado el nombre de la ventana activa. Por ejemplo, a\$=ActiveWindow\$() da "NOTEPAD\_Edit1" en a\$ si esa es la ventana actual.

Puede utilizar el nombre dado por ActiveWindow\$() para anexar esa misma ventana tras cambiar la ventana o aplicación activa.

## Capture\$()

Copia el texto de la ventana especificada.

### Sintaxis

Capture\$(<nombre de la ventana>[,<número de instancia>[,<formato>]])

### Resultado

El texto que aparece en la ventana.

### Descripción

Captura el texto que aparece en la ventana especificada. El parámetro <número de instancia> es opcional a menos que especifique el <formato>.

Tenga en cuenta que esta función sólo da como resultado el texto que aparece en pantalla. Si desea capturar texto de un control de edición que no aparece en pantalla, utilice la función GetText\$(), que da todo el texto, independientemente de que aparezca en pantalla o no.

Si no está especificado el <formato> o está en cero, se concatenan todas las líneas de texto. Si el formato está en 1, las líneas de texto se separan por un retorno de carro/salto de línea. Por ejemplo, a\$=Capture\$( "NOTEPAD\_Edit1", 1,1) da como resultado el texto visible de la ventana "NOTEPAD\_Edit1," separando las líneas según van apareciendo en la ventana.

---

Consulte también

[GetText\\$\(\)](#)

## Chr\$()

Transforma un código ASCII en una serie.

### Sintaxis

Chr\$( <valor ascii> )

### Resultado

Una serie con su equivalente del código ASCII.

### Descripción

Da como resultado el carácter equivalente de un código ASCII de no impresión, como el retorno de carro [13] y el salto de línea [10], con objeto de que otras funciones de serie puedan manipularlos.

Por ejemplo, a\$=Chr\$(13) coloca el valor ASCII 13 en la serie a\$, que puede utilizarse, de esta manera, para forzar un retorno de carro.

## **CommandLine\$()**

Da como resultado las opciones de línea de comando especificadas empleadas para ejecutar un script.

### **Sintaxis**

CommandLine\$(<número de opción>)

### **Resultado**

La opción de línea de comando especificada, a modo de serie.

### **Descripción**

Recupera las opciones de línea de comando pasadas a un script invocado por otro script, mediante los comandos Chain o Run, o ejecutado por una llamada Exec a la aplicación ScriptRunner, WNSRUN.EXE.

El parámetro cero es siempre el nombre del script . Por ejemplo, si devuelve un script denominado MYBAT.RUN, CommandLine\$(0) da como resultado MYBAT.RUN.

Otros números de parámetros dan las opciones en el orden en el que se han introducido en la línea de comando del script de llamada.

---

Consulte también

[Run](#)

[Chain](#)

## DLLCall()

Invoca Dynamic Link Libraries (DLL o bibliotecas de enlace dinámico).

### Advertencia

Utilice esta función exclusivamente si entiende completamente las implicaciones que conlleva la invocación de una DLL.

### Sintaxis

DLLCall( <nombre de la dll>, <nombre de la función>, <parámetro 1>,.....,<parámetro n> )

### Resultado

DLLCall() da como resultado el código procedente de la función DLL ejecutada como Long. Si el número que espera obtener como resultado es inferior a 32767, puede utilizarlo en una variable de número entero para almacenar el valor de resultado.

### Descripción

Invoca DLL que no pertenecen a ScriptTools, incluyendo las llamadas al API Windows.

DLLCall() acepta o da como resultado solamente variables largas, números enteros y series. No admite estructuras ni datos procedentes de miembros de estructuras.

### Ejemplo

Estos fragmentos de código exponen cómo utilizar DLLCall para recuperar las coordenadas de pantalla de una ventana.

```
;coordenada de pantalla x
sm_cxscreen=0
;coordenada de pantalla y
sm_cyscreen=1
;escribe la coordenada de pantalla x en la pantalla screen_x
screen_x=DLLCall("User","GetSystemMetrics",sm_cxscreen)
;escribe la coordenada de pantalla y en la pantalla screen_y
screen_y=DLLCall("User","GetSystemMetrics",sm_cyscreen)
```

## **FileExists()**

Se cerciora de la existencia del archivo o ruta especificados.

### **Sintaxis**

FileExists(<nombre de la ruta>)

### **Resultado**

Da 1 si existe el archivo y 0 en caso negativo.

### **Descripción**

Verifica la existencia de un archivo o ruta específicos.

Introduce el nombre entero de la ruta del archivo. Por ejemplo, a=FileExists( "c:\windows\win.ini" ) da como resultado un 1 si el archivo existe y 0 en caso negativo.

Para comprobar la existencia de C:\WINDOWS, introduzca solamente el nombre de la ruta.

## **Focus(), Focus\$() y FocusWindow\$()**

Dan como resultado información sobre la ventana o programa activos.

FocusWindow\$( )

Facilita el nombre de la ventana activa. Por ejemplo, a\$=FocusWindow\$() da "NOTEPAD\_Edit1" en a\$ si es ésta la ventana activa.

Focus(<nombre de la ventana>)

Devuelve 1 si el módulo o ventana especificados están activos. Utilice el nombre empleado para anexar la ventana o el devuelto por FocusWindow\$() para especificar el <nombre de la ventana>.

Por ejemplo, a=Focus( "NOTEPAD\_Edit1" ) da 1 si la ventana de edición del Bloc de notas es la ventana activa.

Focus\$( )

Da como resultado el nombre de la aplicación activa. Por ejemplo, a\$=Focus\$() da "NOTEPAD.EXE" si la aplicación activa es el Bloc de notas.

## GetText\$()

Recupera texto de una ventana de pantalla como por ejemplo, un cuadro de control de edición en el que cabe la posibilidad de que haya texto que no aparece en pantalla, o una ventana que contiene texto que incluye líneas en blanco.

### Sintaxis

```
GetText$(<nombre de anexo> [, <número de instancia>])
```

### Resultado

El texto de la ventana especificada o una línea en blanco si la ventana responde con el título en lugar de con el contenido.

### Descripción

Tanto GetText\$() como Capture\$() dan como resultado el texto que aparece en una ventana. Emplee GetText\$() para recuperar texto de un control de edición que no muestra todo el texto introducido o de una ventana de edición que contenga líneas en blanco.

Si GetText\$() da como resultado una línea en blanco debido a que la ventana no facilita el contenido, sírvase del comando Capture\$().

### Ejemplo

Para capturar texto de un control de edición, utilice la utilidad Create Attach Name a fin de conseguir el nombre de anexo de control de edición y, a continuación, use este nombre como un argumento de GetText\$(), como muestra este ejemplo del cuadro de diálogo Guardar como de Windows Write.

```
a$=GetText("WRITE_Edit1")
```

---

Consulte también

[Capture\\$\(\)](#)

## GetEnvironment\$()

Da como resultado la definición de cualquier variable del entorno DOS.

### Sintaxis

```
GetEnvironment$("<variable del entorno>")
```

### Resultado

El parámetro actual de la variable especificada, en forma de serie.

### Descripción

Da la definición de una variable del entorno especificado. Es de gran utilidad para verificar la existencia de un archivo en un directorio cuyo nombre es distinto según el usuario, pero que está definida en una variable de entorno.

Tenga en cuenta que GetEnvironment\$() no da como resultado la barra inclinada final del nombre del directorio. Inserte usted mismo dicha barra en el nombre de ruta del programa al utilizar FileExists(), como muestra el ejemplo.

### Ejemplo

Para averiguar si un usuario ha instalado un programa Windows antes de intentar abrirlo, sírvase de GetEnvironment\$() con objeto de conocer el nombre del directorio Windows, como muestra el fragmento de código inferior.

```
Program$="notepad.exe"
CheckDir$=GetEnvironment$("WINDIR")
FileName$=CheckDir$ + "\" + Program$
a=FileExists( FileName$)

If a=1 then
    Exec Program$
else
    Exit
EndIf
```

## Minimized()

Determina si una ventana de aplicación se reduce al tamaño de su icono.

### Sintaxis

Minimized("<nombre de la ventana>", <número de instancia>)

### Resultado

Da 1 si se ha minimizado la ventana especificada.

### Descripción

Especifique <nombre de la ventana> y <número de instancia> con el mismo nombre y número de instancia que haya utilizado para anexar la ventana. Por ejemplo, Minimized( "NOTEPAD\_Edit1", 1 ) da como resultado 1 si la ventana del Bloc de notas está minimizada.

---

Consulte también

[Maximized\(\)](#)

## InStr()

Determina si una serie está insertada en otra.

### Sintaxis

InStr(<serie origen>, <serie de comparación>)

### Resultado

Instr() da como resultado la ubicación del primer carácter de la <serie de comparación> dentro de la <serie origen>. Si la <serie de comparación> no está dentro de la <serie origen>, el resultado de Instr() es -1.

### Descripción

Los números de ubicación de caracteres empiezan con 0.

Por ejemplo, a=Instr( "El perro de San Roque", "perro" ) da 3 en a.

## Left\$() and Right\$()

Copia un número especificado de caracteres a la derecha o izquierda de una serie. Utilice una variable de serie para contener los caracteres de resultado.

Left\$(<serie>, <número>)

Da como resultado una copia de tantos caracteres especificados en <número>, contando desde la izquierda en <serie>. Por ejemplo, a\$=Left\$( "Sígame", 4 ) da "Siga " en a\$.

Right\$(<serie>, <número>)

Da como resultado una copia de tantos caracteres especificados en <número>, contando desde la derecha en <serie>. Por ejemplo, a\$=Right\$( "Sígame", 2 ) devuelve "me" en a\$.

## **Length()**

Determina cuántos caracteres contiene una serie.

### **Sintaxis**

Length(<serie>)

### **Resultado**

El número de caracteres y espacios de la serie especificada.

### **Descripción**

Por ejemplo, si `a=Length( "Nunca digas nunca jamás" )`, `a` es 23.

## Lowercase\$() y Uppercase\$()

Fuerza a las series a pasar a mayúsculas o minúsculas.

Lowercase\$(<serie>)

Transforma los caracteres alfabéticos de la <serie> en minúsculas. Por ejemplo, a\$=Lowercase\$( "NUNCA" ) da como resultado "nunca" en a\$.

Uppercase\$(serie)

Transforma los caracteres alfabéticos de la <serie> en mayúsculas. Por ejemplo, a\$=UPPERCASE( "nunca" ) da como resultado "NUNCA" en a\$.

## **MakeDir()**

Crea un directorio.

### **Sintaxis**

MakeDir( <nombre del directorio> )

### **Resultado**

Da como resultado 1 si el directorio se ha creado y, en caso contrario, un 0.

### **Descripción**

Cuando cree un directorio, especifique la ruta completa.

Por ejemplo, a=MakeDir( "c:\windows\temp" ) da como resultado 1 en "a" si el directorio "c:\windows\temp" se ha creado. Si dicho directorio ya existía o no se puede crear por cualquier motivo, da como resultado 0.

---

Consulte también

[RemoveDir\(\)](#)

## Maximized()

Determina si una ventana específica está maximizada.

### Sintaxis

Maximized(<nombre de la ventana>)

### Resultado

Da como resultado 1 si la ventana especificada está maximizada ó 0 si no lo está o no existe.

### Descripción

Sírvase del <nombre de la ventana> que haya utilizado al anexar la ventana. Por ejemplo, si la ventana especificada del Bloc de notas ocupa toda la pantalla, Maximized( "NOTEPAD\_Edit1", 1 ) da como resultado 1.

---

Consulte también

[Minimized\(\)](#)

## MessageBox()

Muestra un cuadro de mensaje que contiene uno o más botones.

### Sintaxis

MessageBox( <título>, <texto>, <tipo de botón> )

### Resultado

Un número entero que indica el botón en el que se ha hecho clic ó 0 si no hay suficiente memoria para mostrar el cuadro de mensaje.

- 1 Aceptar
- 2 Cancelar
- 3 Interrumpir
- 4 Reintentar
- 5 Omitir
- 6 Sí
- 7 No

### Descripción

Muestra un cuadro de mensaje con el título especificado. Puede usar un cuadro de mensaje cuando no le haga falta un cuadro de diálogo entero.

El parámetro <texto> especifica el texto sito dentro del cuadro de mensaje. El texto se divide automáticamente en líneas dentro del cuadro.

El parámetro <tipo de botón> especifica el icono y botones del cuadro de mensaje. Añada los números enteros del icono y los botones para especificar el conjunto completo de un icono y varios botones. Por ejemplo, 17 especifica un icono de parada y los botones Aceptar y Cancelar.

<b>Botones</b>		<b>Iconos</b>	
0	Aceptar	16	Parada
1	Aceptar y Cancelar	32	Signo de interrogación final
2	Interrumpir, Reintentar e Ignorar	48	Signo de exclamación final
3	Sí, No y Cancelar	64	Información
4	Sí y No		
5	Reintentar y Cancelar		

### Especificador de botón predeterminado

- 256 Segundo botón
- 512 Tercer botón

## **MouseWindow\$( )**

Da como resultado el nombre de la ventana donde se encuentra el cursor del ratón.

### **Sintaxis**

MouseWindow\$( )

### **Resultado**

Facilita el nombre de la ventana en la que se halla el cursor del ratón. Utilice una variable de serie para que contenga los caracteres de resultado.

### **Descripción**

El nombre dado como resultado por MouseWindow\$( ) puede emplearse en el comando Attach. Por ejemplo, a\$ = MouseWindow\$( ) da "NOTEPAD\_Edit1" si el ratón está en el área de edición dentro de la aplicación Bloc de notas.

## PadString\$()

Llena una serie con un número específico de espacios.

### Sintaxis

```
PadString$( <serie>, <longitud> )
```

### Resultado

Da como resultado una serie en blanco , con la longitud especificada.

### Descripción

El principal objetivo de PadString\$() es agregar espacios a las series, con la finalidad de que después de escribir dichas series en archivo, aparezcan en forma de tabla al leerlas.

### Ejemplo

Los siguientes fragmentos de código muestran varias posibilidades de añadir espacios a una variable de serie, a fin de que alcance la longitud necesaria antes de escribirla en un archivo. El script se encarga de verificar que la longitud de a\$ sea menor o igual que 20. Si es mayor que 20, puede truncarla con la función Left\$().

```
;serie original  
a$="Manufacturer"  
;calcula el número de espacios que se necesitan  
b=20- Length(a$)  
;crea la serie string b$ que contiene los espacios que han de agregarse  
b$=Padstring$(b$,b)  
;concatena las series originales y las llena de espacios  
write$=a$ + b$
```

Si se sirve de este método para acumular datos introducidos por el usuario, puede almacenarlos en variables separadas y utilizar la variable en lugar de una serie literal.

---

Consulte también

[Left\\$\(\)](#)

## PromptBox()

Crea un cuadro de edición sencillo con título que solicita al usuario la introducción de una serie.

### Sintaxis

```
PromptBox( <título>, <solicitud>, <resultado$>)
```

### Resultado

Los datos introducidos por el usuario se almacenan en <resultado\$>. Si el usuario selecciona Aceptar, PromptBox() da como resultado un 1. Si, por el contrario, hace clic en cualquier otro botón, el resultado es un 0 y no se almacena ningún texto en <resultado\$>.

### Descripción

Puede utilizar PromptBox() para permitir la introducción de datos al usuario en aquellas ocasiones en las que no es necesario emplear un cuadro de diálogo entero.

El parámetro <solicitud> especifica la petición formulada al usuario y los datos que éste introduce se almacenan en <resultado\$>. Por ejemplo, a=PromptBox( "Nombre", "Escriba cómo se llama:", nombre\$ ) da como resultado el texto introducido en "nombre\$" y 1 en "a" si el usuario hace clic en Aceptar.

## Random( )

Da como resultado un número seudoaleatorio comprendido entre 0 y 32767.

Por ejemplo, `a=Random()` puede dar como resultado 4292 en "a".

## ReadIni() y ReadIni\$()

ReadIni() lee un valor numérico en un archivo .INI Windows. ReadIni\$() lee un valor de serie.

### Sintaxis

ReadIni( <archivo .INI>, <nombre de la sección>, <nombre clave>, <predeterminado> )

ReadIni\$( <archivo .INI>, <nombre de la sección>, <nombre clave>, <predeterminado\$> )

### Resultado

ReadIni() da como resultado el número leído. Si la entrada especificada por los parámetros <nombre de la sección> y <nombre clave> no existe en el archivo, da como resultado el valor <predeterminado> que usted especifique.

ReadIni\$() da como resultado la serie definida en el enunciado de asignación <nombre clave> o, si no existe la entrada especificada, la serie <predeterminada>.

### Descripción

Sírvase de ReadIni() y ReadIni\$() para leer cualquier archivo .INI de Windows que utilice las normas estándar de sección y nombre clave. La norma respecto a un nombre de sección es representarla como una serie entre corchetes: [nombre de la sección] y el nombre clave como una serie seguida del signo de igual: (=).

Por ejemplo, a\$=ReadIni( "win.ini", "windows", "coolswitch", 0 ) da como resultado el número que denota la configuración de "coolswitch" en WIn.InI (ó 0 si no existe la entrada).

a\$=ReadIni\$( "wnuser.ini", "ScriptTools.CodeGeneration", "Shell1", "none" ) da como resultado el nombre del programa shell en la variable a\$. Si no existe la entrada, da como resultado "none" (nada).

## **RemoveDir()**

Borra un directorio.

### **Sintaxis**

RemoveDir( <nombre del directorio> )

### **Resultado**

Un 1 si la función ha sido satisfactoria ó 0 en caso contrario.

### **Descripción**

Borra un directorio vacío. No es posible borrar el directorio activo ni el raíz.

Por ejemplo, a=RemoveDir( "c:\temp" ) da 1 en "a" si se borra el directorio. Si C:\TEMP contiene archivos, no puede borrarse el archivo y RemoveDir() da un 0.

## Running()

Determina si está ejecutándose un programa en ese momento.

### Sintaxis

Running(<nombre del programa>)

### Resultado

Da un 1 si el programa especificado se está ejecutando y un 0 en caso contrario.

### Descripción

Emplee Running() para averiguar si ya existe una instancia de un programa antes de anexarse a él.

Por ejemplo, a=Running( "NOTEPAD.EXE" ) da 1 en a si el Bloc de notas está ejecutándose o cargándose y 0 si no lo está. Compruebe el valor de a para averiguar si el Bloc de notas está ya ejecutándose, de forma que puede anexarse a la instancia ya existente del mismo, en lugar de iniciar una nueva.

Este comando aparece explicado en contexto en el segundo [ejemplo pormenorizado](#) del comando Attach.

## **SubStr\$()**

Da como resultado una sección especificada de una serie definida.

### **Sintaxis**

SubStr\$( <serie origen>, <posición inicial>, <posición final> )

### **Resultado**

Da como resultado el segmento especificado de la serie o en blanco si no resulta satisfactorio.

### **Descripción**

Da como resultado los caracteres de <serie origen>, empezando por el carácter en <posición inicial> y continuando hasta el carácter en <posición final>.

El primer carácter de la serie origen es el carácter número 0. Por ejemplo, a\$=SubStr\$( "abcdef", 1, 3) da "bcd" en a\$.

### **TopInstance() y TopWindow\$()**

Da como resultado el número de instancia de la ventana superior, que es normalmente la ventana activa. TopWindow\$() da el nombre Attach de la ventana.

TopInstance() Devuelve el número de instancia de la ventana superior. Por ejemplo, a=TopInstance() da 2 si se están ejecutando dos instancias de NOTEPAD.EXE y la segunda es la activa.

TopWindow\$() Da como resultado el nombre que haya utilizado con el comando Attach para la ventana superior. Por ejemplo, a\$=TopWindow\$() puede devolver "NOTEPAD\_Edit1" en a\$.

## Version() y Version\$()

Ambas funciones dan como resultado el número de versión de ScriptTools.

Version( )            Da el número de versión de ScriptTools para Windows, en forma de número entero.  
Por ejemplo, a=Version() puede devolver 100.

Version\$( )            Da como resultado el número de versión de ScriptTools para Windows en forma de serie. Por ejemplo, a\$=Version\$() daría "1.0".

## WindowExists()

Determina si existe una ventana específica.

### Sintaxis

WindowExists( <nombre de la ventana>, <instancia> )

### Resultado

Da 1 si existe el <nombre de la ventana> especificado con el número de <instancia>. Si dicha ventana no existe, da un 0.

### Descripción

Utilice WindowExists() para verificar la existencia de una ventana antes de proceder a realizar una operación en ella. Por ejemplo, a=WindowExists("NOTEPAD\_Edit1", 1) da 1 si la ventana especificada existe.

Este comando aparece explicado en contexto en el segundo [ejemplo pormenorizado](#) del comando Attach.

## Writeln() y Writeln\$()

Las funciones Writeln() y Writeln\$() escriben datos, definidos por los parámetros pasados a la función, en el archivo .INI Windows especificado.

### Sintaxis

```
Writeln( <archivo .INI>, <nombre de la sección>, <nombre clave>, <número entero> )  
Writeln$( <archivo .INI>, <nombre de la sección>, <nombre clave>, <serie$> )
```

### Resultado

Writeln() da 1 si la escritura es satisfactoria y, en caso contrario, 0. Writeln\$() no da ningún valor como resultado.

### Descripción

Writeln() escribe en un archivo .INI un valor numérico . Writeln\$() escribe un valor de serie. Es posible escribir en cualquier archivo .INI de Windows que utilice las normas estándar de sección y nombre clave. La norma respecto a un nombre de sección es representarla como una serie entre corchetes: [nombre de la sección] y el nombre clave como una serie seguida del signo de igual: (=).

Por ejemplo, para cambiar el nombre del usuario almacenado en WNUSER.INI a Pedro Herrero, introduzca la siguiente línea de código:

```
Writeln$( "wnuser.ini", "UserInformation", "UserName", "Pedro Herrero")
```

## Funciones de fecha y hora

Muchas de las funciones existentes dan como resultado todo o parte del sistema actual de fecha y hora. Consulte esta lista para decidir cuál es la función que más conviene a su programa.

AmPm\$( )	Da "AM" o "PM", según corresponda. Por ejemplo, a\$=AmPm\$( ) da "AM" si es antes del mediodía o "PM" si es después.
Date\$( )	Devuelve la fecha en el formato especificado por la variable de sistema <u>_DateMode</u> . Por ejemplo, a\$=Date\$( ) puede dar "25 enero 1992" en a\$. El formato de la fecha viene determinado por la configuración de <u>_DateMode</u> .
Day\$( )	Da como resultado el día de hoy en forma de serie, lunes,...domingo. Por ejemplo, a\$=Day\$( ) daría "lunes" en a\$. El formato del día viene determinado por la configuración de <u>_DayMode</u> .
Hours( )	Da la hora del sistema, en formato de 24 horas. Por ejemplo, a=Hours( ) da 10 en a.
Mins( )	Presenta los minutos de la hora. Por ejemplo, a=Mins( ) daría 42.
Month\$( )	Facilita el nombre del mes actual. Por ejemplo, a\$=Month\$( ) podría ser "marzo" en a\$.
Secs( )	Da los segundos de la hora. Por ejemplo, a=Secs( ) puede ser 37.
Time( )	Da la hora y los minutos en formato de 24 horas, sin puntos. Por ejemplo, a=Time( ) da 1341 cuando son las 13:41.
Time\$( )	Presenta la hora en formato de 24 horas, a modo de serie, incluyendo puntuación. Por ejemplo, a\$=Time\$( ) da "13:41:23" en a\$ cuando son las 13:41:23.
Weekday( )	Facilita un número para el día de la semana, 0 = domingo, ..., 6 = sábado. Por ejemplo, si hoy es lunes, a=Weekday( ) da 1.
Weekday\$( )	Da el nombre del día de la semana. Por ejemplo, a\$=WeekDay\$( ) podría dar "miércoles" en a\$.
Year( )	La función Year( ) da como resultado los cuatro dígitos del año corriente. Por ejemplo, a=Year( ) podría dar 1993.

## **Variables de sistema de ScriptTools.**

Sírvase de las siguientes variables de sistema para obtener información acerca de cómo ejecutar un script o para influir en el funcionamiento del mismo.

ActiveWindow

AttachWindow

AttachMouseX, AttachMouseY

DateMode, DayMode

EOF

MouseX, MouseY

Path\$

PauseMode

ReadPos, WritePos

ReadFile\$, WriteFile\$

## **\_ActiveWindow**

Almacena el identificador de la ventana activa. En ocasiones se utiliza al hacer llamadas DLL.  
Es una variable de sólo lectura.

## **\_AttachWindow**

Almacena el identificador de la ventana anexa en ese momento.

Es posible configurar esta variable, pero cerciórese de utilizar un valor válido. Normalmente se utiliza el valor de `_AttachWindow` previamente almacenado en una variable creada por usted mismo.

## **\_AttachMouseX, \_AttachMouseY**

Las variables de sistema `_AttachMouseX` y `_AttachMouseY` de sólo lectura facilitan la posición del cursor del ratón relativa a la ventana anexa.

- ▶ Para descubrir el nombre de la ventana anexa, utilice la función [MouseWindow\(\)](#).

## \_DateMode, \_DayMode

\_DateMode y \_DayMode determinan el formato de presentación de las fechas y días de la semana. Emplee los números enteros de los siguientes ejemplos para especificar el formato que desea utilizar.

\_DateMode determina el formato de fechas resultado de Date\$().

Ejemplo Navidad 1991

_DateMode = 0	Date\$ ( ) da 25 diciembre 1991(predeterminado)
_DateMode = 1	Date\$ ( ) da 25-12-1991
_DateMode = 2	Date\$ ( ) da 12-25-1991

\_DayMode determina el formato de los días de la semana resultado de WeekDay\$()

Ejemplo

_DayMode = 0	WeekDay\$() = "martes"
_DayMode = 1	WeekDay\$() = "mar"

Consulte también

WeekDay\$()

Date\$()

## **\_EOF**

ScriptTools configura `_EOF` en 1 si el último carácter leído de un archivo era el último del mismo.  
`_EOF` es una variable de sólo lectura.

Consulte también

[\\_ReadPos](#)

## **\_MouseX, \_MouseY**

Las variables `_MouseX` y `_MouseY` se configuran en las coordenadas X e Y actuales del cursor del ratón.

`_MouseX` y `_MouseY` son variables de sólo lectura.

## **\_Path\$**

Almacena la ruta predeterminada actual para leer y escribir archivos.

\_Path\$ es una variable de sólo lectura.

▶ Al usar el comando ChangeDir para cambiar el directorio predeterminado, también se cambia el valor de \_Path\$.

## **\_PauseMode**

Configure `_PauseMode` en 0 si desea hacer caso omiso de los enunciados `Pause`, de manera que el programa se ejecute lo más rápidamente posible.

Sin embargo, si ha insertado pausas para que se puedan leer los paneles de texto, los mensajes de los mismos aparecerán en pantalla solamente durante unos breves instantes.

---

Consulte también

[Pause](#)

## **\_ReadPos, \_WritePos**

\_ReadPos almacena la posición del archivo tras la última lectura del mismo, mientras que \_WritePos almacena la posición tras la última escritura.

Son variables Long. Declare un tipo de variable Long si desea guardar el valor de \_ReadPos o \_WritePos.

---

Consulte también

Long

Read

Write

## **\_ReadFile\$, \_WriteFile\$**

\_ReadFile\$ y \_WriteFile\$ contienen la ruta completa de los archivos de entrada y salida actuales.

Son variables de sólo lectura. Para cambiar el archivo de entrada y salida actual, utilice los comandos Read o Write para abrir un archivo distinto.

## **Nombres de variables**

Los nombres de las variables pueden tener hasta 16 caracteres. No distinguen entre mayúsculas y minúsculas.

El primer carácter tiene que ser alfabético (a...z), pero los siguientes pueden ser letras, números o el signo de subrayado. Los nombres de las variables de serie tienen que terminar en el símbolo \$. No se permiten otro tipo de caracteres.

No está permitido el uso de una palabra reservada como nombre de variable.

---

Consulte también

[Palabras reservadas](#)

## Tipos de datos

ScriptTools utiliza los tipos de datos enumerados en la lista. Sólo hay que declarar los números enteros largos y las matrices antes de utilizarlos.

- Números enteros** Los números enteros comprendidos entre -32768 y +32767. Las variables de este tipo no tienen que declararse antes de usarse.  
Ejemplo: `testno = 145`  
Los números hexadecimales se introducen entre corchetes con x como prefijo del número, como `{x3b}`. Al asignar un número hexadecimal a una variable, se le evalúa como un número decimal. Por ejemplo, si asigna `{xffff}` a una variable de número entero, se la evalúa como -1.
- Números enteros largos** Los números enteros largos están comprendidos entre -2.147.483.648 y +2.147.483.647. Use el enunciado de lenguaje Long para declarar las variables largas antes de utilizarlas. Los valores de Long introducidos como literales se distinguen por el sufijo 'L' mayúscula o minúscula.  
Ejemplo: `Long testffff`  
`testffff = 65535L`
- Series** Las series pueden tener un máximo de 32766 caracteres. Al definir una variable de serie en el script, escríbala entre comillas dobles (""). Si tiene que utilizar comillas dobles dentro de una serie, utilice las sencillas (') para encerrar la serie entera. Puede incluir caracteres especiales en las series, siempre que estén encerrados en corchetes ({}). Los nombres de las variables tienen que acabar con el carácter \$.  
Ejemplo: `SayHello$ = "Hola a todos {return}"`  
El [ejemplo](#) de matrices y series incluye un grupo de ejemplos de series.
- Matrices** Utilice el enunciado de lenguaje Array para declarar una matriz antes de usarla. Las matrices se distinguen por los corchetes en los que está encerrada []. Puede ser de tipo de número entero o de serie. Es posible acceder a los elementos de una matriz por separado. Las matrices pueden asignarse a otras matrices, como muestra el [ejemplo](#).

Consulte también

[Variables de sistema](#)

## Operadores

Emplee los siguientes operadores para los números enteros.

La lista está clasificada en orden de precedencia de evaluación. Las expresiones con el mismo orden de precedencia se evalúan de izquierda a derecha. Los paréntesis permiten varios niveles de expresión, evaluándose primero el más alto o el más interno.

### Nota

Las expresiones booleanas pueden evaluarse exclusivamente como argumento de la construcción If...Then...EndIf. Por ejemplo, "If (a=3 OR b=7)..." es correcto, pero "x=(a=3 OR b=7)" no.

Tipos	Símbolos	Ejemplos
Paréntesis	( ), [ ]	
Operadores unarios	NOT, !, +, -, ~	NOT    a = ~a NEG    a = -a
Expresiones aritméticas	*, /, %, +, -	MOD    a = a % 5 DIV    a = a / 5 MUL    a = a * 5 SUB    a = a - 5 ADD    a = a + 5
Operadores Bitwise	&,  , ^	XOR    a = a ^ 5 OR     a = a   5 AND    a = a & 5
Comparaciones	LT, <, LE, <=, GT, >, GE, >=, EQ, =, NE, !=, <>	
Booleanas	AND, &&, OR,   , XOR, ^^	AND    if a=1 && b=2 then... OR     if a=1    a=2 then... XOR    if a=1 ^^ b=2 then...
Asignación	EQ, =	

## Ejemplos de matrices y series

Estos ejemplos muestran cómo utilizar matrices y series con distintos propósitos.

Array a[15]	;declara la matriz a con 15 elementos
a[10] = 14	;asigna 14 al elemento 10 de la matriz a
b\$[0] = "Hola a todos"	;asigna la serie "Hola a todos" al elemento 0 de la matriz b\$[]
a[ ] = b[ ]	;asigna la matriz b a la matriz a
a\$="Buenos días"	;asigna la serie "Buenos días" a a\$
b\$="Clase"	;asigna la serie "Clase" a b\$
q\$=", "	;asigna una coma y un espacio a q\$
z\$=a\$ + q\$ + b\$	;concatena tres series para hacer una sola

## Palabras reservadas

No pueden usarse palabras reservadas como nombres de variables'.

Entre las palabras reservadas de ScriptTools se encuentran las siguientes:

▶ El nombre de cualquier comando, variable de sistema, función o palabra opcional que forme parte de la sintaxis de un comando o función. Se facilita una tabla con las palabras opcionales.

▶ Los nombres de los meses o días de la semana.

▶ Cualquier operador, como NE, GE, GT, LE, LT, AND, OR, XOR, EQ y NOT.

Las palabras reservadas de diverso tipo se encuentran enumeradas en la tabla siguiente. **No** incluye comandos, funciones ni variables de sistema.

activate	dialog	into	rectangle
activewindow	dosset	item	return
all	double	left	right
ampm	eq	length	screen
any	every	line	sec
array	exclamation	local	second
at	for	long	seconds
black	from	many	secs
button	function	menu	shift
buttonclick	ge	message	singleclick
by	go	middle	size
child	gray	min	sorted
chr	grey	mins	tick
comma	holding	module	ticks
control	horz	month	time
controlshift	hours	none	to
crlf	hscroll	on	until
date	icon	one	up
day	id	onerror	vert
deactivate	in	otherwise	vscroll
debugger	infocus	page	with
default	information	parent	wscroll
delimited	insert	password	xor
destroy	inside	question	



## Personalizar cuadros de diálogo

Personaliza las funciones siguientes:

Barra de herramientas Comandos de la Barra de herramientas.

Menú Comandos de los menús.

Teclado Teclas de método abreviado para comandos.

Comandos Comandos personalizados que inician otras aplicaciones o ejecutan programas.

Los cuadros de diálogo pueden contener cualquiera de los botones siguientes. Haga clic en el nombre del botón para ver lo que hace.

Botones Guardar Predeterminado Copiar

## Barra de herramientas

Crea una nueva serie de comandos en la Barra de herramientas, o cambia uno que ya existe en ella. (Tenga presente que las carpetas del Escritorio no tienen barras de herramientas.

- ▶ Para cambiar el aspecto de los botones de la barra, elija el botón Estilo y seleccione un estilo de las opciones que se presentan.
- ▶ Para añadir un comando a la Barra de herramientas, seleccione éste en el cuadro de lista desplegable de Comandos y elija Insertar.

### Opciones del cuadro de diálogo

Esquema Haga clic en la flecha que hay a la derecha del cuadro de la lista para ver los esquemas de la Barra de herramientas que usted ha grabado. El nombre del esquema actual aparece como predeterminado. Los nombres de los esquemas hacen diferencia entre mayúsculas y minúsculas. Por ejemplo, un esquema llamado Proyectos no es igual que otro llamado PROYECTOS.

Barra de herramientas Una representación de la Barra de herramientas activa.

Comandos Este cuadro de lista desplegable presenta los nombres de los comandos que se pueden añadir a la Barra de herramientas. La lista incluye todos los comandos personalizados que se han definido.

Botones Copiar Quitar  
Insertar Eliminar  
Flechas Predeterminado  
Estilo

### Nota

Utilice la Barra de herramientas únicamente para comandos que realicen tareas. Algunos comandos de los que aparecen en el cuadro de lista no realizan acciones, sino que muestran posibilidades de acción, como por ejemplo el comando Visor de formatos. Estos comandos no son apropiados para la Barra de herramientas.

## Estilo de la Caja de herramientas

Muestra en la ventana los botones de herramientas en estilos y lugares diferentes .

### Opciones del cuadro de diálogo

- Estilo de botón      Seleccione uno de los tres estilos, Icono y Texto, Solo texto o Sólo icono. Cuando se selecciona un estilo, el botón de comando de ejemplo muestra cómo queda éste.
- Ubicación            En muchas de las aplicaciones PC Tools for Windows, la Barra de herramientas puede ponerse en la parte superior de la ventana (la posición predeterminada), en la parte inferior o en el lado izquierdo o derecho. En las aplicaciones donde se necesita una ventana larga, puede ser más conveniente que los botones estén en un lateral de la ventana.

**Nota:** El comando Ubicación no está disponible en todas las aplicaciones PC Tools for Windows.

## Estilo del botón

Seleccione uno de los tres estilos de botón de herramienta.

Icono y texto      Muestra el icono y el nombre del comando.

Sólo texto          Muestra sólo el nombre del comando.

Sólo icono          Muestra sólo el icono del comando.

## Posición de la Barra de herramientas

En muchas aplicaciones de PC Tools for Windows la Barra de herramientas se puede poner en la ventana en cualquiera de las cuatro posiciones posibles.

Arriba	Horizontalmente, a lo largo de la parte superior de la ventana, bajo la barra de menús.
Abajo	Horizontalmente, a lo largo de la parte inferior de la ventana, encima de la línea de estado.
Izquierda	Verticalmente, en el borde izquierdo de la ventana.
Derecha	Verticalmente, en el borde derecho de la ventana.

## Teclado

Añade o suprime teclas abreviadas de teclado para comandos.

▶ Para cambiar la tecla abreviada asociada a un comando, seleccione el comando, introduzca una nueva tecla abreviada y elija Insertar o Quitar. (Quitar aparece si la tecla abreviada ya está asignada a otro comando).

### Opciones del cuadro de diálogo

**Esquema** Haga clic en la flecha que hay a la derecha del cuadro de lista desplegable para ver los esquemas de teclado que se han definido. Los nombres de los esquemas hacen diferencia entre mayúsculas y minúsculas. Por ejemplo, un esquema cuyo nombre es Juan no es igual que otro esquema llamado JUAN.

**Comandos y Teclas de método abreviado**

Una lista de comandos y teclas abreviadas asociadas a éstos.

**Tecla de método abreviado** Los nombres de las teclas se introducen automáticamente en el cuadro al pulsar éstas. Si introduce una tecla abreviada que ya se está usando, se sustituye el comando que tenía asignado esa tecla.

**Comandos** Esta lista muestra los nombres de los comandos a los que se puede asignar teclas de método abreviado. Esta lista incluye cualquier comando personalizado que se haya definido.

**Botones** Reemplazar o Insertar Eliminar  
Predeterminado

## **Esquemas de teclado**

Se pueden crear esquemas de teclado con nombre para ocasiones especiales.

Si elige Predeterminado, el esquema especificado se reinicializa a los valores predeterminados del teclado.

## Menú

Cambia el menú.

- ▶ Para añadir un comando al menú, al nivel del elemento de menú seleccionado, siga estos pasos:
  1. Desplácese a lo largo del cuadro de la lista Comandos.
  2. Seleccione el comando que desea añadir.
  3. Elija Insertar.

### Opciones del cuadro de diálogo

<u>Esquema</u>	Haga clic en la flecha que hay a la derecha del cuadro de lista para ver los esquemas de menú que hay definidos. Los nombres de los esquemas hacen diferencia entre mayúsculas y minúsculas. Por ejemplo, un esquema llamado María no es igual que otro llamado MARIA.
Menú	La ventana Menú muestra el menú seleccionado en ese momento. Se pueden eliminar y añadir nuevos comandos a cualquier nivel. Seleccione el <u>Separador</u> de comandos para insertar una barra separadora.
Comandos	Este cuadro de lista desplegable muestra los nombres de los comandos que se pueden añadir al menú. La lista incluye todos los comandos personalizados que hay definidos.
Botones	<u>Copiar</u> <u>Quitar</u> <u>Insertar</u> <u>Eliminar</u> <u>Predeterminado</u> 



### Nota

Elija el botón Predeterminado únicamente si quiere hacer corresponder el esquema de menú seleccionado con el esquema de menú predeterminado.

Véase también

[Editar nombres de comando](#)

[Sobre editar menús](#)

## Sobre editar esquemas de menú

En varias de las aplicaciones de PC Tools for Windows se puede seleccionar un esquema Menú largo o un esquema Menú corto. En el Administrador de archivos, el esquema de menú Corto es una copia exacta del esquema de menú del Administrador de archivos de Microsoft Windows. Y además de los menús corto y largo, el Escritorio tiene un esquema de Menú personalizado.

Un esquema de menú puede modificarse por las siguientes razones:

- ▶ Eliminar los comandos que no se usan.
- ▶ Añadir una nueva categoría de menú.
- ▶ Añadir comandos personalizados que haya creado uno mismo.
- ▶ Cambiar el orden de los comandos de un menú.
- ▶ Transferir comandos de un menú a otro.
- ▶ Añadir barras de separación.

## Editar nombres de comandos de menú

Es muy probable que en alguna ocasión tenga que editar los nombres de los comandos que usted ha creado. Por ejemplo, pudiera ser necesario editar una cabecera de nivel de menú que ha añadido, y darle un nombre único.

▶ Para cambiar el nombre de un comando del menú, seleccione el comando y después pulse F2 para abrir el cuadro de edición. Escriba el nombre que desea asignar al comando y pulse Intro.

▶ Para crear un carácter de subrayado para usarlo como tecla abreviada, inserte un signo ampersand (&) delante de la letra que desea usar. Asegúrese de que la tecla abreviada que ha creado no produce ningún conflicto con otras teclas abreviadas. Si quiere incluir un carácter ampersand literal en el nombre del comando, inserte dos ampersand.

Por ejemplo,

Mi&Comando produce MiComando.

Mi&&Comando produce Me&Comando.

## Esquemas de menú predeterminados

Todas las aplicaciones de PC Tools For Windows tienen un esquema de menú predeterminado. Cuando elige el botón Predeterminado en el cuadro de diálogo Personalizar menús, el menú seleccionado se reinicializa y queda como el menú predeterminado.

Si con la aplicación se incluye más de un esquema de menú, a continuación se enumera el menú que se utiliza para reinicializar un esquema a su estado predeterminado.

<b>Aplicación</b>	<b>Esquema de menú predeterminado</b>
Escritorio	Menú corto
Administrador de archivos	Menú corto
SmartFind	Menú corto
Editor de iconos	Menú largo

## Esquemas de menú

Los esquemas de menú disponibles aparecen en el menú Opciones.

Varias aplicaciones de PC Tools for Windows ofrecen menús largos y cortos predeterminados. Por ejemplo, el Administrador de Archivos tiene una opción Menú corto que es exactamente igual al menú del Administrador de archivos de Windows y una opción Menú corto que contiene todos los comandos del Administrador de archivos de PC Tools for Windows.

Además de los menús largos y cortos, el Escritorio tiene un esquema de menú llamado Menú personalizado que es idéntico al esquema de menú Corto, salvo que añade un menú especial que contiene aplicaciones PC Tools for Windows seleccionadas.

Si ha creado esquemas de menú personalizado, éstos están disponibles desde el menú Opciones.

- ▶ Para cambiar a otro esquema de menú, elija el que desee en el menú Opciones.

## Separador

Para añadir una línea que divida un grupo de comandos de un menú de otro grupo, seleccione el comando Separador en la lista de nombres de comando que se muestra en el cuadro lista.

La línea aparece encima del comando seleccionado cuando se elige Insertar.

## **Esquemas de Barra de herramientas**

La lista de esquemas de Barra de herramientas aparece en el menú pop-up de la Barra de herramientas.

Si ha creado esquemas de Barra de herramientas personalizados, los nombres de éstos también aparecerán en la lista.

- ▶ Para cambiar a otro esquema de Barra de herramientas, elija éste en el menú pop-up.

## **Comandos personalizados**

Usted puede añadir sus propios comandos personalizados para aplicaciones PC Tools for Windows.

Los comandos personalizados no tienen ayuda en línea a nivel de menú. Si el comando es un programa Windows, se puede usar la ayuda de Microsoft Windows después de elegir el comando para iniciar el programa.

Si el comando es una aplicación DOS o un programa ScriptTools, es probable que no haya ayuda.

## Crear comandos personalizados

Añade comandos a la lista disponible de un menú y de la Barra de herramientas. Un comando puede ser cualquiera de los programas o documentos siguientes:

- ▶ Otra aplicación Windows, como por ejemplo, Word for Windows.
- ▶ Un programa RUN ScriptTools.
- ▶ Una aplicación DOS que se ejecuta en una ventana DOS.
- ▶ Un documento asociado con una aplicación.

Para usar un comando personalizado después de creado, es necesario asignarlo a un menú, a la Barra de herramientas o a una tecla abreviada.

### Opciones del cuadro de diálogo

Comandos	El cuadro Comandos muestra una lista con los comandos personalizados que se han añadido.
Nombre de comando	El nombre que se asigna al comando.
Línea de comando	La ruta completa del documento de comando o del programa de aplicación. Utilice el botón Examinar para buscar el archivo e insertar su nombre. Si está definiendo comandos personalizados en el Administrador de archivos, puede insertar un <u>parámetro reemplazable</u> en la línea de comando para que actúe sobre un archivo seleccionado, una unidad o un directorio que aparezcan en la ventana del árbol o de la lista de archivos
Directorio activo	El directorio donde el comando busca los archivos de datos y otros documentos. Esta información es opcional. Si deja el cuadro del directorio activo en blanco al crear un nuevo comando, el directorio predeterminado es aquel donde se encuentra el programa. Si elimina la información del directorio activo, entonces el directorio predeterminado es el de Windows.
Descripción	Un comando descriptivo que aparece en la barra de estado.
Mostrar estilo	El tamaño de la ventana donde se ejecuta el comando: normal, minimizado o maximizado.
Texto de botón	Un título corto que aparece en el botón correspondiente al comando.
Texto de menú	Un título corto que aparece en el listado de menú correspondiente al comando.
Botones	<u>Nuevo</u> <u>Copiar</u> <u>Eliminar</u> <u>Examinar</u> <u>Icono</u> <u>Contraseña</u>

## Parámetros reemplazables en la línea de comando

Inserte uno de los nombres de opción que se indican a continuación, para introducir la unidad, directorio o archivo seleccionado como opción de línea de comando para un comando personalizado.

Por ejemplo, se podría usar %FILE% como opción de línea de comando para un editor de texto (EDITOR.EXE %FILE%), o %SELECTION% para una aplicación que pueda aceptar un archivo, un directorio o una unidad, como por ejemplo Central Point Backup (CPBACKUP.EXE %SELECTION%).

Si especifica	La aplicación usa
%FILE%	El nombre del archivo seleccionado.
%DIR%	El nombre del directorio seleccionado.
%DRIVE%	La letra de unidad lógica seleccionada.
%SELECTION%	Cualquier elemento seleccionado.

### Nota

Estos parámetros sólo son válidos para comandos personalizados creados en el Administrador de archivos.

## Cuadro de diálogo Seleccionar icono

Selecciona un icono para el comando que se ha creado.

Se puede seleccionar un icono de cualquier archivo que contenga iconos en el formato estándar, como por ejemplo, MORICONS.DLL. También se puede seleccionar un icono que usted mismo ha creado con el Editor de iconos.

▶ Para usar un archivo de iconos diferente, elija Examinar y use el árbol de directorios y la lista de archivos para seleccionar el archivo de iconos deseado. Después elija Aceptar para aplicarlo al comando personalizado.

▶ Para seleccionar un nuevo icono, seleccione éste en el cuadro de selección y elija Aceptar.

### Opciones del cuadro de diálogo

Archivo actual El nombre del programa ejecutable para el que se va a seleccionar un icono.

Total de iconos El número de iconos que hay en el archivo seleccionado. Los iconos se muestran en el cuadro de selección. Utilice la barra de desplazamiento que hay en la parte inferior del cuadro para ver todos los iconos de que se dispone.

### Disponible si se elije el botón Examinar:

Nombre de archivo Los nombres de los archivos de iconos que hay en el directorio activo.

Directorios El árbol de directorios de la unidad activa.

Mostrar lista de archivos de tipos El tipo del archivo que se muestra. Seleccione en la lista de tipos de archivos que contienen iconos.

## **Archivos de icono**

Los archivos que contienen iconos tienen los siguientes tipos de archivo: DLL, EXE, ICO, RSL y ICL.

**Guardar**

Guardar todas las configuraciones nuevas o modificadas para esta categoría de Selecciones.

## Copiar

Hace una copia del esquema especificado, para así poder modificarlo y crear uno nuevo.

## Copiar

Hace una copia del comando personalizado seleccionado.

La copia puede modificarse editando la línea de comando o las opciones de los cuadros de texto para crear una versión diferente del comando.

## Predeterminado

Vuelve a poner el esquema especificado a sus valores predeterminados, incluso si éste ha sido grabado. Se pierden todos los cambios realizados.

Si con la aplicación se incluye más de un esquema de menú, a continuación se enumera el menú que se utiliza para reinicializar un esquema a su estado predeterminado.

<b>Aplicación</b>	<b>Esquema de menú predeterminado</b>
Escritorio	Menú corto
Administrador de archivos	Menú corto
SmartFind	Menú corto
Editor de iconos	Menú largo

## Contraseña

Introduzca una contraseña para este comando. Una vez asignada una contraseña a un comando, cada vez que lo elija se le pedirá que la introduzca. También deberá introducir la contraseña para cambiar cualquiera de los atributos de comando del cuadro de diálogo Comandos personalizados.

## Insertar

Inserta el comando que aparece en el cuadro Comandos, en la Barra de herramientas, menú o lista de teclas abreviadas del teclado en el esquema especificado. El comando se sitúa a la izquierda del elemento seleccionado o encima de éste.

En el cuadro de diálogo Teclado, el botón Insertar permanece inactivo hasta que se especifica una tecla abreviada.

## Eliminar

Elimina el elemento seleccionado de la Barra de herramientas, menú o lista de teclas abreviadas de teclado.

**Eliminar**

Elimina el comando personalizado que se ha seleccionado.

## Nuevo

Añade un nuevo esquema de configuración especificado.

Cuando se selecciona Nuevo, aparece un nombre predeterminado seleccionado en el cuadro Esquema. Para introducir un nombre diferente, escríbalo encima del actual.

## Nuevo

Visualiza la plantilla de comando genérico.

Puede introducir toda la información necesaria usted mismo o elegir Examinar para seleccionar un archivo .PIF, .RUN, .EXE o cualquier otro archivo ejecutable. Cuando selecciona el archivo, el Nombre de comando, Línea de comando, Descripción, los cuadros Texto de botón y Texto de menú se rellenan automáticamente con los parámetros predeterminados para el archivo ejecutable. Estos parámetros predeterminados se pueden editar para hacerlos más descriptivos.

### **Reemplazar o Insertar**

Este botón tiene el rótulo Insertar si la tecla abreviada que ha introducido no está asignada a otro comando.

El botón tiene el rótulo Reemplazar si la tecla abreviada que ha introducido ya está asignada a otro comando. No aparece ningún mensaje de confirmación.

## Quitar

Elimina la configuración completa especificada que se muestra en el cuadro de texto Esquema.

## **Menú Nuevo**

Añade un nuevo menú al nivel del cuadro seleccionado.

El nombre predeterminado para el nuevo menú es Menú personalizado.

Inmediatamente después de crear el nuevo elemento de menú, puede introducir un nombre descriptivo para reemplazar el de Menú personalizado. Si más tarde quiere cambiar el nombre, seleccione el nombre de menú y después pulse F2 para abrir el cuadro de edición.

Si se selecciona un nombre de menú, se inserta un nuevo menú encima. Si se selecciona un comando de menú, se inserta un submenú en cascada, al que se accede mediante el comando.

## **Flechas**

Haga clic en las teclas de flecha para mover el elemento seleccionado hacia la derecha o izquierda en la Barra de herramientas o arriba o abajo en la jerarquía de lista de menú.

## Icono

El icono aparece en el botón si se pone el comando en la Barra de herramientas.

Se puede seleccionar cualquier icono que haya disponible en la unidad, incluyendo los iconos que haya creado usted mismo con el Editor de iconos. Elija Examinar en el cuadro de diálogo Icono para buscar archivos de iconos y aplíquelos al nuevo comando.

### **Contraseña para los cuadros de diálogo Personalizar**

Si va a usar PC Tools for Windows desde una red, el administrador de red puede tener un conjunto de contraseñas en el cuadro de diálogo Personalizar para evitar que otros usuarios puedan cambiar la Caja de herramientas, menús, teclas abreviadas o lista de comandos personalizados.

Si se encuentra con que no puede usar los cuadros de diálogo Personalizar porque están protegidos con contraseña, consulte al administrador de la red.

## Examinar

Selecciona una unidad, un directorio y un archivo desde el cuadro de diálogo Examinar.

## Cuadro de diálogo Examinar

Selecciona un archivo de programa de comando personalizado en el árbol de directorios y en la lista de archivos.

- ▶ Para seleccionar un archivo, resalte su nombre en la lista de archivos. Después elija Aceptar.

### Opciones del cuadro de diálogo

Nombre de archivo	Introduzca el patrón de nombre de archivo que busca. Si introduce la ruta completa de los archivos que desea seleccionar, la unidad y el nombre de directorio cambian automáticamente.  Por ejemplo, si el directorio activo es C:\WINDOWS, puede introducir D:\CARTAS en el recuadro. Los archivos de D:\CARTAS aparecen al pulsar Retorno. La unidad y el nombre del directorio que aparecen en los paneles Unidad y Directorio también cambian.
Lista de archivos	Los archivos que cumplen el patrón de nombre de archivo aparecen aquí. Seleccione un nombre de archivo y elija Aceptar para seleccionarlo.
Directorio	Cambia el directorio seleccionado haciendo clic en las carpetas de directorio.
Unidad	Cambia la unidad seleccionada haciendo clic en la flecha que hay a la derecha del cuadro y seleccionando otra unidad en la lista que aparece.
Mostrar lista de archivos de tipos	Haga clic en la flecha que hay a la derecha del cuadro para seleccionar una categoría de archivo diferente. La categoría que aparece aquí se anula escribiendo otra especificación de categoría en el cuadro de texto Nombre de archivo.



