



**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA
A LOS ASISTENTES A LOS CURSOS**

Las autoridades de la Facultad de Ingeniería, por conducto del jefe de la División de Educación Continua, otorgan una constancia de asistencia a quienes cumplan con los requisitos establecidos para cada curso.

El control de asistencia se llevará a cabo a través de la persona que le entregó las notas. Las inasistencias serán computadas por las autoridades de la División, con el fin de entregarle constancia solamente a los alumnos que tengan un mínimo de 80% de asistencias.

Pedimos a los asistentes recoger su constancia el día de la clausura. Estas se retendrán por el periodo de un año, pasado este tiempo la DECFI no se hará responsable de este documento.

Se recomienda a los asistentes participar activamente con sus ideas y experiencias, pues los cursos que ofrece la División están planeados para que los profesores expongan una tesis, pero sobre todo, para que coordinen las opiniones de todos los interesados, constituyendo verdaderos seminarios.

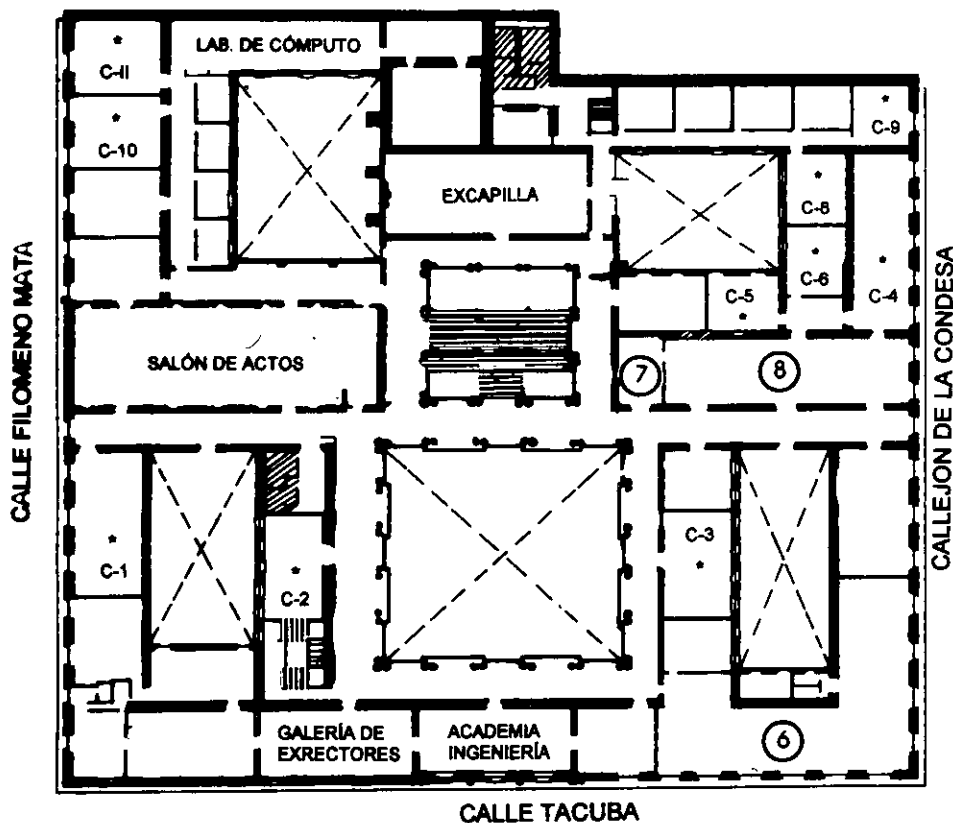
Es muy importante que todos los asistentes llenen y entreguen su hoja de inscripción al inicio del curso, información que servirá para integrar un directorio de asistentes, que se entregará oportunamente.

Con el objeto de mejorar los servicios que la División de Educación Continua ofrece, al final del curso deberán entregar la evaluación a través de un cuestionario diseñado para emitir juicios anónimos.

Se recomienda llenar dicha evaluación conforme los profesores impartan sus clases, a efecto de no llenar en la última sesión las evaluaciones y con esto sean más fehacientes sus apreciaciones.

**Atentamente
División de Educación Continua.**

PALACIO DE MINERIA



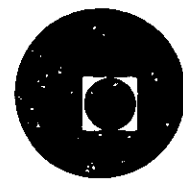
1er. PISO

GUÍA DE LOCALIZACIÓN

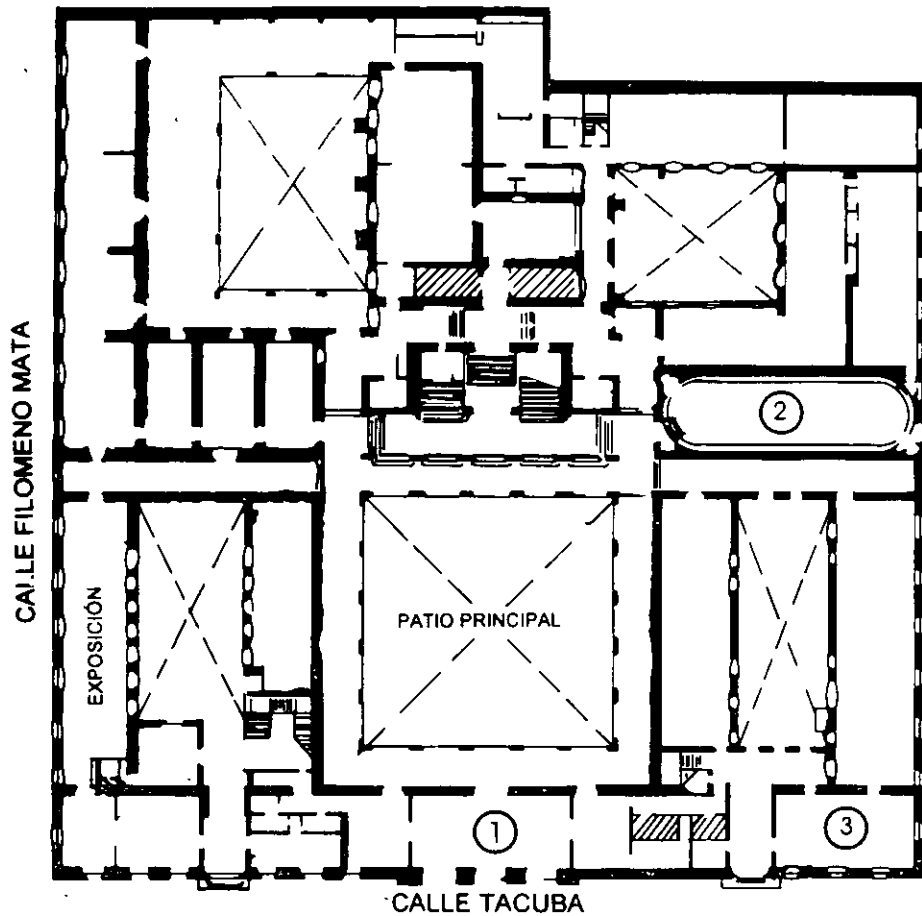
1. ACCESO
 2. BIBLIOTECA HISTÓRICA
 3. LIBRERÍA UNAM
 4. CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN "ING. BRUNO MASCANZONI"
 5. PROGRAMA DE APOYO A LA TITULACIÓN
 6. OFICINAS GENERALES
 7. ENTREGA DE MATERIAL Y CONTROL DE ASISTENCIA
 8. SALA DE DESCANSO
- SANITARIOS
- * AULAS



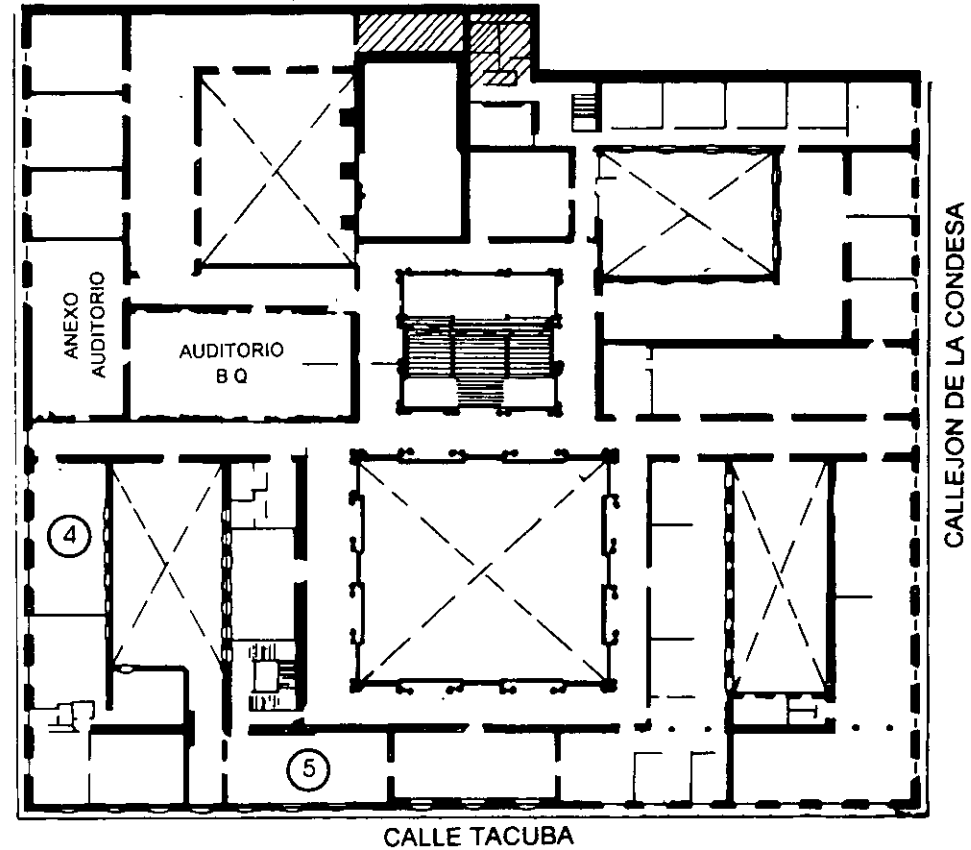
DIVISIÓN DE EDUCACIÓN CONTINUA
FACULTAD DE INGENIERÍA U.N.A.M.
CURSOS ABIERTOS



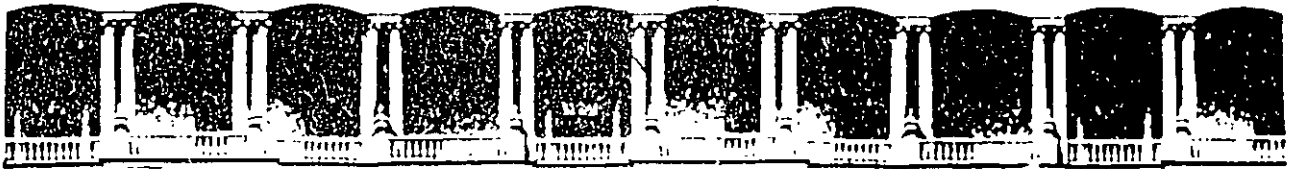
PALACIO DE MINERIA



PLANTA BAJA



MEZZANINNE



FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA

CURSOS INSTITUCIONALES

SOPORTE A MICROSOFT OFFICE PARA WINDOWS 95

Del 26 al 30 de mayo de 1997.

Apuntes Generales

Palacio de Minería
1997.

SOPORTE A MICROSOFT OFFICE PARA WINDOWS 95

1.- INTRODUCCION A MICROSOFT OFFICE 95



SOPORTE A MICROSOFT OFFICE PARA WINDOWS 95



Soporte a Microsoft Office para Windows 95 para personal profesional de S.C.T.

Objetivo: El asistente conocerá con detalle el manejo de Microsoft Office 95 para poder brindar un adecuado soporte técnico a los usuarios finales.

1. Introducción a Microsoft Office 95

- Lo nuevo en Microsoft Office
- Requerimientos de hardware
- Optimización de los recursos de la computadora para el uso de Microsoft Office 95
- Barra de acceso directo a Microsoft Office
- Uso conjunto de varias aplicaciones de Office
- Personalización de Office
- Opciones de instalación (desde el servidor, nodo de red o local)

2. Elaboración de documentos en Word 95

- Creación del primer documento en Word
- Creación de membretes y sobres
- Creación de un boletín sencillo
- Creación de un prospecto
- Creación de un plan de negocios
- Inserción de imágenes y gráficas en un documento
- Mejorar el aspecto de un documento en Word

3. Trabajo con hojas de cálculo en Excel 95

- Creación del primer libro de Microsoft Excel
- Mejorar el aspecto de una hoja de cálculo con Microsoft Excel
- Creación de un gráfico con los datos de una hoja de cálculo
- Agregar una gráfica a un documento
- Personalización de una gráfica

4. Elaboración de presentaciones en Power Point 95

- Creación de la primera presentación de Power Point
- Preparar rápidamente una presentación en Blanco y Negro
- Creación de documentos para los participantes y notas del orador
- Personalización de presentaciones
- Transferencia de información entre Power Point y otras aplicaciones

- Comunicar ideas mediante imágenes
- Preparar presentaciones electrónicas
- Hacer una presentación electrónica

5. Manejo de bases de datos con Access

- Creación de la primera base de datos de Microsoft Access
- Diseñar una base de datos de inventario personalizada
- Mover una lista de producto a Microsoft Access
- Agregar proveedores a una base de datos de inventario
- Introducción sencilla de datos
- Mejorar el aspecto de un formulario de productos
- Creación y mejora de un informe de inventario
- Convertir una base de datos de inventario en una aplicación

6. Intercambio de información a través de la red

- Compartir documentos electrónicamente
- Revisión en equipo de un documento en Word
- Compartir libros de Microsoft Excel
- Creación de un manual en pantalla
- Trabajo en una base de datos compartida

7. Programación de citas, tareas y reuniones

- Programación de una cita
- Mantenimiento de una lista de tareas pendientes
- Programación de una reunión

Duración: 30 hrs

Requisitos previos: Conocimientos generales de Windows 95 y versiones anteriores de Microsoft Office

Nuevas formas de trabajar

por Patty Winter y Rick Winter

El hecho de combinar los productos de software para venderlos como un todo no resulta nuevo. Los paquetes de software han existido desde hace años. No obstante, la novedad radica en la integración total de esos productos de software. La integración ya no consiste en conmutar entre aplicaciones, o en una aplicación que trata sin éxito de realizar la tarea de cuatro aplicaciones a la vez.

En lo que se denomina el *paquete integrado*, los principales vendedores de software ofrecen aplicaciones esenciales de oficina tales como procesadores de textos, hojas de cálculo, bases de datos y gráficos de presentación. Los productos esenciales conforman las aplicaciones estrella en sus respectivas categorías. Los vendedores añaden a este núcleo aplicaciones auxiliares, tales como correo electrónico, creación de gráficos, gestión de información personal y organigramas, ofreciendo el paquete completo de productos por un precio total que representa hasta el 50% menos que el precio de compra de la suma de cada uno de los productos.

Microsoft Office 95 se sitúa a la cabeza en lo referente a aplicaciones clave que funcionan en conjunto y que ofrecen un interfaz común de usuario. Las aplicaciones se parecen entre sí y funcionan de manera parecida, lo cual reduce la curva de aprendizaje y mejora la productividad. Office 95 facilita a los usuarios la tarea de compartir información, documentos y gráficos a lo largo de las aplicaciones.

En este capítulo se estudia:

- Qué incluye Office 95
- Las novedades de Office 95
- Los objetivos de diseño de Office 95
- Cómo determinar la aplicación a utilizar

Qué incluye Office 95

La edición estándar de Microsoft Office 95 incluye las siguientes aplicaciones:

- Microsoft Word para Windows, versión 7.0
- Microsoft Excel, versión 7.0

- Microsoft PowerPoint, versión 7.0
- Microsoft Schedule+, versión 7.0
- Cuaderno de Microsoft Office
- Barra de acceso directo de Microsoft Office

La edición profesional de Office 95 añade al paquete básico una potente base de datos, Microsoft Access, versión 7.0.

Las novedades de Office 95

Microsoft Schedule+ y Cuaderno de Microsoft Office son dos nuevas aplicaciones que vienen en el paquete de Office 95. Por añadidura, se incluyen una serie de nuevas funciones a todos los productos, así como en lo que respecta a la operatividad entre ellos.

Windows 95

Windows 95 no forma parte de Office 95. De hecho, antes de nada se necesita instalar Windows 95 para ejecutar Office 95. Sin embargo, existen muchas funciones de Windows 95 que se integran bien con Office 95.

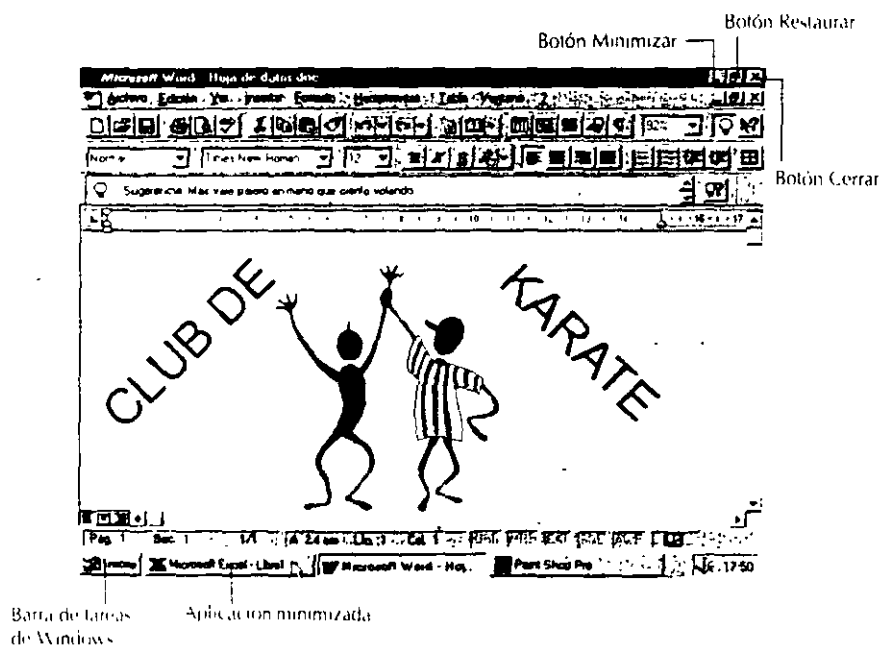
Consejo

Para buscar un resumen en pantalla de nuevas funciones elija **Ayuda**, **Temas de Ayuda de Microsoft**, pulse en la ficha **Buscar** y escriba lo nuevo en el primer cuadro de texto

Pantalla

La nueva barra de tareas (véase fig. 1.1) le permite no sólo iniciar aplicaciones, sino también abrir aplicaciones minimizadas con una simple pulsación. El nuevo botón de cerrar (x), localizado en la esquina superior derecha de la ventana, facilita en gran medida las operaciones de cerrar documentos y aplicaciones. Cuando minimiza una aplicación usando el botón de minimizar, rediseñado con un subrayado, ésta aparecerá minimizada en un rectángulo situado en la barra de tareas.

Fig 1. 1
Para abrir la ventana de la aplicación, pulse en una aplicación minimizada



Nombres de archivo y propiedades

Windows 95 actualmente le permite utilizar nombres de archivo de hasta 255 caracteres, incluyendo espacios. Esta función aumenta en gran medida la flexibilidad, la búsqueda y la comprensión de lo que se ha guardado. Windows 95 también ha añadido una serie de propiedades de archivo, que Office 95 amplía.

Ahora puede examinar rápidamente estadísticas de su documento tales como fecha de creación, número de palabras y guardar información del autor y del documento en su archivo. Además, directamente diseñadas a modo de cuadros de diálogo de archivos estándar, se encuentran posibilidades mejoradas de búsqueda que le ayudan a encontrar su archivo en función del contenido, tiempo, u otras propiedades del mismo (véase fig. 1.2).

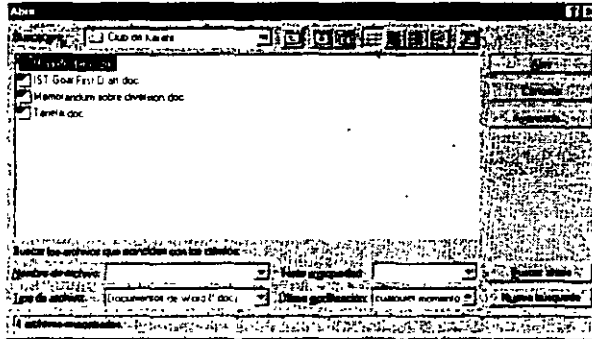


Fig. 1.2
Los nombres de archivo largos le ayudan a recordar y encontrar sus documentos.

Microsoft Exchange y Microsoft Network

En la versión 4 de Microsoft Office se incluía con el producto una licencia para utilizar Microsoft Mail. El administrador tenía que establecer el programa para cada usuario. En Windows 95, Microsoft Exchange se entrega con el producto, incluyendo Microsoft Mail y Microsoft Fax, en conjunción con otro nuevo producto, Microsoft Network. Ud. tendrá la oportunidad de enviar su documento de Word, hoja de cálculo de Excel, o presentación de PowerPoint a personas que se hallen tanto dentro como fuera de su oficina. Microsoft Mail también se ha combinado con Word para permitirle crear y acceder a mensajes haciendo uso de todas las potentes herramientas que ofrece Word (véase fig. 1.3).

Nuevos productos

Además de las nuevas funciones de Windows 95, podrá encontrar programas nuevos en Office 95. Schedule+ es una aplicación actual que ha sido mejorada y añadida a Office 95. Cuaderno es una nueva función que le permite combinar documentos desde aplicaciones diferentes.

Schedule+

Microsoft Schedule+ (véase fig. 1.4) ha sido añadida a Office 95. El anterior producto se vendía por separado y comenzó su andadura en 1992.

La versión para Office 95 incluye la primera versión actualizada de Schedule+1.0. Schedule+ le permite concertar citas, escribir listas de tareas pendientes y establecer reuniones con sus colegas usando el nuevo Asistente para reuniones. Schedule+ contiene ahora una función que le ayuda a establecer y gestionar sus objetivos. Consulte el Asistente para los Siete Hábitos de Personas Altamente Eficientes, basado en el famoso seminario y libro de Steven Covey. Para más información sobre cómo trabajar con Schedule+, remítase al capítulo 37, "Cómo utilizar Schedule+".

Fig. 1.3
Puede utilizar WordMail para crear mensajes de correo electrónico dentro de Word.

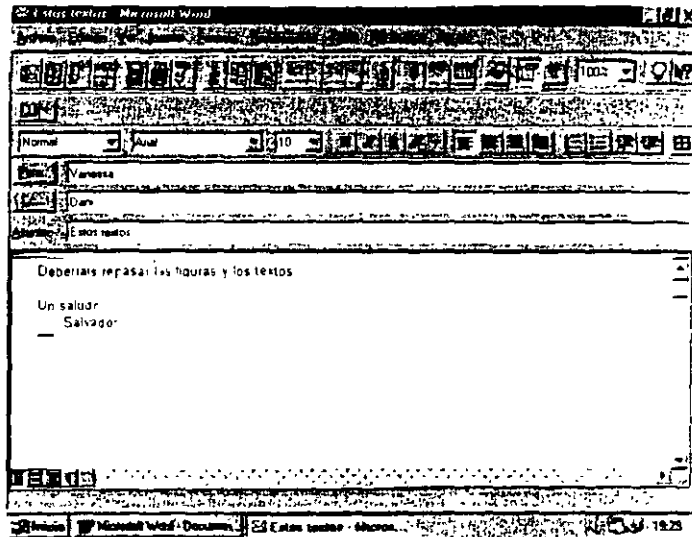
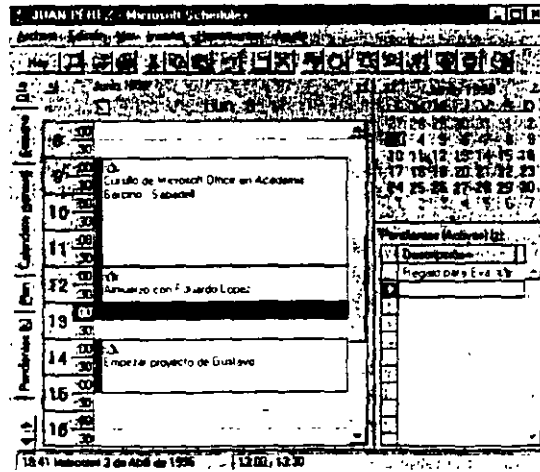


Fig. 1.4
Schedule+ automatiza su vida haciendo que sus citas y tareas le sean más fáciles de administrar



Cuaderno de Microsoft

Utilice Cuaderno de Microsoft para añadir documentos de Word, Excel y PowerPoint y crear un único documento para la impresión, edición, almacenamiento y distribución. La fig. 1.5 muestra un documento cuaderno que contiene dos documentos de Word, una hoja de cálculo de Excel y una presentación de PowerPoint. La parte izquierda de la pantalla muestra los documentos que integran el archivo cuaderno. Añada, elimine y asigne nuevo nombre a los documentos mediante el Menú **Sección**. Abra, guarde, cree e imprima documentos cuaderno mediante el Menú **Archivo**. Los otros menús y barras de herramientas cambian en función de los documentos fuente (Word, Excel y PowerPoint). Para más información sobre el Cuaderno, consulte el capítulo 33, "Cómo utilizar el cuaderno".

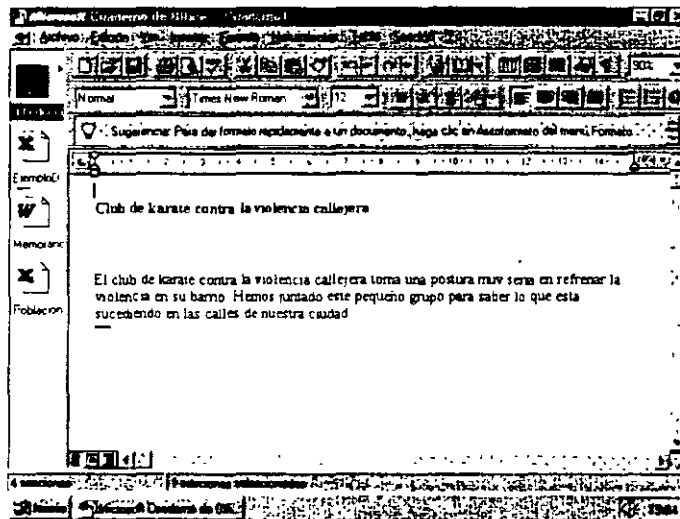


Fig. 1.5 La parte izquierda de la pantalla muestra los documentos fuente. Si un documento de Word (Introducción) se encuentra resaltado, las barras de herramientas son las correspondientes a Microsoft Word.

Mejoras en todos los productos de Office

En Office 95 existen mejoras, así como nuevas funciones que le facilitarán la conmutación entre aplicaciones. La tabla siguiente incluye una lista de las principales modificaciones, nuevas funciones, así como todo lo necesario para iniciarse:

Función	Iniciación
Autocorrección está actualmente disponible en Word, Excel y PowerPoint y puede corregir faltas de ortografía a medida que se escriben	Herramientas, Autocorrección o, pulse sobre el botón [Ortografía], página 86



(continúa)

Función	Iniciación
<p>La <i>Ayuda</i> es actualmente la misma para todas las aplicaciones de Office. Una nueva función de Búsqueda le permite buscar un texto no sólo desde los títulos de las páginas de ayuda (como lo hacía la búsqueda anterior), sino además a través del archivo completo de ayuda o en algunos casos a través de múltiples archivos de ayuda.</p>	<p>Ayuda, Temas o <F1>, página 140</p>
<p><i>Asistente para Ayuda</i> (véase fig. 1. 5) es otra nueva función de la ayuda que le permite escribir una solicitud de ayuda con sus propias palabras.</p>	<p>Ayuda, Asistente para Ayuda, página 144</p>
<p><i>El Asistente para ideas</i> observa lo que hace en Word y Excel. Cuando el Asistente de ideas tenga una sugerencia para facilitar una tarea, su botón se iluminará. Pulse sobre el botón para obtener una nota rápida.</p>	<p>Pulse sobre el botón [<i>Asistente para ideas</i>], página 152</p>
<p>La <i>barra de acceso directo de Microsoft Office</i> (véase fig. 1. 6) le permite iniciar nuevos Barra de tareas Inicio Windows archivos. Elija de una serie organizada de fichas que muestran los tipos generales de documentos, tales como folletos, cartas, memorandos, planes comerciales, certificados, informes y tarjetas de tiempo. La barra de acceso directo de Office también iniciará una aplicación que cree un documento. No tiene por qué recordar la aplicación que utilizo. La barra de acceso directo a su vez le permite concertar citas, añadir tareas, agregar contactos a Schedule+, e iniciar el Asistente para Ayuda en todas las aplicaciones.</p>	<p>Programas, barra de acceso directo de Microsoft Office, página 31</p>
<p><i>El Libro Obtenga Resultados con Microsoft Office para Windows 95</i> (fig. 1. 5) está en el CD-ROM que viene incluido con <i>Microsoft Office para Windows 95</i>. Al pulsar en el botón de la barra de acceso directo y tener el CD-ROM de Microsoft Office en el lector de CD-ROM, se podrán leer y buscar las instrucciones detalladas, así como las explicaciones sobre tareas que se necesitan llevar a cabo.</p>	<p>Pulse sobre el botón [Libro Obtenga Resultados]</p>

(continúa)

Función	Iniciación
<i>Compatible con Office</i> (fig. 1. 5) también inicia información sobre el CD-ROM. Esta característica muestra otros productos compatibles con Microsoft Office.	Pulse sobre el botón [Compatible con Office]

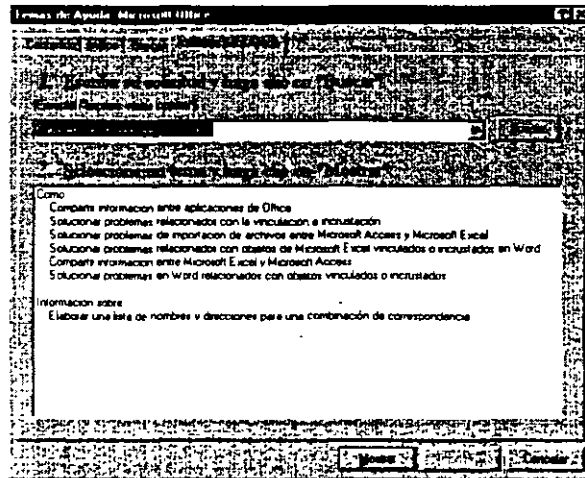


Fig. 1. 6
El Asistente para Ayuda le permite escribir preguntas de la forma en que quiera formularlas.

Mejoras en el software individual

Además de las mejoras generales realizadas en Office 95, cada uno de los productos del paquete contiene mejoras.

Microsoft Word para Windows 95

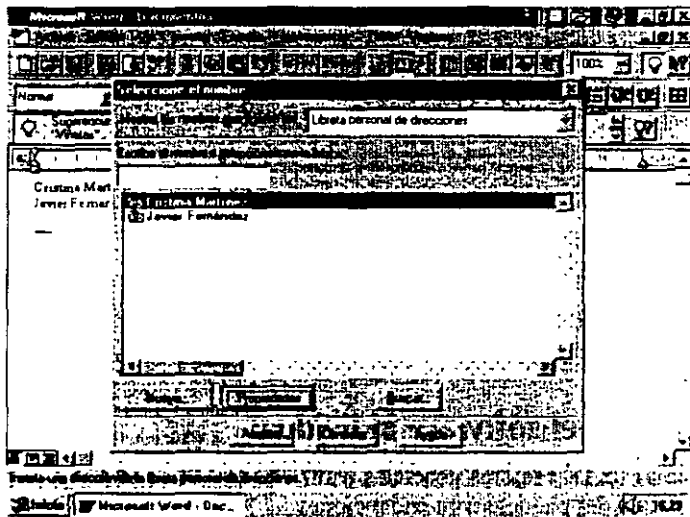
La siguiente tabla muestra algunas de las mejoras de Microsoft Word y la forma de iniciar la función

Función	Iniciación
<i>Revisión ortográfica</i> , una nueva función, identifica las faltas de ortografía tan pronto como las palabras son escritas, a través de una línea roja ondulada y da sugerencias pulsando el botón derecho del ratón o mediante <Shift>+<Retorno>	Herramientas, Opciones, Ortografía. Revisar la ortografía automáticamente, página 196
<i>Autoformato Mejorado</i> aplica bordes, títulos, encabezados, tracciones y formato de lista a medida que escribe.	Herramientas, Opciones, ficha de Autoformato página 240

(continúa)

Función	Iniciación
<i>La Libreta de Direcciones</i> (véase fig. 1. 7) gestiona sus contactos para sobres, etiquetas, combina correspondencia y puede conectar con su información de contacto de Schedule+.	Pulse sobre el botón [Libreta de Direcciones], página 925.
<i>Destacar</i> acentúa las partes importantes del documento con un fondo de color.	Pulse sobre el botón [Destacar], página 183.

Fig. 1. 7
La Libreta de direcciones le administra nombres y direcciones =



Microsoft Excel para Windows 95

La siguiente tabla muestra algunas de las mejoras de Microsoft Excel y la forma de iniciar la función.

Función	Iniciación
<i>Data Map</i> (véase fig. 1. 8) representará sus datos geográficamente	Pulse sobre el [Map] página 499.
<i>Asistente para Plantillas con seguimiento de Datos</i> le permitirá convertir sus hojas de cálculo actuales en figuras y capturar lo escrito en una base de datos separada.	Datos, Asistente para Plantilla , página 423.

(Continúa)

Función	Iniciación
<p><i>Autocompletar y Lista de Entradas</i>. <i>Previas</i> le permitirán completar una entrada automáticamente. Ud. podrá crear una entrada en función de las entradas previas según escribe las primeras letras dentro de la celda. También puede pulsar con el botón derecho del ratón en Elegir de la Lista, y escoger de una lista de entradas previas.</p>	página 285.
<p><i>Información sobre celdas</i> aparecerá cuando se detenga con el ratón sobre la celda con una nota</p>	<p>Insertar, Notas, página 458.</p>
<p><i>Filtro Automático con los 10 más</i> le permite ver los ítems superiores o inferiores de la lista.</p>	<p>Data, Filtro, Filtro Automático y a continuación los 10 más, página 428.</p>
<p><i>AutoCálculo</i> calcula automáticamente sus <u>datos</u> resaltados seleccionando un rango y visualizando en la barra de estado la suma, el promedio la cuenta con el botón derecho del ratón.</p>	<p>Véase barra de estado, página 282.</p>
<p>Los ajustes de <i>Proteger</i> se han mejorado con vistas a una protección más completa y flexible del trabajo del usuario.</p>	<p>Herramientas, Proteger, página 453.</p>

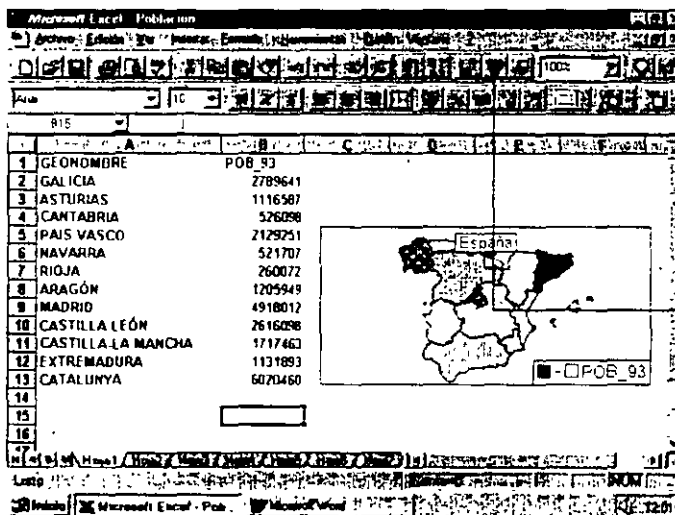


Fig. 1. 8
 Ahora podrá hacer mapas geográficos de su información con Data Map

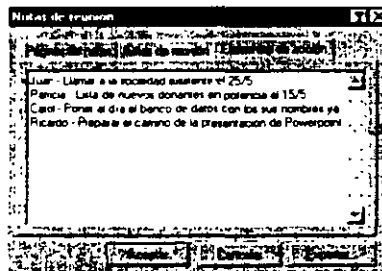
Boton Mapa

Microsoft PowerPoint para Windows 95

La siguiente tabla muestra algunas de las mejoras de Microsoft PowerPoint y la forma de iniciar la función:

Función	Iniciación
<i>AutoClipArt</i> le sugiere imágenes apropiadas para su mensaje.	Herramientas, AutoClipArt , página 515.
<i>Asistente para Presentaciones Portátiles</i> le permite guardar su presentación en un disco para su visualización en otros ordenadores.	Archivo, Presentaciones portátiles , página 657.
<i>Asistente para Autocontenido</i> cambia su mensaje en función del tamaño y tipo de audiencia y el tiempo de que disponga para su presentación.	página 578
<i>Animación de Títulos</i> le permite crear efectos especiales para títulos de Animación.	Ver, Barras de Herramientas Efectos de Animación o Efectos de Herramientas , página 657.
<i>Presentación en red</i> le permite a un grupo de trabajo repasar simultáneamente una presentación en la red mientras verifica sus notas, timing y siguiente día positiva.	Herramientas, Presentación Remota , página 870.
<i>Notas de la reunion</i> (véase fig. 1.9) le permite tomar y guardar notas durante su presentación -o exportarlas a Word para crear actas. También puede grabar elementos de acción y mostrarlos al final de su presentación.	Herramientas, Notas de a Reunión , pagina 650

Fig. 1.9
Utilice Notas de la Reunion para escribir sus elementos de acción de forma que estos aparezcan en la última diapositiva.



Objetivos de diseño de Microsoft Office

El objetivo de Microsoft Office es el de proporcionar a los usuarios lo siguiente:

- Un interfaz común de usuario (operaciones estandarizadas de los menús, barras de herramientas y cuadros de diálogo)
- Acceso rápido de una aplicación a otra dentro del paquete de oficina
- Información compartida a través de las aplicaciones
- Recursos compartidos a través de las aplicaciones
- Información compartida a través de grupos de trabajo
- En el futuro, un lenguaje automatizado de tareas comunes

Microsoft se está esforzando en cumplir los citados objetivos. Muchas de las aplicaciones principales se han sometido (y continúan someténdose) a revisiones con el objeto de alcanzar estos objetivos. Para aquellos iniciados en los productos principales, el menú actual o los cambios en las barras de herramientas pudieran resultar enojosos. A largo plazo, no obstante, un interfaz común de usuario a través de las aplicaciones aumenta el uso eficiente y real de todas ellas.

Cómo proporcionar un interfaz común de usuario

Una de las ventajas claras del interfaz común de usuario a través de las aplicaciones consiste en que, conociendo una aplicación del paquete, se conocen los fundamentos operativos del resto de las aplicaciones. La fig. 1.10 ilustra la similitud entre las barras de menús y las barras de herramientas de Excel y de Word. Observe que Word contiene una opción de menú **Tabla**, mientras que Excel contiene una opción de menú **Datos**. Aunque el objetivo sea el de proporcionar un interfaz común de usuario, permanecerá en

► Véase "Ayuda en pantalla", página 135

► Véase "Cómo utilizar los asistentes", página 153

► Véase "Ver sugerencias", página 151

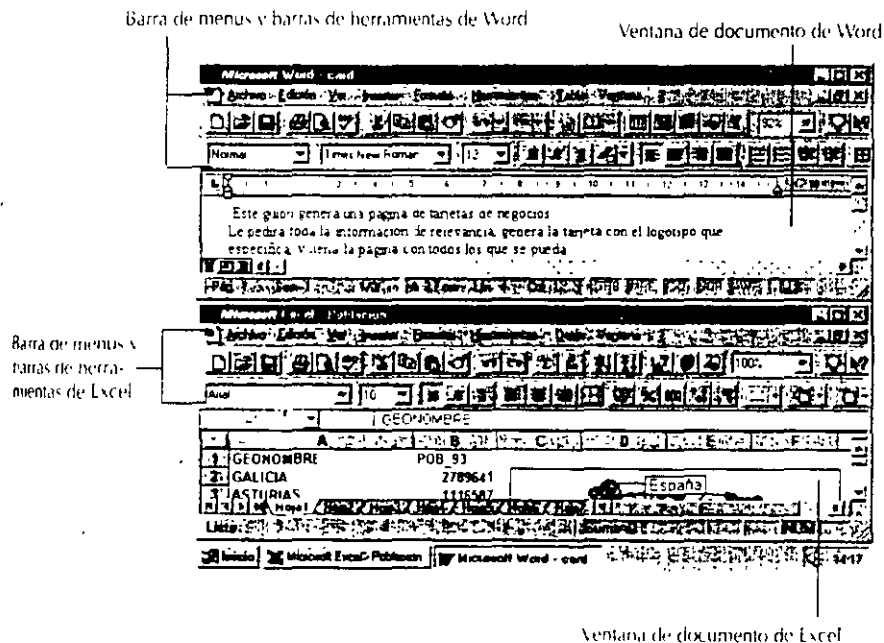


Fig. 1.10
Los menús y las barras de herramientas resultan consistentes a través de las aplicaciones

cada aplicación un cierto grado de singularidad. No obstante, funciones comunes principales tales como los comandos **Archivo**, **Abrir**, **Edición** y **Buscar**, se encuentran exactamente en el mismo lugar en cada aplicación.

Las aplicaciones de Microsoft Office son consistentes no sólo en lo que se refiere a barras de herramientas y menús similares. Los cuadros de diálogo, funciones que se pueden personalizar y funciones operativas son también similares. En las aplicaciones de Office 95, la ayuda en pantalla está disponible de diferentes formas:

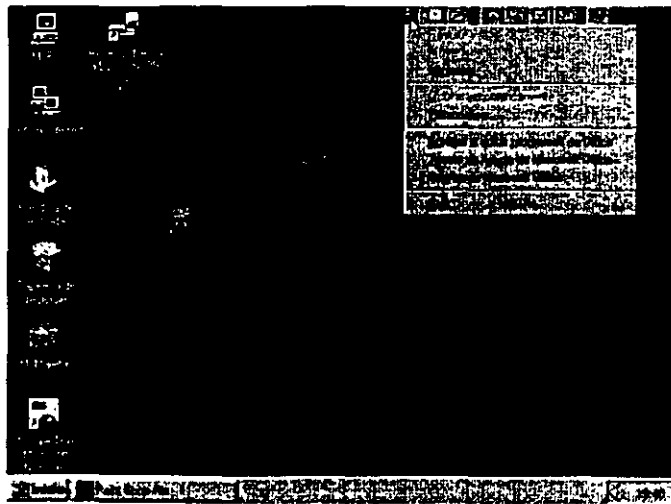
- Aplicación Ayuda
- Asistentes para Ayuda
- El uso de Asistentes
- Asistentes para ideas y Sugerencia del Día

Acceso rápido a otras aplicaciones

Microsoft Office ofrece la barra de acceso directo de Microsoft Office. Por defecto, esta barra de herramientas aparece en la esquina superior derecha del escritorio de Windows 95 (véase fig. 1. 11). Puede utilizar la barra de acceso directo para lo siguiente:

- Iniciar archivos nuevos basados en plantillas y asistentes.
- Abrir archivos existentes e iniciar automáticamente la aplicación correspondiente.
- Agregar tareas, concertar citas y agregar contactos.
- Usar funciones de Office tales como configuración y Asistentes para Ayuda.
- Agregar otras aplicaciones a la barra de acceso directo de Office.
- Conmutar entre aplicaciones de Microsoft Office.
- Iniciar aplicaciones de Microsoft Office.

Fig. 1. 11
La barra de acceso directo de Microsoft Office facilita el acceso a abrir y crear documentos y funciones del Administrador de Office



La barra de acceso directo es tan sólo una de las formas en que Microsoft Office proporciona acceso rápido a las aplicaciones. En cada aplicación de Microsoft Office, la barra de herramientas de la aplicación ofrece acceso directo a las funciones correspondientes de otras aplicaciones. Por ejemplo, en Word podrá insertar una Hoja de Cálculo de Excel en un documento simplemente pulsando en un botón de la barra de herramientas. Esta operación inicia Excel y ofrece la totalidad de sus funciones para esa hoja de cálculo incrustada (véase fig. 1.12). Observe que sin necesidad de salirse de Word, los menús y las barras de herramientas cambian a los de Excel cuando se edita la hoja de cálculo.

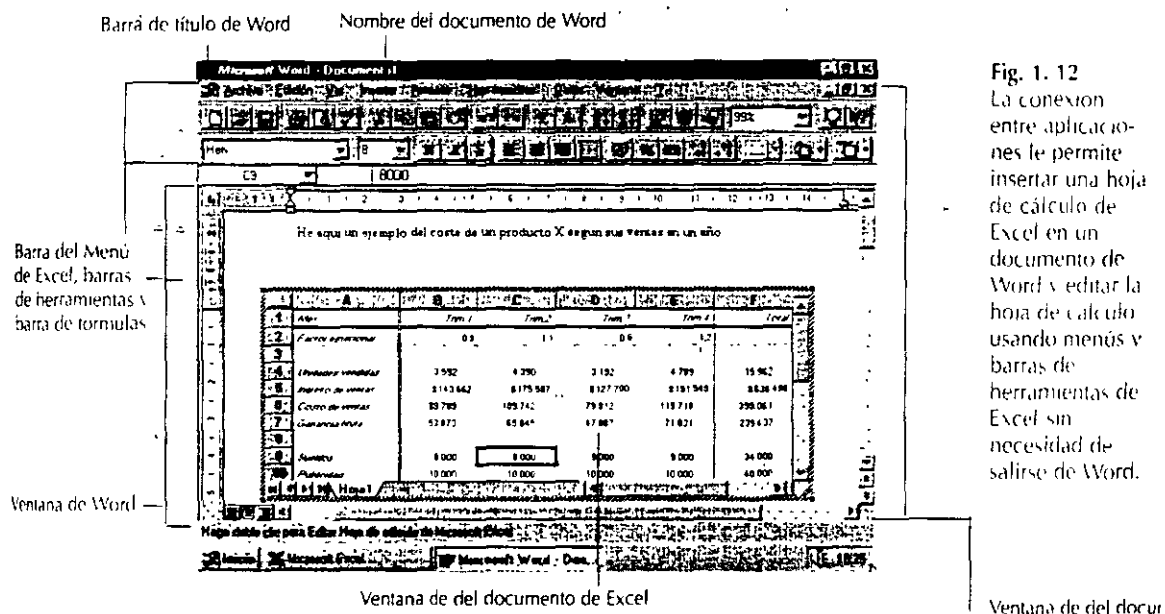


Fig. 1.12
La conexión entre aplicaciones le permite insertar una hoja de cálculo de Excel en un documento de Word y editar la hoja de cálculo usando menús y barras de herramientas de Excel sin necesidad de salirse de Word.

Cómo compartir datos entre aplicaciones

Los productos de Microsoft Office ofrecen varios métodos de compartir datos entre aplicaciones

Método	Descripción
Copiar	Copia la información desde la aplicación fuente a la aplicación término a través del Portapapeles
Vincular	Vincula una copia de los datos del documento fuente con el documento término (y guarda los datos con el documento fuente)
Incrustar	Incrusta datos del documento fuente en el documento término (guarda datos con el documento término).

Las aplicaciones de Microsoft Office comparten información con facilidad. Cuando copia una tabla de Excel a Word, por ejemplo, aparecerá una tabla de Word que retendrá todas las fuentes y formatos de Excel. No necesitará

► VÉASE "Cómo conmutar entre documentos", página 131

► VÉASE "Cómo copiar información de una hoja de cálculo", página 675

► VÉASE "Cómo utilizar procedimientos comunes para vincular documentos", página 689

► VÉASE "Cómo utilizar clipart gallery de Microsoft", página 515

dar un nuevo formato a su tabla, cosa que sin embargo podría ocurrir con otros productos.

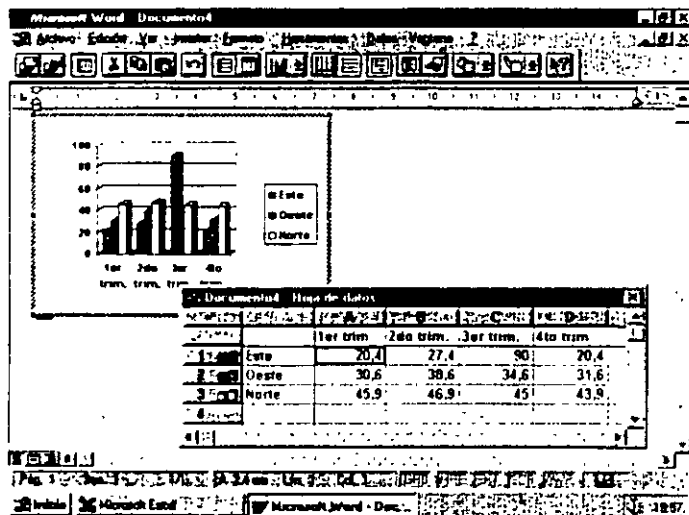
La acción de vincular e incrustar funciones se beneficia de las especificaciones OLE de Microsoft Windows. Los documentos vinculados se actualizan automáticamente al modificar un documento fuente. Los documentos incrustados dan acceso a la aplicación fuente mientras se almacene la información en la aplicación término. Cada función tiene sus pros y sus contras y sirve para un propósito específico.

Microsoft Office amplía la información compartida más allá de la integración entre aplicaciones, ofreciendo integración entre grupos de trabajo con Microsoft Exchange. Los usuarios pueden enviar documentos, hojas de cálculo y archivos de información desde dentro de la aplicación fuente. Las circulares pueden unirse a archivos que Exchange a continuación enviará al grupo o bien las dirigirá a cada persona, secuencialmente, de una en una.

Cómo compartir recursos entre aplicaciones

En Microsoft Office, un elemento clave es que reconoce que ciertos recursos son necesarios para más de una aplicación. Clip art es necesario para llevar a cabo, por ejemplo, diferentes tareas: de procesamiento de textos, de hojas de cálculo y de presentación de gráficos. Microsoft Office no sólo duplica estructura de programa, sino que además proporciona una aplicación auxiliar común a todas las aplicaciones: Microsoft ClipArt Gallery. Es igualmente necesario hacer común a todas las aplicaciones un sistema de preguntas (que formule preguntas sobre sus datos), una herramienta para gráficos (véase fig. 1.13) y una herramienta de dibujo para organigramas.

Fig. 1.13
Microsoft Graph, una de las aplicaciones auxiliares que conecta con Microsoft Office, se utiliza para dibujar gráficos en todas las aplicaciones



Cómo ofrecer un lenguaje común

El ofrecer un lenguaje común entre aplicaciones constituye el objetivo más ambicioso de Microsoft Office. Anteriormente, cada producto estaba programado de una manera diferente y/o tenía un lenguaje de macros distinto. La versión 5.0 de Excel es el primer producto del paquete de Microsoft Office que ofrece un lenguaje común para todos los productos de Office: Visual Basic, para Aplicaciones (VBA). VBA utiliza OLE y es capaz de enviar pulsaciones a otras aplicaciones (posibilitando que VBA ejecute un proceso compartido entre aplicaciones).

Hasta que VBA se añada al resto de los productos del paquete, los usuarios tendrán que aprender WordBasic y VBA para automatizar sus tareas comunes de oficina.

► VÉASE
"Comprendiendo
Visual Basic para apli-
caciones", página 767

Cómo determinar la aplicación que se va a utilizar

La siguiente tabla presenta algunas de las tareas comunes de oficina y herramientas sugeridas para la aplicación, a fin de llevar a cabo cada tarea:

Tarea	Aplicación	Comentario
Crear cartas, memorandos, informes y documentos periódicos	Word	Use Word para proyectos extensos.
Crear presupuestos, facturas, seguimiento de declaraciones, de la renta, estadísticas.	Excel	Use Excel para proyectos que contengan muchas cifras y que entrañen muchos cálculos
Crear diapositivas, esquemas, presentaciones	PowerPoint	Use PowerPoint para presentaciones cuando desee crear diapositivas de 35 mm, transparencias, folletos y presentaciones por ordenador.
Mantener listas de correspondencia.	Word, Excel o Schedule+	Mantenga la información en Word, Excel o Schedule+.
Crear envíos de correo o documentos personalizados.	Word	Fusione la información sobre la lista de correspondencia en Word para imprimir portadas, folletos y etiquetas.
Crear una tabla con información financiera para usarla en una presentación.	Excel PowerPoint	Cree una tabla en Excel. Incruste una tabla en una diapositiva de PowerPoint.

(continúa)

Consejo

Utilice el botón Inicio de la barra De Acceso directo y deje que Office decida que aplicación le conviene más utilizar

Fin
Introducción

Tarea	Aplicación	Comentario
Enviar un documento a un grupo para recibir respuesta.	Word, Excel Exchange	Cree un documento, hoja de cálculo, o presentación en la aplicación deseada. Envíe el archivo usando la función de direccionamiento de Exchange.
Ofrecer seguimiento de auditoría entre los datos de hoja de cálculo y documento de informe anual.	Excel Word	Cree los planes de apoyo que necesita Excel. Cree el documento del informe anual en Word. Use OLE para vincular los datos de Excel al documento de Word. Siempre que se modifiquen los datos de la hoja de cálculo, el documento del informe anual se actualizará automáticamente.
Crear, imprimir, y distribuir un boletín de noticias de departamento	Word Exchange	Cree un boletín de noticias en Word. Distribuya un boletín de noticias electrónicamente usando la función Enviar de Exchange.

Teniendo cuatro o cinco aplicaciones de software nuevas tan estrechamente integradas, la tarea de decidir qué producto utilizar para cada tarea se hace difícil. Experimentar con cada aplicación es la mejor manera de saber combinar las capacidades de cada una de ellas.

Este libro trata de ayudarle en esta tarea. Los siguientes capítulos de la Parte I repasan las características comunes de todas las aplicaciones del paquete de Microsoft y señalan algunos detalles.

Las Partes II, III y V le guiarán a través de las funciones y capacidades de cada producto. La Parte IV le enseñará a utilizar objetos de multimedia tales como gráficos, imágenes, sonido y películas. La Parte VI ilustra cómo los productos de Microsoft Office se complementan, simulando situaciones de trabajo. La Parte VII le enseñará a personalizar Microsoft Office con sus propias barras de herramientas, menús y automatización (macros y VBA). La Parte VIII le mostrará cómo utilizar Microsoft Office en un Grupo. La Parte IX ha sido escrita por cinco expertos en su campo cuyo objetivo es presentarle potentes técnicas de usuario.

Tome asiento y disfrute con la nueva forma de trabajar de Microsoft Office.

Consejo

Puede revisar la ortografía de todo el documento o seleccionar texto y comprobar únicamente la ortografía del texto seleccionado.

Cómo revisar la ortografía

Microsoft ofrece el comando Ortografía en Word, Excel y PowerPoint. Por lo general, se suele revisar la ortografía en un documento de procesador de texto, pero ahora es posible revisarla en documentos como hojas de cálculo y presentaciones. Los comandos Herramientas y Ortografía buscan palabras que no están en el diccionario. Además, el comando Ortografía avisa sobre cualquier problema de puntuación y mayúsculas, así como de repetición de palabras. Podrá elegir reemplazar la ortografía o ignorar el problema, e incluso podrá agregar la palabra al diccionario para un uso posterior.

Como usar el comando Ortografía

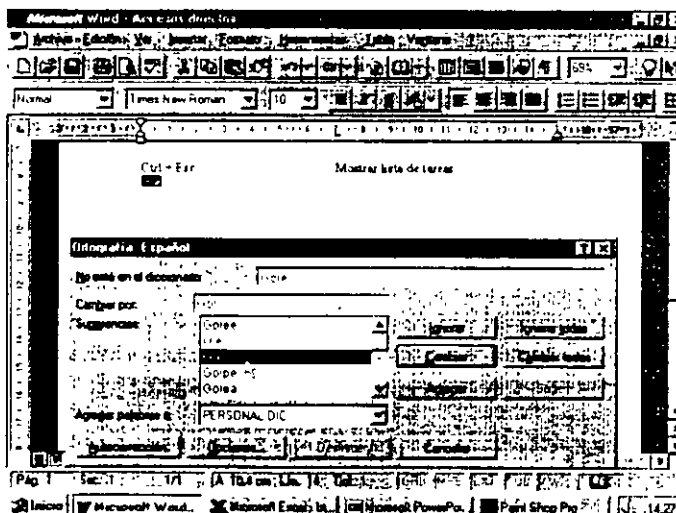
El comando Ortografía lee el texto y avisa cuando encuentra una palabra que no está en el diccionario principal o en el diccionario personalizado. El cuadro de diálogo Ortografía muestra la palabra en cuestión, sugiere una palabra de reemplazo y despliega una lista de palabras similares en ortografía que pueden servir de reemplazo.

Existen varias opciones a la hora de resolver un problema de ortografía. Ud. puede reemplazar la palabra, ignorarla, agregarla al diccionario, etc. Este apartado describe estas opciones.

Para usar el comando **Ortografía** en Word, Excel o PowerPoint, elija **Herramientas, Ortografía**. El cuadro de diálogo Ortografía aparecerá en pantalla (véase la fig. 3.20).



Fig. 3.20
 Seleccione una palabra en el cuadro
 Consejos o escriba la palabra correcta en el cuadro de texto
Reemplazar A para corregir el error del texto



La tabla 3.4 describe cada una de las opciones del cuadro de diálogo Ortografía, y que opciones existen en cada una de las aplicaciones.

Tabla 3.4 Opciones de Ortografía

Opción	W	E	P	Descripción
No en diccionario	X	X	X	Muestra la palabra en cuestión.
Reemplazar A	X	X	X	Sugiere una ortografía alternativa; es posible introducir la nueva ortografía en este cuadro de texto.
Eliminar	X	X	X	Si el cuadro de texto reemplazar A está vacío, el botón [Reemplazar] se convierte en Eliminar, que borra la palabra.
Eliminar Todo	X	X	X	Si el cuadro de texto Reemplazar A está vacío, el botón [Reemplazar Todo] se convierte en Eliminar Todo, que borra todas las apariciones de la palabra en el documento.
Consejos	X	X	X	Elija una palabra de la lista Consejos para reemplazar una palabra mal deletreada.
Ignorar	X	X	X	Se salta la aparición de la palabra.
Reanudar/Iniciar	X		X	El botón [Ignorar] cambia a Reanudar o Iniciar cuando se pulsa fuera del cuadro de diálogo Ortografía para editar el documento. El cuadro de diálogo se mantiene en pantalla. Elija [Reanudar] o [Iniciar] para volver a empezar la revisión ortográfica.
Ignorar Todo	X	X	X	Se salta todas las apariciones de la palabra en el documento.
Reemplazar	X	X	X	Sustituye la palabra mal deletreada por la palabra seleccionada introducida.

Consejo

También puede iniciar el corrector ortográfico pulsando en el botón [Ortografía] de la barra de herramientas estándar o pulsando **Ctrl**.

Tabla 3.4 Continuación

Opción	W	E	P	Descripción
Reemplazar Todo	X	X	X	Sustituye todas las apariciones de la palabra deletreada en el documento por la palabra seleccionada o introducida.
Agregar	X	X	X	Actualiza el diccionario por medio de la inclusión de una palabra que se teclé a menudo. La palabra añadida debe aparecer en el cuadro de texto de No se encuentra en Diccionario.
Consejos	X	X	X	Si la opción [Sugerir Siempre] del cuadro de diálogo de Opciones está desactivada, elija [Sugerir] para ver la lista de Consejos de corrección de la palabra mal deletreada.
Agregar Palabras A	X	X	X	Si ha instalado más de un diccionario, seleccione el diccionario al que quiere agregar la palabra en la lista desplegable.
Autocorrección	X	X		Agrega palabras mal deletreadas frecuentemente a la lista Autocorrección. Para más información, ver más adelante en este capítulo el apartado "Deje a Autocorrección Corregir su Ortografía".
Opciones	X	X		Permite elegir parámetros predeterminados o personalizar el Corrector de Ortografía.
Deshacer	X	X		Elija [Deshacer] cuando cambie de parecer sobre el último cambio de ortografía.

Tabla 3.4 Continuación

Opción	W	E	P	Descripción
Cerrar y Cancelar	X	X	X	Cancelar da por terminada la revisión ortográfica y cancela los cambios. Cerrar da por terminada la revisión ortográfica y guarda los cambios. El botón [Cancelar] cambia a Cerrar después de hacer un cambio en el documento.
Avuda en la esquina superior derecha del cuadro de diálogo	X	X	X	Elija este botón para una ayuda detallada de las opciones del cuadro de diálogo Ortografía.

Edición de un documento sin abandonar la corrección ortográfica

Si considera necesario editar el documento después de haber iniciado la corrección ortográfica, siga los siguientes pasos:

1. Pulse sobre el documento para activar la ventana del documento. El botón [Ignorar] en el cuadro de diálogo Ortografía reemplazará a Reanudar en Word y a Iniciar en PowerPoint.
2. Haga los cambios necesarios en el documento.
3. Pulse sobre el botón [Reanudar] o [Inicia]r para continuar con el proceso de revisión de la ortografía.

Uso de los diccionarios

Cuando se hace uso del comando Ortografía, las tres aplicaciones buscan por defecto en un diccionario denominado PERSONAL.DIC. Sin embargo, podrá crear sus diccionarios personalizados.

Cómo agregar palabras a un diccionario

El diccionario PERSONAL.DIC, así como cualquier otro diccionario que crea, se almacena en el carpeta de Archivos del Programa/Archivos Comunes/Carpeta Prueba o en el carpeta WINDOWSMSAPPSPROOF de su disco duro. Si pulsa en el botón [Agregar] del cuadro de diálogo Ortografía, cualquier palabra nueva que se añada pasará automáticamente a formar parte del archivo PERSONAL.DIC. Si se agrega una palabra en Excel o Word, los dos programas reconocerán la palabra añadida. PowerPoint, por el contrario, no reconocerá la palabra añadida hasta que cierre Word o Excel y PowerPoint, y abra PowerPoint de nuevo.

Cómo crear diccionarios adicionales

Si existe un léxico específico asociado con su trabajo, quizás sea conve-

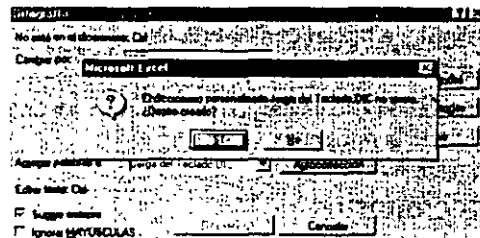
niente crear un diccionario aparte. En este caso, es posible compartir el diccionario con colaboradores.

Cuando está en Word podrá crear el nuevo archivo diccionario mientras usa la Ortografía. Si el comando Ortografía se detiene en una palabra deletreada correctamente, y ha decidido incluirla en el nuevo diccionario personalizado, siga los siguientes pasos:

1. Pulse sobre el botón de comando **Opciones**.
2. Pulse sobre el botón comando **Diccionarios Personalizados**.
3. Pulse sobre el botón de comando **Nuevo**.
4. Escriba el nombre del nuevo diccionario y elija el botón de comando **Guardar**.
5. Pulse sobre **Aceptar** dos veces para volver al cuadro de diálogo **Ortografía**.
6. Seleccione el nuevo diccionario en la lista de **Agregar Palabras A**
7. Pulse sobre el botón de comando **Agregar** para añadir la palabra al nuevo diccionario.

En Excel también podrá crear el nuevo archivo de diccionario mientras usa Ortografía. Antes de agregar ninguna palabra, cambie el nombre del archivo en el cuadro de texto de **Agregar Palabras A** del cuadro de diálogo Ortografía. Si el archivo de diccionario no existe, se le pedirá que lo cree. Elija [**Si**] para crear el nuevo diccionario (véase la fig. 3.21).

Fig. 3.21
El comando
Ortografía de
Excel le permite
crear nuevos
diccionarios a
los que Word
puede acceder



Cuando se usa el comando Ortografía, tanto Word como Excel revisan todos los diccionarios disponibles. Uno puede estar comprobando la ortografía de un documento Word o Excel y encontrarse con una palabra que necesita agregar a uno de los diccionarios personalizados. Para seleccionar un diccionario personalizado, pulse con la flecha en la lista de **Agregar Palabras A** y después sobre el diccionario que quiera utilizar.

Cómo activar y desactivar diccionarios

A veces es necesario utilizar un diccionario creado para un léxico específico de una compañía o industria. Si se encuentra en Word y el diccionario que quiere utilizar no está en la lista, siga los siguientes pasos:

1. Pulse sobre el botón **Opciones** en el cuadro de diálogo **Ortografía**.
2. Pulse sobre el botón **Diccionarios** personalizados del cuadro de diálogo **Opciones**.

3. Seleccione el nombre del archivo de diccionario en la lista de **Diccionarios personalizados**. (Una marca de verificación significa que el diccionario está activado.)
4. Seleccione **Aceptar** en el cuadro de diálogo **Diccionarios personalizados**.
5. Seleccione **Aceptar** en el cuadro de diálogo Opciones.
6. Pulse con la flecha en el cuadro de la lista **Agregar Palabras A** y seleccione de la lista el diccionario que desea utilizar.

Deje a Autocorrección corregir su ortografía

Al usar Autocorrección, puede ordenar a Word, Excel o PowerPoint que corrija los errores de ortografía según se cometen. Si, por ejemplo, teclea con frecuencia dle en vez de del, Autocorrección puede subsanar el error inmediatamente después de que éste se produzca.

La función Autocorrección corrige errores ortográficos y de formato automáticamente, o reemplaza caracteres tecleados por palabras o frases específicas. El uso de esta función le ahorrará tiempo. Imagine que tiene que teclear Rechazo de la Violencia Organizada, S. A. de forma constante; mejor que escribirlo todo cada vez, se puede agregar la abreviatura Rdlvo a la lista de Autocorrección y dejar que Word, Excel o PowerPoint hagan el trabajo de escribirlo entero. Es posible introducir errores comunes o abreviaturas de este tipo en Autocorrección, de modo que la próxima vez que se cometa el error o se escriba la abreviatura Word, Excel o PowerPoint los corrijan automáticamente.

Atención

Si PowerPoint no admite sus nuevas entradas Autocorrección, salga de PowerPoint y Word o Excel, y abra PowerPoint de nuevo

Cómo agregar palabras mientras se corrige la ortografía

Para agregar palabras a Autocorrección mientras corrige la ortografía en Word o Excel, asegúrese de que la palabra incorrecta está en el cuadro de **No en Diccionario** y la correcta en el cuadro de **Reemplazar A**. Entonces elija el botón **Autocorrección** mejor que el de **Reemplazar** o **Reemplazar Todo** en el cuadro de diálogo Ortografía.

Agregar palabras desde el menú de Herramientas

Para fijar opciones y construir entradas para Autocorrección, elija el comando **Herramientas. Autocorrección**. La figura 3.22 muestra el cuadro de diálogo Autocorrección con una nueva entrada para Word y Excel.

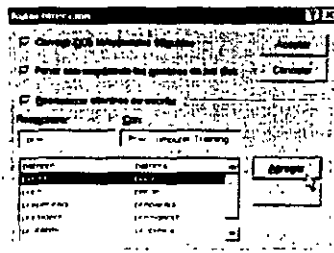
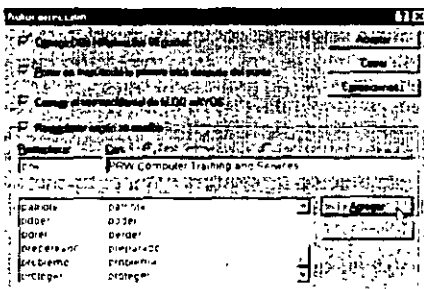
El cuadro de diálogo Autocorrección en Word ofrece cinco opciones, incluida la de conversión de comillas y de corrección de problemas de mayúsculas. Puede elegir entre mantener estas opciones activas o inactivas. El cuadro de diálogo de Excel no ofrece las opciones de Comillas o Frase. Los cuadros de texto **Reemplazar** y **Con** le permiten introducir sus propios elementos, y la lista de la parte inferior del cuadro de diálogo Autocorrección enseña las listas por defecto de Word y Excel además de los elementos añadidos por Ud. Es posible añadir y eliminar elementos en cualquier momento.

Para utilizar la función Autocorrección, siga los siguientes pasos:

1. Elija **Herramientas, Autocorrección**. Aparecerá el cuadro de diálogo **Autocorrección**.
2. Seleccione las opciones que quiera utilizar.

Fig. 3.22

Se pueden añadir abreviaturas a la lista de entradas de Autocorrección. Por ejemplo introduzca dos o tres letras para representar un nombre o un nombre de compañía que escriba a menudo.



SOLUCIÓN DE PROBLEMAS

Tengo muchas palabras que contienen números, como medidas en mi documento y quiero que Word ignore estas palabras.

Seleccione **Herramientas, Opciones** y seleccione la ficha **Ortografía**. En la zona **Ignorar en** elija **Palabras con Números**. Elija [Aceptar] para cerrar el cuadro de diálogo.

Quiero editar algunas palabras en un diccionario personalizado.

Seleccione **Herramientas, Opciones** y seleccione la ficha **Ortografía**. En la zona **Diccionarios Personalizados**, seleccione el diccionario que quiere modificar y después pulse sobre el botón [Editar].

Administración de archivos y áreas de trabajo

por Rick Winter y Páttý Winter

Uno de los cambios más llamativos de Microsoft Office para Windows 95 se encuentra en lo referente a la administración de archivos. Particularmente a la hora de guardar, abrir y buscar archivos. Al trabajar con archivos, encontrará más flexibilidad, más funciones y más opciones. Como ocurre con cualquier cosa nueva, estos cambios pueden ser abrumadores al principio. Sin embargo, si emplea tiempo en aprender estas funciones de administración de archivos, se verá recompensado con mejoras importantes. Las mejoras en la gestión de archivos incluyen nombres de archivos expandidos, muchos cambios en los cuadros de diálogo de los archivos y la expansión del Resumen de Información a las propiedades del archivo. Los cuadros de diálogo Guardar y Abrir ahora tienen sus propias barras de herramientas que proporcionan visualizaciones y otras opciones. En vez de la antigua opción Archivo Buscar de la versión anterior de Microsoft Office, las funciones de búsqueda están incluidas directamente dentro de los cuadros de diálogo de los archivos.

En este capítulo aprenderá a:

- Abrir, guardar y cerrar documentos
- Seleccionar una localización en la que guardar su archivo
- Encontrar un archivo por su nombre, contenido o propiedades
- Adjuntar información identificativa a un archivo y visualizar otras propiedades del archivo
- Imprimir documentos
- Conmutar a diferentes documentos

Como trabajar con archivos

Para la mayor parte de las aplicaciones, el trabajo que se hace en pantalla está únicamente en la memoria del ordenador. Si hay un corte de corriente u ocurre cualquier otro accidente, puede perder todo el trabajo o parte de él. El proceso de guardar un archivo copia la información en memoria a un archivo en un disco blando. Se puede guardar el archivo manualmente o configurar el programa para que guarde el archivo automáticamente. Cuando se cierra un archivo, se elimina la información de la memoria del

95

◀ VÉASE "Cómo abrir un documento desde el explorador de windows", pág. 28.

◀ VÉASE "Cómo abrir un documento desde barra de tareas", pág. 27.

95

ordenador o de la pantalla. El programa sugiere guardar la información si ha hecho cambios desde la última operación de guardar, si aún no se ha guardado el archivo. Abrir un archivo supone copiar la información desde un disco a la memoria. Cuando se crea un archivo nuevo, se abre una ventana de documento nueva.

Como usar cuadros de diálogo de archivo

Además de utilizar las carpetas del Explorador de Windows y del Escritorio, es posible manejar los archivos por medio de los programas de Microsoft Office. Cuando abre, guarda o inserta un archivo, aparece un cuadro de diálogo de archivo estándar. La tabla 4.1 muestra comandos que enseñan un cuadro de diálogo de archivo. Muchas de las funciones del cuadro de diálogo de archivo son iguales tanto si usa Word, Excel o PowerPoint, como si está abriendo, guardando o insertando. La tabla 4.1 muestra un cuadro de diálogo Abrir estándar con muchas de las funciones comunes. La barra de herramientas le ayuda a organizar sus archivos. La sección inferior y el botón de comando Búsqueda Avanzada le ayudan a encontrar sus archivos.

La tabla 4.2 muestra una lista de funciones de cuadros de diálogo y resume sus usos. En las otras secciones de este capítulo se describen estas funciones más en detalle.

Fig. 4.1
El cuadro de dialogo Abrir de Excel tiene muchas funciones en comun con todos los cuadros de dialogo de archivo estándar

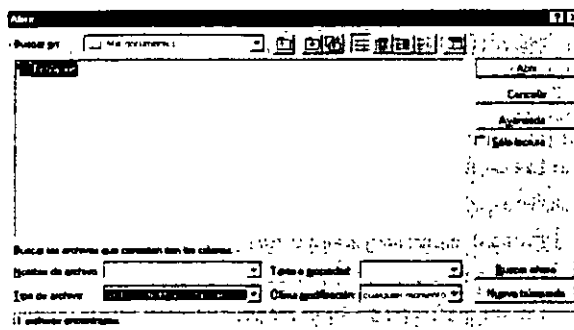


Tabla 4.1 Comandos que muestran los cuadros de diálogo

Comando	Descripción	Aplicación
Archivo. Abrir	Busca y abre un documento ya creado.	W, F, P
Archivo. Guardar	Guarda su trabajo	W, E, P
Archivo. Guardar Como	Guarda su trabajo y le permite ver o crear un archivo adicional con un nuevo nombre.	W, F, P

Tabla 4.1 Comandos que muestran los cuadros de diálogo (Continuación)

Comando	Descripción	Aplicación*
Archivo, Guardar Área de trabajo	Guarda todos los documentos abiertos de Excel en un Contenedor que le permitirá abrir múltiples documentos a la vez.	E
Insertar, Archivo	Sitúa un documento existente dentro del documento en curso.	W
Insertar, Imagen	Inserta un archivo de gráficos dentro del documento.	W, E, P
Insertar, Base de Datos, Obtener Datos	Inserta una tabla de base de datos dentro del documento.	W
Insertar, Objeto, Crear de un Archivo, Examinar	Inserta un archivo en su documento para que pueda abrir posteriormente el objeto con la aplicación que lo creó.	W, E, P
Insertar, Diapositivas del Archivo	Inserta diapositivas de otro archivo de PowerPoint.	P
Insertar, Diapositivas del esquema	Crea diapositivas de un esquema de Word o de Excel.	P
Insertar, Película	Inserta una película de vídeo en la presentación.	P
Insertar, Sonido	Inserta sonido en la presentación.	P
Herramientas, Combinar Correspondencia	Utiliza una base de datos o un archivo de hoja de cálculo Obtener Datos para combinar datos.	W

* W= Word, E= Excel, P= PowerPoint

Tabla 4.2 Funciones comunes de los cuadros de diálogo. Archivos












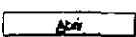





Botón/Elemento	Descripción	Uso
	Botón de ayuda	Después de pulsar en el signo de interrogación, pulse sobre la parte del cuadro de diálogo de la que quiera una explicación
	Botón Cerrar	Cierra el cuadro de diálogo sin llevar a cabo acción alguna. También se puede pulsar <Esc>

Tabla 4.2. Continuación

Botón/Elemento	Descripción	Uso
Lista desplegable	Lista desplegable Guardar En /Mirar En	Cambia la localización de sus archivos
Cuadro de la Lista	Cuadro de Lista de Contenidos	Muestra los nombres y opcionalmente otra información acerca de los archivos.
	Botón [Subir a Nivel Superior]	Va al siguiente nivel superior en la jerarquía de los contenedores de archivos.
	Botón [Buscar En Favoritos]	Va a las localizaciones que designe como favoritas con el botón [Agregar Favoritos]
	Botón [Agregar Favoritos]	Añade el archivo o la carpeta a la localización en donde tiene sus documentos más usados.
	Botón [Crear Nueva Carpeta]	Crea una nueva carpeta en la localización señalada en la lista desplegable Guardar En/Mirar En
	Botón [Lista]	Muestra los nombres de archivos sólo de las listas de contenidos.
	Botón [Detalles]	Muestra el nombre, tamaño y fecha de los archivos.
	Botón [Propiedades]	Muestra información introducida por el usuario acerca del archivo.
	Botón [Presentación preliminar]	Muestra la primera parte del archivo
	Botón [Comando] y [Configuraciones]	Muestra el menú de opciones de impresión, las propiedades de edición, clasificación, subcarpetas, y guarda las búsquedas
	Botones Guardar /Abrir/Insertar/Aceptar	Lleva a cabo las acciones requeridas con todos los parámetros del cuadro de diálogo

Botón/Elemento	Descripción	Uso
	[Botón Cancelar]	No hace nada. Vuelve al documento.
	Botón Opciones	Muestra cuadros de diálogo de las opciones de guardar.
	Botón Búsqueda Avanzada	Sugiere capacidades avanzadas de encontrar archivos.
Cuadro de Texto	Nombre de Archivo	Tecllea el nombre del archivo o usa comodines para desplegar una lista reducida de archivos.
Cuadro de Texto	Tipo de Archivo /Guardar como Tipo	Cambia el despliegue para mostrar archivos creados por programas diferentes o guarda el archivo para que otro programa pueda abrirlo.
Cuadro de Texto	Texto o Propiedad	Busca texto dentro del documento o en los parámetros de propiedad.
Cuadro de Texto	Ultima modificación	Busca archivos basado en la fecha en que fueron guardados por última vez.
Casilla de verificación	Casilla de verificación Leer solamente	Abre archivos de modo que no se pueda sobrescribir en el archivo existente
	Botón Buscar Ahora	Aplica los criterios de búsqueda y busca un archivo.
	Botón Nueva Búsqueda	Elimina cualquier criterio de búsqueda.
Casilla de verificación	Casilla de verificación Vincular o Archivo	Al insertar un archivo, adjunta instrucciones para leer el archivo en disco, pero no inserta un verdadero archivo en el documento.
Casilla de verificación	Casilla de Verificación Guardar con Documento	Al vincular un archivo, incluye el archivo completo en el documento.

Como Guardar, Abrir y Cerrar archivos

► VÉASE "Cómo cerrar y abrir documentos de word", pág. 171.





► VÉASE "Cómo guardar libros de trabajo", pág. 259.

► VÉASE "Cómo guardar una presentación", pág. 590.

Los procedimientos para guardar, abrir y cerrar archivos son similares entre sí. Ud. elige los comandos necesarios usando un botón, un ítem de menú o una tecla de método abreviado. En la mayoría de los casos, un cuadro de diálogo archivo estándar se abre y le pide información acerca del nombre del archivo, la localización (disco y carpeta) y el tipo de archivo que está usando. A veces el cuadro de diálogo no se abre. Por ejemplo, cuando guarda un archivo después de darle un nombre, el programa da por supuesto que quiere utilizar las opciones predeterminadas del cuadro de diálogo de archivo para el nombre, localización y el tipo de archivo. Se puede cambiar el nombre del archivo con el comando Guardar Como.

La tabla 4.3 muestra los distintos métodos de guardar, abrir y cerrar archivos.

Tabla 4.3 Métodos para Guardar, Abrir y Cerrar archivos

Acción	Botón	Comando del Menú	Tecla de método abreviado
Guardar		Archivo, Guardar	<Control> + <S>
Guardar Como	No hay tecla de método abreviado	Archivo, Guardar Como	
Abrir		Archivo, Abrir	<Control> + <O>
Nuevo		Archivo, Nuevo	<Control> + <N>
Cerrar		Archivo, Cerrar	<Control> + <F4>

o doble pulsación en el icono del menú

Consejo



Antes de dejar su ordenador, pulse siempre <Control> + <S> si tiene un documento en pantalla <Control> + <S> también o está de más antes de comprobar la conexión o comenzar la impresión.

Cómo guardar un documento

Si quiere volver a utilizar un documento o mantener una copia del mismo, necesitará guardar el documento. Si para crear un documento se necesita un rato (digamos que más de 15 minutos), también le será conveniente guardar el documento para protegerlo de cortes de corriente, de problemas desagradables que ocurren en los ordenadores y de darle al botón equivocado.

Se puede guardar el documento de varias maneras. Ud. puede pulsar en el botón **Guardar** de la barra de herramientas estándar, pulsar <Control> +

<S>, o elegir **Archivo, Guardar**. Cuando intente cerrar un documento no guardado o salir de un programa con un documento no guardado, puede contestar Sí a la sugerencia para guardar los cambios. Sea cual sea el méto-

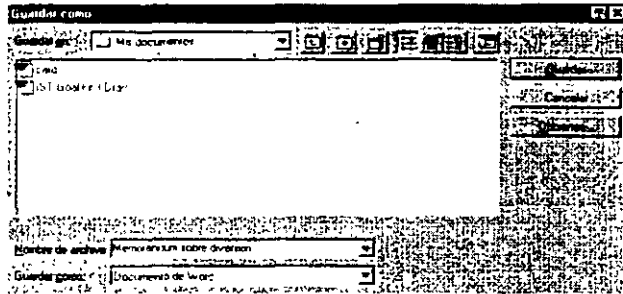


Fig. 4.2
Cuando guarda un documento por primera vez o cuando elige **Archivo, Guardar Como**, obtendrá el cuadro de diálogo Guardar Como.

do que elija, la primera vez que guarde un documento deberá entrar en el cuadro de diálogo Guardar Como, según se muestra en la figura 4.2.

Para guardar el archivo, escriba el nombre en el cuadro de texto **Nombre de Archivo**, elija una localización en la lista desplegable **Guardar En**, y elija el botón [Guardar]

Cómo asignar nombre a un documento

Las viejas reglas para dar nombre a un documento con un máximo de ocho caracteres, un punto y tres caracteres más era demasiado limitada. Estas reglas producían nombres de archivo crípticos como PLNNEG.DOC, ROCUPO.XLS, y PRBACO.PPT. Ahora puede tener nombres como *Plan de Negocios*, *Rotura de Cuentas por Porcentaje*, y *Presentación de Banda Comunitaria*. Una de las adiciones más útiles es la capacidad de utilizar espacios a la hora de dar nombres.

Los nombres válidos de archivo pueden contener ahora lo siguiente:

- Espacios
- Hasta 225 caracteres
- Todos los caracteres excepto \ / ? . : [" < > y |

95

Nota

Windows 95 no altera realmente el sistema de dar nombre a archivos de 8 + 3 de DOS. Windows 95 convierte el nombre de archivo que teclea en un nombre aceptable que consiste normalmente en los seis primeros caracteres sin espacios, una tilde y un 1 para el primer caso de ese nombre (un 2 para el segundo, etc.). Por ejemplo, el nombre de archivo Word Plan de Negocios se convierte en PLANDE~1.DOC. Cuando copia o usa este archivo en un ordenador DOS, el nombre del archivo es la versión reducida. En los sistemas Windows 95, el nombre largo se almacena con una referencia al nombre corto del archivo. Para ver los nombres cortos de archivos DOS cuando está en un cuadro de diálogo de archivo, pulse con el botón derecho del ratón, elija Propiedades, y mire en la ficha Propiedades Generales.

Los caracteres en mayúscula y minúscula se muestran generalmente tal y como los teclea, pero no es posible darle a un documento el mismo nombre en casos diferentes (no se pueden tener un archivo llamado PLAN y otro llamado plan en la misma carpeta).



Como guardar cambios en un documento con nombre

Después de guardar los cambios por primera vez, el nombre del documento aparece en la barra de título, según se muestra en la figura 4.3. Cuando que-

Fig. 4.3
La barra de título muestra el nombre del documento Donaciones (Enc-Mari)

1	Donaciones	Folio	Fecha	Monto	Total
2					
3	Individuales	35000	59500	75000	295500
4	Eventuales	44500	149500	59500	253500
5	Colectivos	0000	1000000	300000	1710000
6	Total	80000	1009000	1345000	5199000
7					

ra guardar un documento una segunda vez y continuar usando el mismo nombre, use cualquiera de los procedimientos mencionados anteriormente (el botón Guardar, <Control> + <S>, **Archivo, Guardar**, o salir del documento o programa y seleccionar Sí para guardar los cambios)

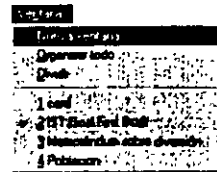
Si desea darle al documento otro nombre, elija **Archivo, Guardar como**. El cuadro de diálogo **Guardar como** aparecerá con el nombre existente en el cuadro de texto Nombre de Archivo. Edite el nombre del documento o cambie la localización en la lista desplegable **Guardar En**.

Cómo guardar todos los documentos abiertos

Si continúa abriendo documentos o creando nuevos documentos sin cerrar el anterior, tendrá más y más documentos en memoria para la aplicación en uso. Para ver una lista de documentos en memoria, elija el menú **Ventana**.

Los documentos abiertos aparecen en la parte inferior del menú, según se muestra en la figura 4.4

Fig. 4.4
Hay cuatro documentos abiertos. Para ir a un documento abierto, pulse sobre el nombre del documento.



En Word es posible guardar todos los documentos abiertos con **Archivo, Guardar Todo**. Si no ha dado nombre a uno o más documentos, el cuadro de diálogo Guardar aparecerá, sugiriéndole el nombre y localización del archivo.

Elección de una localización para guardar su archivo

Cuando se guarda un archivo, éste va a alguna parte del disco. En el más sencillo de los casos, un ordenador puede tener un disco duro dentro del mismo ordenador (unidad C:), y otra unidad de disquete (unidad A:). Sin embargo, su ordenador personal puede tener muchas unidades de disquete y discos duros adicionales, cartuchos de discos y lectores ópticos removibles, y un CD-ROM. Su ordenador también puede estar conectado a una red, dándole acceso a unidades de ordenadores que no están conectados directamente con su ordenador personal.

Cuando entra por primera vez en un cuadro de diálogo archivo estándar, verá una lista desplegable. En el cuadro de diálogo Guardar verá la lista **Guardar En**. En el cuadro de diálogo Abrir, verá la lista **Buscar En**. Una sugerencia en la lista representa la localización o contenedor en curso donde puede guardar o abrir archivos. Debajo de la lista desplegable hay una gran área abierta (el cuadro de la lista de contenidos) que le muestra el contenido del contenedor en curso. El contenido puede ser otros contenedores o archivos.

Cuando selecciona la lista desplegable tal y como se muestra en la figura 4.5, se le proporciona una vista general de los contenedores disponibles para que guarde o abra archivos. Se despliega una lista de localizaciones en cascada. Esta lista representa la jerarquía de localizaciones en donde puede encontrar o almacenar archivos. Esta jerarquía puede tener muchos niveles diferentes; cada nivel se representa por una sangría. Cuando elige un nivel de la lista desplegable, el contenido del nuevo nivel se muestra en el cuadro de lista de contenidos. Si la lista tiene más ítems de los que caben, verá una barra de desplazamiento vertical a la derecha del cuadro de lista. Si desea volver a un nivel superior, puede elegir las listas desplegables **Guardar en** o **Buscar en** o pulsar en el botón Subir nivel superior.

La jerarquía se muestra en forma diagramática en la figura 4.6. En el nivel superior está el Escritorio indicando que todas las localizaciones potenciales parten de esta ubicación. Las dos ramas principales que parten del Escritorio son Mi PC y Entorno de Red, indicando que está trabajando con un archivo que es parte física del sistema de su ordenador o de otros ordenadores que estén conectados al suyo por medio de una red.

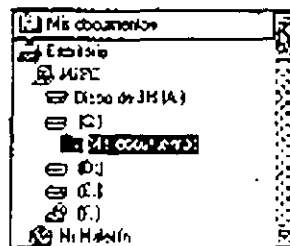
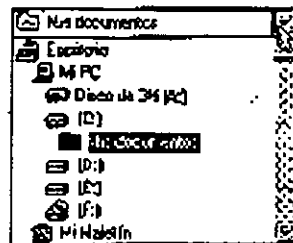


Fig. 4.5
Localice el lugar en donde quiera trabajar con los archivos en la lista desplegable **Buscar En**.

Si selecciona Entorno de Red en la lista desplegable, verá una lista de los ordenadores de su red en el cuadro lista de contenidos según se muestra en la figura 4.7. Si realiza entonces una doble pulsación en cualquier ordenador

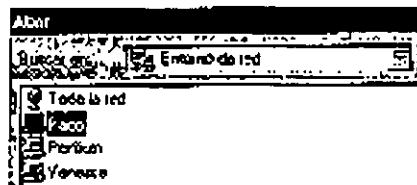
además del suyo, aparecerá una lista de carpetas compartidas en el cuadro lista de contenidos. Si realiza una doble pulsación en una carpeta podrá ver las subcarpetas y archivos en el cuadro lista de contenidos.

Fig. 4.6
La jerarquía para almacenar archivos contiene muchos niveles diferentes.



Bajo Mi PC en la lista desplegable están los distintos aparatos de almacenamiento conectados al ordenador, tales como una unidad de disquete de 3 1/2" (A:) y otro de discos blandos de 5 1/4" (B:), mostrados con anterioridad en la figura 4.5. Cuando pulsa sobre una de estas unidades, realizará su trabajo con archivos de disquetes. Si va a guardar un archivo, puede optar entre teclear A: o B: antes del nombre del archivo en el cuadro de texto Nombre de archivo.

Fig. 4.7
Hay dos ordenadores anidados en esta red de ordenadores.



El icono de un disco duro (normalmente fijo dentro de su ordenador) es ligeramente diferente en la figura 4.5 y va normalmente indicado con una C:. Como el disco duro tiene mucho más espacio, los archivos van normalmente colocados dentro de carpetas. En la figura 4.5, el resaltado está en la carpeta Rechazo de la Violencia Organizada, que es una subcarpeta de Datos.

Para navegar por la jerarquía de los ordenadores, discos y carpetas, puede optar por cualquiera de los siguientes pasos:

- Usar el cuadro de lista desplegable **Guardar En** o **Buscar En**.
- Realizar una doble pulsación en una carpeta del cuadro de lista de contenidos para ver el contenido de la carpeta.
- Pulsar sobre la carpeta con el botón derecho del ratón y pulsar en [Abrir] para ver el contenido de la carpeta.
- Pulsar sobre el botón [Subir nivel superior] para volver a un nivel superior en la jerarquía.
- Pulsar sobre el botón [Crear] nueva carpeta para crear una carpeta debajo del contenedor o carpeta identificada en la lista desplegable **Guarda En**.
- Si conoce la localización de la carpeta que quiere abrir o guardar, puede



teclea la jerarquía de discos y carpetas en el cuadro de texto **Nombre de Archivo** como en este ejemplo: C:\Office95\Excel\Impuestos95



Cómo abrir un documento

Existen más opciones para abrir un documento que para guardarlo. Algunas de estas opciones fueron descritas en el capítulo 2, "Iniciarse e Identificar Partes de la Pantalla". Este apartado contiene un resumen de las opciones existentes para abrir un documento.

Desde fuera de un programa, siga uno de los siguientes pasos:

- Desde la barra de acceso directo de Microsoft Office, pulse sobre el botón [Abrir Documento], seleccione la localización y elija el archivo. También puede personalizar la barra de acceso directo para que contenga archivos y carpetas. Ver el capítulo 34, "Personalización del Escritorio, Barras de Herramientas y Menús". Después de pulsar en el botón [Abrir Documento], entrará en un cuadro de diálogo Abrir.
- Desde la barra de tareas de Windows, elija **Inicio**, después **Documentos** y pulse sobre uno de sus últimos 15 documentos.
- Desde la barra de tareas de Windows, elija **Inicio**, después **Programas**, y busque el archivo por medio del Explorador de Windows.
- Minimice o cierre todos los programas abiertos y realice una doble pulsación en un icono del escritorio, como [Mi PC], [Entorno de Red], o carpetas de acceso directo y navegue por las ventanas para encontrar su archivo.
- Desde la barra de tareas de Windows, elija **Inicio** y después **Abrir documento de Office**.
- Pulse cualquier tecla de método abreviado que haya asignado a un archivo usando la ficha Acceso Directo en el cuadro de diálogo Propiedades de un icono de método abreviado. (Véase el apartado siguiente "Cómo usar Búsqueda avanzada de archivos".)

Desde dentro de un programa, siga uno de los siguientes pasos:

- Pulse sobre el botón [Abrir].
- Pulse <Control> + <O>
- Elija **Archivo, Abrir**.

Si elige cualquiera de estos pasos entrará en el cuadro de diálogo Abrir. Escriba el nombre del archivo en el cuadro de texto **Nombre de archivo** y elija la localización en la lista desplegable **Buscar en**.

También es posible encontrar fácilmente uno de los últimos archivos con los que ha trabajado. En la parte inferior del menú **Archivo** hay una lista de los archivos usados más recientemente (véase la fig. 4.8). Si no aparece ningún disco o carpeta con el nombre del archivo, éste procede de la carpeta activa. Elija el nombre del archivo para abrirlo.

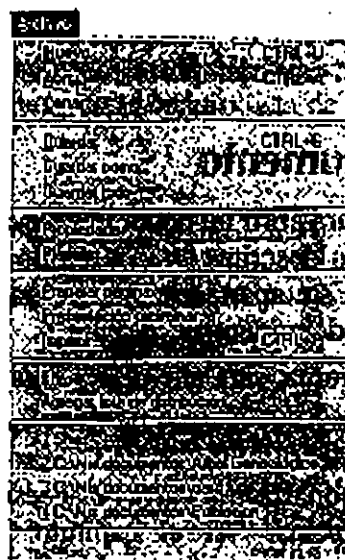
► VÉASE "Cómo abrir un documento", pág. 172.



◄ VÉASE "Cómo iniciar programas", pág. 26.

► VÉASE "Cómo abrir una presentación existente", pág. 590.

Fig. 4.8
Los últimos
archivos que
abre aparecen
en la parte
inferior del
menú Archivo.



SOLUCIÓN DE PROBLEMAS

Cuando busco un archivo, el nombre del archivo no aparece en el cuadro de lista de contenidos.

El nombre del archivo puede no aparecer en un cuadro de lista de contenidos por varias razones. En primer lugar, es posible que sea necesario usar la barra de desplazamiento para acceder a más archivos de la lista. En segundo lugar, el archivo puede no ser del tipo adecuado. En el cuadro de lista **Tipo de Archivo**, seleccione **Todos los Archivos**, y mire si su archivo está en la lista. En tercer lugar, es posible que tenga que reemplazar el disco y la carpeta. Por último, si todo lo demás falla, busque el archivo usando las funciones descritas en el siguiente apartado, "Cómo buscar Archivos".

No puedo acceder a un archivo del ordenador de otra persona de la red.

En el Escritorio de Windows elija Entorno de Red, y después elija el nombre del ordenador de la otra persona. El otro ordenador tiene que tener archivos compartidos para permitirle acceder a sus archivos.

No puedo leer un archivo del CD-ROM.

Asegúrese de que el CD-ROM está instalado correctamente. Si no lo ha hecho, instale el CD-ROM eligiendo Inicio en el barra de tareas de Windows. Entonces elija Instalar, Panel de Control, Agregar Nuevos Programas, y siga al Asistente para Programas. Además, asegúrese de que tiene el reproductor de CD-ROM encendido y que el CD apropiado está en la unidad de CD.

Cómo buscar archivos

Es posible utilizar el botón [Abrir Documento] en la Barra de acceso Directo de Microsoft Office para abrir un documento e iniciar el programa que lo creó al mismo tiempo. Después de pulsar en el botón Abrir Documento, aparecerá el cuadro de diálogo **Abrir**. Este cuadro de diálogo es el mismo que aparece cuando está en el programa y usa **Archivo, Abrir**.

Una vez está en el cuadro de diálogo Abrir, tiene muchas opciones para restringir o expandir su búsqueda. Es posible reemplazar la organización y



orden de los archivos, utilizar comodines en los nombres de los archivos, elegir un tipo de archivo o identificarlo por su contenido, fecha u otras propiedades. Estas opciones del cuadro de diálogo Abrir sustituyen a la función Buscar Archivo de la versión anterior de Microsoft Office.

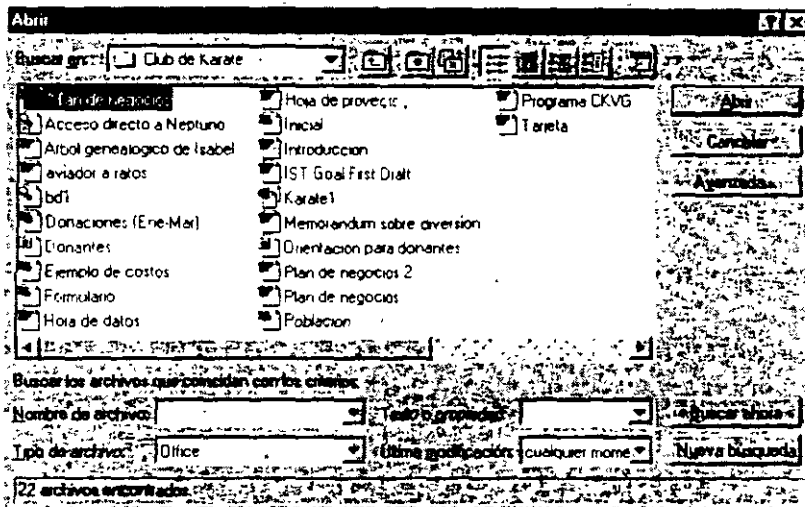


Fig. 4.9 Pulse sobre el botón [Lista] para acceder a los nombres de archivos.

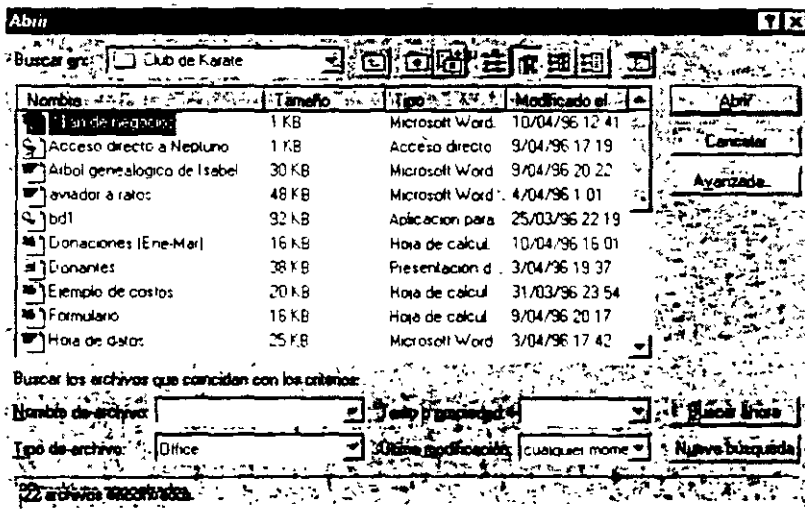


Fig. 4.10 Pulse sobre el botón [Detalles] para acceder a los nombres de archivos, el tamaño, el tipo y la fecha.

Cómo abrir listas de archivos

Después de seleccionar un ordenador, una unidad y una carpeta por medio de la lista desplegable **Buscar En**, verá una lista de archivos (y posiblemente otras carpetas) en el cuadro de lista de contenidos. Ud. dispone de cuatro opciones para ver esta lista de archivos. Las figuras 4.9 a la 4.12 muestran las diferentes modalidades de ver los archivos. Pulse sobre uno de los siguientes botones:





Botón	Nombre	Propósito
	Lista	Muestra nombres de archivos con un icono del tipo de archivo solamente.
	Detalles	Muestra nombres de archivos, icono del tipo de archivo, tamaño del archivo, tipo de archivo y hora y fecha de modificación.
	Propiedades	Muestra información resumida añadida por el usuario como título, autor, palabras-clave, etc.
	Presentación preliminar	Muestra qué aspecto tiene la primera parte del archivo.

Fig. 4.11

Pulse sobre el botón [Propiedades] para acceder a los nombres de archivo y la información acerca del archivo proporcionada por el usuario.

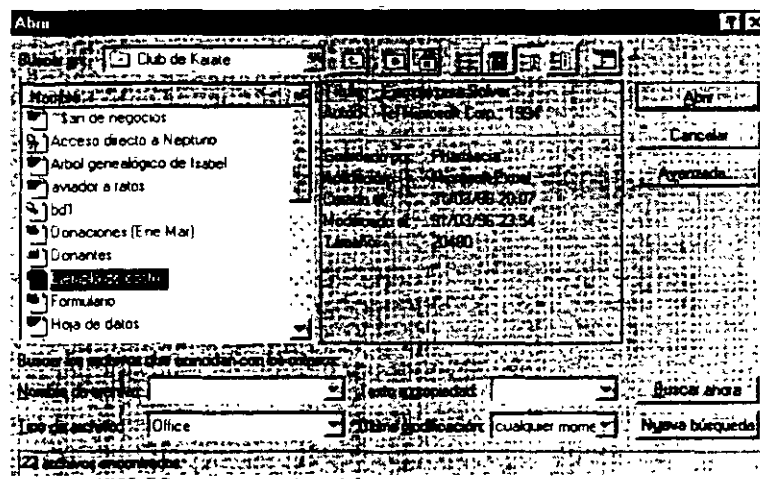
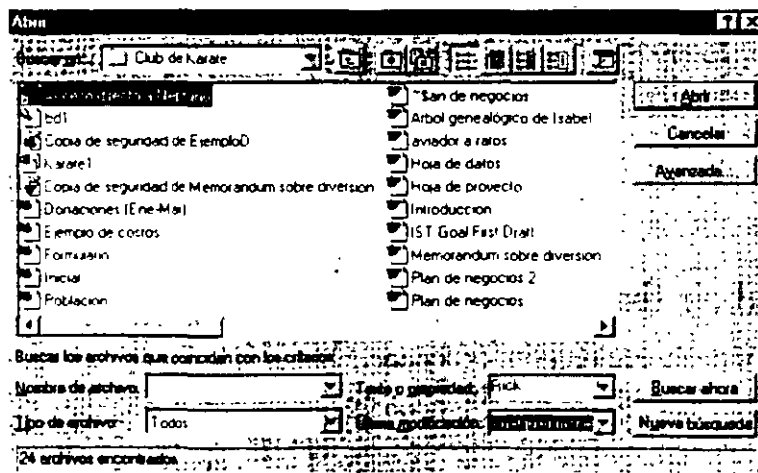


Fig. 4.12

Pulse sobre el botón [Presentación preliminar] para ver una imagen del archivo.



Cómo clasificar la lista de archivos

Cuando elige el botón [Detalles], la lista de archivos aparecerá con el nombre del archivo, el tamaño, el tipo y la fecha y hora de modificación. La parte



superior del cuadro de lista de contenidos muestra los encabezados de columna Nombre, Tamaño, Tipo y Modificación, tal y como se muestra en la figura 4.10. Pulse sobre cualquiera de estos encabezados de columna para clasificar la lista de forma ascendente según esa categoría. Vuelva a pulsar en el encabezado de columna para clasificar la lista de forma descendente según esa categoría. Por ejemplo, para clasificar los archivos según la fecha, pulse sobre el encabezado Modificación. Pulse otra vez para clasificar los archivos con la última fecha en la parte superior de la lista.

Cómo usar los comodines para reducir la lista de archivos

Si su lista de archivos es particularmente larga, posiblemente quiera limitarla según el nombre de archivo. Dos comodines pueden ayudarle a limitar la lista según los caracteres del nombre del archivo: * (asterisco) y ? (signo de interrogación). El asterisco significa sustituir cualquier número de caracteres. El signo de interrogación significa sustituir un carácter. En el cuadro de texto **Nombre de Archivo** del cuadro de diálogo Abrir puede teclear el texto con comodines y después elegir el botón de comando **Buscar Ahora**. La lista siguiente muestra varios ejemplos.

Para eliminar el comodín y todas las restricciones de la lista de archivos, elija el botón de comando **Nueva Búsqueda**.

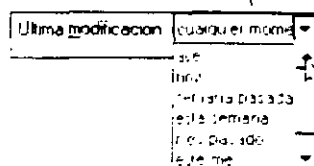
Consejo

A diferencia del comodín de DOS, es posible utilizar múltiples asteriscos dentro de los nombres de archivos.

Tipo	A Mostrar
B*	Todos los archivos que empiezan por B
Rev??	Rev95, Rev96, Rev97
Don	Razones para Donar, Donaciones YTD, Orientación a Donantes, Curriculum de Don Campbell

Cómo buscar un archivo por la fecha

Cuando la lista es larga y Ud. no se acuerda de la fecha exacta, pero sí de cuándo aproximadamente trabajó con él la última vez, puede limitar la lista de archivos para que muestre únicamente aquellos archivos modificados en un período determinado. Use la lista desplegable **Última Modificación** y pulse sobre el botón de comando **Buscar Ahora** para limitar la lista de archi-



vos por tiempo. La figura 4.13 muestra que las opciones de **Última Modificación** incluyen hoy, ayer, la semana pasada, esta semana, el mes pasado y este mes. Para hacer que la lista de archivos muestre todas las fechas, seleccione cualquier tiempo.

Fig. 4.13 Para acceder a archivos que cumplan criterios de fecha, elija la lista desplegable **Última Modificación**.

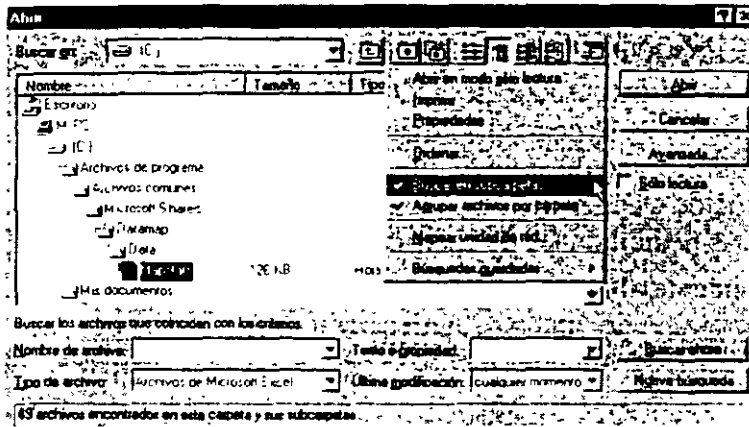


Fig. 4.15 El cuadro lista de contenidos muestra carpetas y archivos dentro de carpetas que satisfacen sus criterios.

Tabla 4.4 Tipos de archivo que puede abrir en una aplicación distinta

Aplicación que abre el Archivo	Tipo de Archivo	Resultado
Word	Hoja de cálculo de documento Excel	Tabla Word (puede ser un combinación de datos.
Aplicación que Abre el Archivo	Tipo de Archivo	Resultado
PowerPoint	Word (esquema)	Encabezado 1 = título de diapositiva, Encabezado 2, 3 = puntos y subpuntos
PowerPoint	Hoja de calculo de Excel	Cada fila se convierte en el título de la diapositiva

Nota

Si la lista Tipo de archivos tiene más opciones de las que puede ver, aparecerá una barra de desplazamiento vertical, permitiéndole desplazarse a otros ítems de la lista y seleccionar el tipo de archivo que desee.

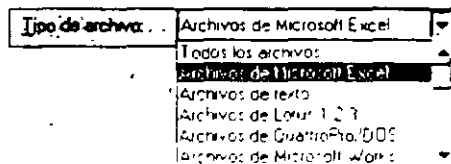


Fig. 4.16 Use **Tipo de archivos** para abrir o guardar archivos que no son del tipo predeterminado en su aplicación.

Esto resulta útil si quiere abrir un archivo creado por un programa diferente o si quiere guardar un archivo con un formato diferente para utilizarlo con otro programa. La figura 4.16. muestra algunas de las opciones Archivo de Tipo para el cuadro de diálogo Abrir de Excel. Estos tipos incluyen archivos de texto, diferentes formatos de hojas de cálculo, antiguas versiones de Excel y formatos de archivos de bases de datos.

Normalmente modificará el tipo de archivo cuando desee convertir un archivo de un tipo a otro, como una hoja de cálculo (Lotus 1-2-3 a Excel), un documento de procesador de texto (WordPerfect a Word) o una base de datos (dBASE a Access). En algunos casos, sin embargo, es posible que quiera abrir un tipo de archivo diferente. Por ejemplo, puede abrir un archivo de hoja de cálculo de Excel en un documento Word. La tabla 4.4 muestra algunas posibilidades de apertura de distintas clases de archivos.

Cuando intente abrir un archivo de tipo diferente, si su aplicación no puede convertir el tipo del archivo, aparecerá un mensaje de error diciendo que el formato del archivo no es válido.

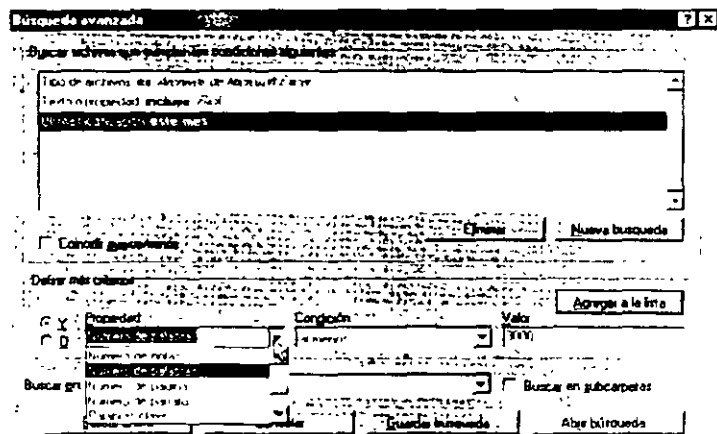
Cómo utilizar búsqueda avanzada de archivos



Aunque la opción **Texto** o **Propiedad** busca el contenido del archivo o cualquiera de sus propiedades, puede limitar la búsqueda a una sola propiedad o utilizar muchas propiedades al mismo tiempo. Siga los siguientes pasos:

1. Pulse sobre el botón [Abrir] de la barra de herramientas estándar de su programa o pulse sobre el botón [Abrir Documento] de la barra de acceso directo de Microsoft Office.
2. Si quiere, rellene las opciones **Nombre de Archivo**, **Texto** o **Propiedad**, **Tipo de Archivos** y **Última Modificación**, tal y como se describió anteriormente.
3. Elija el botón **Búsqueda Avanzada**. El cuadro de diálogo Búsqueda Avanzada aparecerá con las elecciones que hizo en el cuadro de diálogo [Abrir].

Fig. 4.17
Para buscar en la lista desplegable Propiedades, elija una propiedad



4. Elija la lista desplegable **Propiedades** y seleccione la propiedad que quiera, según se muestra en la figura 4.17.
5. En el cuadro de texto **Valor**, escriba el texto que busca.
6. Elija el botón **Agregar a la Lista**.
7. Si desea añadir alguna otra condición, elija el botón opción **Propiedad Y** (las dos condiciones deben cumplirse) o el botón **opción O** (una de las dos condiciones debe cumplirse) y complete los pasos 2-6.

Tabla 4.5 Otras opciones del cuadro de diálogo búsqueda avanzada

Opción	Descripción
Casilla de verificación Coincidir Todas las Formas de palabras	Para contenidos, comentarios y otros ítems de búsqueda, puede encontrar archivos que casen con las diferentes formas de la palabra. Por ejemplo, si el Valor es ser, la búsqueda incluye ser, soy, eres, es.
Casilla de verificación Coincidir Mayús/minus	Para contenidos, comentarios y otros ítems de la opción Propiedad , puede encontrar sólo los ítems que coincidan con la primera letra de la palabra (mayúscula o minúscula) según haya tecleado las opciones en el cuadro de texto Valor.
Botón Eliminar	Pulse sobre uno de los ítems en Buscar Archivo que cumplan las condiciones siguientes y después elija el botón Eliminar para borrar los criterios.
Botón Nueva Búsqueda	Elimina todos los criterios y empieza de nuevo.
Lista desplegable Condición	Esta lista cambia según la Propiedad seleccionada y le permite encontrar coincidencias que contienen la opción Valor por completo, una parte de ella, uno de los ítems de la opción Valor , ítems mayores o menores que una fecha, y otras cosas.
Lista desplegable Buscar En	Es lo mismo que Buscar En el cuadro de diálogo Abrir; le permite cambiar la localización de su búsqueda.
Casilla de verificación Buscar en Subcarpetas	Es lo mismo que la opción de menú en el menú Comandos y Configuraciones del cuadro de diálogo Abrir; le permite incluir archivos de la carpeta en uso y todos las subcarpetas que desee.
Botón Guardar Búsqueda	Guarde cualquier criterio de búsqueda con un nombre para recuperarlo más tarde.
Botón Abrir Búsqueda	Elija una de las búsquedas con nombre creadas por Ud. con el comando Guardar Búsqueda .
Botón Cancelar	No ejecuta ninguna de las opciones Búsqueda Avanzada y vuelve al cuadro de diálogo Abrir tal y como lo dejó Ud.

Cuando quiera ver sus archivos o carpetas Favoritos, pulse sobre el botón [Buscar] En Favoritos en el cuadro de diálogo de cualquier archivo.



Cómo Copiar y Mover archivos, Imprimir y otras funciones ocultas

Cuando se encuentra en un cuadro de diálogo Abrir o Guardar, dispone de otras opciones que están ocultas y que no se presentan como elección en el cuadro de diálogo. Puede acceder a estas opciones seleccionando un archivo o archivos y realizando después una pulsación con el botón derecho del ratón (véase la fig. 4.19).

Puede seleccionar archivos de varias maneras:

- Pulse sobre el nombre de un archivo.
- Mantenga presionada la tecla <Control> y pulse sobre varios archivos para seleccionar más de un archivo.
- Pulse sobre el primer archivo, mantenga presionada la tecla de <Mayús> y pulse sobre el último archivo para seleccionar un grupo de archivos adyacentes.

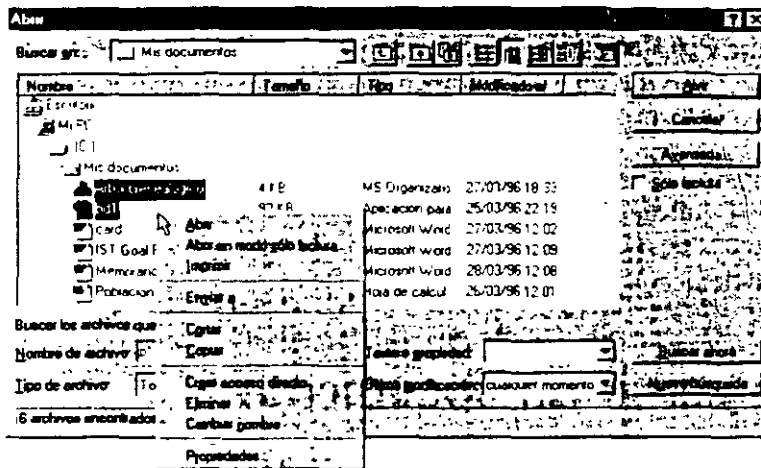


Fig. 4.19 Pulsando con el botón derecho del ratón en un archivo le proporciona más opciones de gestión de archivos.

Después de seleccionar su archivo o archivos, pulse con el botón derecho del ratón en parte de la selección y elija uno de los siguientes procedimientos:

- Elija **Abrir** para abrir todos los archivos seleccionados.
- Elija **Abrir Leer Solamente** para abrir los archivos, pero necesitará asignarles un nuevo nombre si desea guardarlos.
- Elija **Imprimir** para imprimir los archivos.

Nota

El Maletín es una nueva función de Windows 95 que le permite mantener sus documentos actualizados entre un ordenador de mesa y uno portátil.

- Elija **Enviar A.** después elija una unidad de disquete para copiar el archivo, Receptor de Fax para enviar el archivo por fax, Receptor de Correo para enviar una versión de correo del archivo, o Mi Maletín.
- Elija **Cortar** para mover archivos al Portapapeles (y eliminar los archivos originales).
- Elija **Copiar** para copiar archivos al Portapapeles (y mantener los archivos originales en su lugar).
- Elija **Crear Acceso Directo** para crear otro icono en el directorio que es un atajo al archivo (pulse con el botón derecho del ratón en el nombre abreviado y elija **Propiedades** para dar a la abreviatura una combinación de teclas. El nombre abreviado comienza con "abreviatura De".)

Nota

Si borra archivos accidentalmente, puede recuperarlos si no ha vaciado la Papelera de Reciclaje. Con todos los programas minimizados, elija Papelera de Reciclaje en el Escritorio de Windows. Elija el nombre del archivo y elija **Archivo, Restaurar.**

- Elija **Eliminar** para borrar los archivos de la Papelera de Reciclaje.
- Elija **Cambiar Nombre** para dar a un archivo un nuevo nombre.
- Elija **Propiedades** para asignar una combinación de teclas a un archivo de acceso directo, o ver información de archivos como nombre de archivo DOS, fecha de creación, atributos de archivo, información resumen y estadística.

Si elige **Cortar** o **Copiar**, elija la localización a donde quiera que vaya el archivo y después pulse con el botón derecho del ratón y elija **Pegar** en el menú de acceso directo.

Cómo cerrar un documento

◀ VÉASE "Cómo cerrar programas", pág. 36

Puede cerrar un documento de varias maneras:

- Realice una doble pulsación en el icono del menú de Control situado en la esquina superior izquierda de la ventana del documento, tal y como se muestra en la figura 4.20.
- Pulse una vez en el icono del menú de Control y elija **Cerrar** del menú desplegable.
- Elija **Archivo, Cerrar.**
- Pulse <Control> + <W>
- Pulse sobre el botón [Cerrar] del documento (X)

Si ha guardado sus últimos cambios en el documento, el documento se cierra. Si ha hecho cambios desde que guardó por última vez o no ha guardado aún, se le avisa de que guarde el documento, según se muestra en la figura 4.21.

Atención

Asegúrese de que pulsa en el botón Cerrar del documento y no en el botón Cerrar del programa. Si el documento está maximizado, el botón Cerrar del documento está en la misma fila que la barra del menú. Si el documento no está maximizado, el botón Cerrar está en la parte derecha de la barra de título del documento.

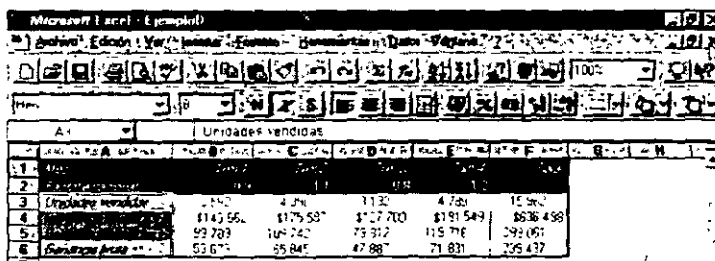


Fig. 4.20
Pulse sobre el botón [Cerrar] para cerrar un documento

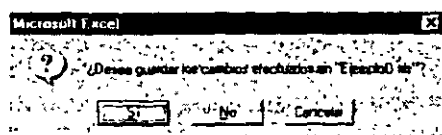


Fig. 4.21
Si intenta cerrar un documento no guardado, el cuadro de diálogo le preguntará si quiere guardar los cambios

Elija uno de los siguientes botones:

- Elija [Sí] para guardar los cambios. Si ya le ha dado al documento un nombre, el documento se cierra. Si no le ha dado un nombre al documento, aparece el cuadro de diálogo Guardar Como.
- Elija [No] para no guardar los cambios. Perderá cualquier cambio que haya hecho desde la última vez que guardó, o el documento entero si no lo ha guardado nunca.
- Elija [Cancelar] para volver al documento sin guardar el documento o salir de él.

Si no ha guardado los últimos cambios en su archivo y lo cierra usando cualquier método normal, incluida la salida de la aplicación, el aviso mostrado en la figura 4.21 le pregunta si quiere guardar los cambios. Si elige Sí, el cuadro de diálogo Guardar Como aparecerá si no ha dado nombre al archivo.

Nota

Obviamente, no obtendrá ningún aviso para guardar su documento si Ud. o la madre naturaleza corta el suministro eléctrico del ordenador.

Cómo iniciar un nuevo documento

En Excel o Word, cuando se pulsa en el botón Nuevo o pulsa <Control> + <N>, se abre una ventana de documento vacía. En PowerPoint, se abre una nueva presentación con la pregunta qué tipo de diapositiva desea. Si usa el



◀ VÉASE “Cómo iniciar un documento desde la barra de acceso directo”, pág. 29

▶ VÉASE “Cómo usar los asistentes para plantilla”, pág. 245

▶ VÉASE “Comprender documentos maestros y plantillas”, pág. 571

▶ VÉASE “Cómo crear una presentación usando un asistente”, pág. 578

comando del menú **Archivo, Nuevo** en un programa o pulsa en el botón [Inicio de Nuevo] Documento en la Barra de acceso directo de Microsoft Office. el cuadro de diálogo Nuevo aparecerá con una serie de fichas en la parte superior, tal y como se muestra en la figura 4.22. Las fichas presentan categorías generales de archivos que puede crear. En cada ficha hay archivos que son plantillas o asistentes. Las plantillas son archivos que pueden haber almacenado formatos, macros, estilos, texto y diversos menús y barras de herramientas. Después de abrir una plantilla con el comando **Archivo, Nuevo**, aún será necesario darle al documento un nombre para guardarlo. Los asistentes son una serie de cuadros de diálogo que le guían en los pasos a seguir para crear un documento o ejecutar una función.

SOLUCIÓN DE PROBLEMAS

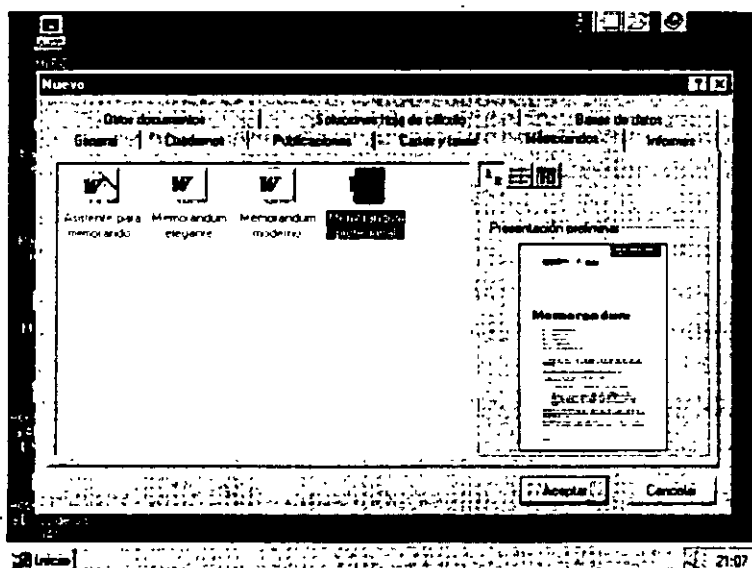
Cuando elijo subcarpetas con el botón Comandos y Configuración no veo ninguna subcarpeta.

No se ven subcarpetas con la opción lista seleccionada. Deberá pulsar en los botones Detalles, Propiedades o Presentación preliminar. Otra posibilidad es que no tenga ningún archivo en otras áreas que coincidan con sus criterios de búsqueda.

Sé que el archivo está en la carpeta, pero no lo veo.

Es posible que tenga que usar la barra de desplazamiento en la parte inferior o derecha del cuadro de lista de contenidos. Otra posibilidad es que tenga criterios de búsqueda que excluyan su archivo. Pulse sobre el botón **Nueva Búsqueda** en un cuadro de diálogo de archivo. Asegúrese de que su tipo de archivo es correcto en la lista desplegable **Tipo de Archivos**.

Fig. 4.22
El botón [Inicio de Nuevo] Documento] presenta el cuadro de diálogo Nuevo con plantillas y asistentes para las aplicaciones de Microsoft Office



Cómo usar las opciones guardar

Aunque la mayor parte de las aplicaciones exigen que guarde sus archivos, Word y Excel le permiten crear copias de seguridad así como establecer un

de guardado automático. Aún así, debería utilizar el botón [Guardar], el comando del menú o la tecla de método abreviado para guardar a menudo, especialmente después de emplear una significativa cantidad de esfuerzo en darle al documento el aspecto que desea o antes de llevar a cabo una operación de envergadura en el archivo (como una corrección ortográfica, una clasificación, una sustitución, un formato automático o una importación). Puede utilizar la función Guardar Como para guardar diferentes revisiones del mismo archivo hasta que el proyecto esté completo.

Un archivo de seguridad tiene un icono diferente y el nombre comienza con *Copia de Seguridad de* (véase la fig. 4.23). Ud. crea un archivo de seguridad después de que haya guardado el archivo al menos una vez. Cuando guarda el archivo la segunda vez y todas las veces subsiguientes, el procedimiento de copia de seguridad asigna nuevo nombre al viejo archivo en el disco con el mismo nombre del archivo, pero con un prefijo *Copia de Seguridad*. El documento en pantalla se guarda con el nombre original del archivo. Si necesita utilizar su archivo de seguridad cambie el tipo de archivo a Todos los Archivos o escriba **Copia de Seguridad *** en el cuadro de texto **Nombre de Archivo**. Para iniciar el procedimiento de Copia de Seguridad, elija el comando **Archivo, Guardar Como** y después elija el botón de comando **Opciones**, y compruebe la casilla de verificación **Crear Siempre Copia de Seguridad**.

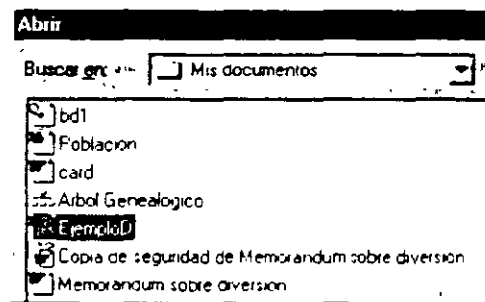


Fig. 4.23
El icono de tipo de archivo cambia para mostrar un archivo de seguridad.

En Word y Excel, puede fijar la función de guardado automático de modo que el programa guarde su trabajo a intervalos de tiempo especificados por Ud. En Word, fije la función de guardado automático de esta manera:

1. Elija **Herramientas, Opciones**
2. Pulse sobre la ficha **Guardar**
3. Compruebe la casilla de verificación **Autoguardar** cada e introduzca los parámetros de Minutos tecleando o usando los botones de incremento.

Si lo desea, para cambiar la localización donde se guardan los archivos, continúe como sigue, o salte al paso 7:

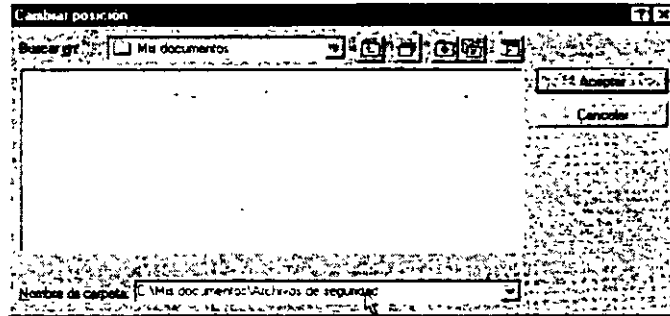
4. Pulse sobre la ficha de **Localización** de archivos
5. En el cuadro de lista **Tipos de archivo**, pulse sobre **Autoguardar** archivos y elija el botón **Modificar**.
6. En el cuadro de diálogo **Modificar localización**, escriba el nombre de la localización donde quiera guardar los archivos de **Autoguardado** en el

cuadro de texto **Nombre** de carpeta, tal y como se muestra en la figura 4.24 y elija [Aceptar].

7. Elija el botón [Cerrar] en el cuadro de diálogo Opciones.

Fig. 4.24

Escriba el nombre de la carpeta de seguridad el cuadro de texto **Nombre de Carpeta**.

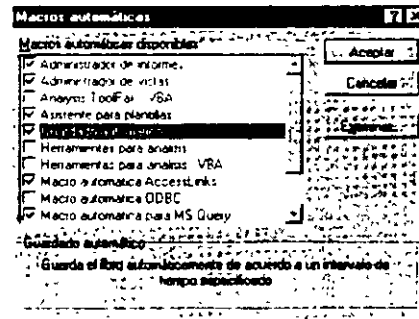


Fije la función Autoguardar en Excel de esta manera:

1. Elija **Herramientas, Macros Automáticos Disponibles**. El cuadro de diálogo Macros Automáticos aparecerá, tal y como se muestra en la figura 4.25.

Fig. 4.25

Active la opción Autoguardar en el cuadro de diálogo Macros Automáticos para activar el guardado automático.



Nota

El cuadro de diálogo Macros Automáticos puede tener un aspecto distinto en su pantalla debido a su instalación.

2. Active la casilla de verificación Autoguardar y elija [Aceptar].
3. Para activar **Autoguardar**, elija **Herramientas, Autoguardar**. El cuadro de diálogo Autoguardar aparecerá.
4. Active la casilla de verificación **Autoguardar** cada y rellene el parámetros **Minutos**.
5. Elija [Aceptar] para volver al documento.

Cuando reinicia Word tras un problema de corriente, el archivo puede aparecer automáticamente en pantalla debido a la función de guardado automático. En Excel, el programa le avisa para que le dé un nombre al archivo si todavía no lo ha guardado.

Cómo usar las propiedades de archivo

En la antigua versión de Windows el nombre de los archivos estaba limitado a 8 caracteres más una extensión de tres caracteres. Windows 95 y Microsoft Office han ampliado notablemente la capacidad hasta 255 caracteres. Si necesita algo más que el nombre de su archivo para identificar un documento, ahora dispone de muchas propiedades que puede rellenar en la función de nuevas propiedades.



Cómo ver y editar las propiedades de archivo

Puede introducir o ver propiedades de archivo de muchas maneras. Las listas siguientes describen algunas maneras de usar las propiedades de archivo.

Para ver y modificar las propiedades de archivo:

- Con el archivo abierto en Word, Excel, o PowerPoint, elija **Archivo, Propiedades**.
- En un cuadro de diálogo, seleccione el archivo y pulse sobre el botón [Comandos] y [Configuración]. Después elija **Propiedades** en el menú.

Para ver las propiedades únicamente:

- En un cuadro de diálogo, seleccione el archivo y pulse sobre el botón [Propiedades]. Las propiedades del archivo seleccionado aparecen en una área pequeña por la que puede moverse con una barra de desplazamiento vertical, tal y como se muestra en la figura 4.26.

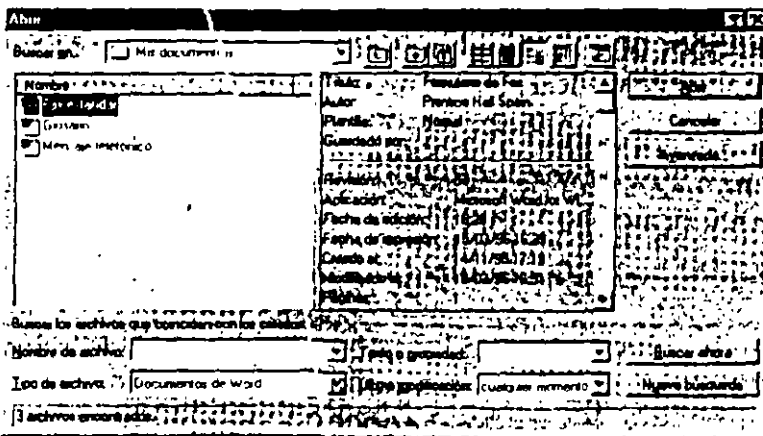
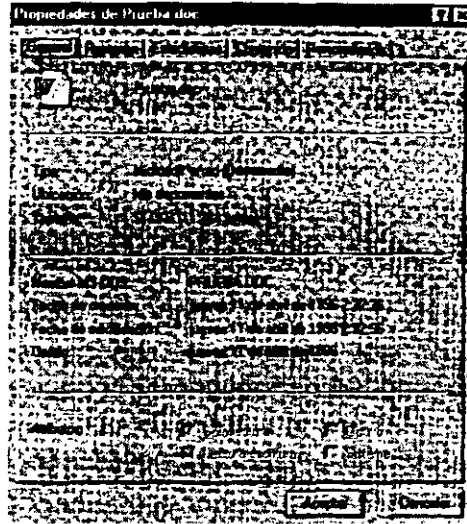


Fig. 4.26 Desplácese en el cuadro de lista de contenidos para ver más propiedades.

Sólo puede imprimir las propiedades de los archivos Word. Elija **Archivo, Imprimir** y en la lista desplegable **Imprimir** elija **Resumen**.

Fig. 4.27

La página General en la hoja Propiedades muestra información acerca del archivo.



Un repaso a las propiedades del archivo

Dependiendo de cómo haya introducido las propiedades de archivo, verá un cuadro de diálogo similar al que se muestra en la figura 4.27. En el cuadro de diálogo Propiedades aparecerán hasta cinco fichas. Las fichas se describen en los siguientes apartados.

Ficha general

La figura 4.27 muestra la página General de un documento de Word. El programa ha introducido la información en esta ficha. La información incluye lo siguiente:

Ficha Resumen

La figura 4.28 muestra la página Resumen, donde puede introducir información acerca del archivo para ayudarle a organizar el archivo y a localizarlo más tarde. Elija cualquiera de los cuadros de texto (**Título, Asunto, Autor, Director, Compañía, Categoría, Palabra-Clave o Comentarios**) introduzca el texto que desee. Aunque puede introducir más caracteres, sólo podrá ver 42 en cada uno de los campos excepto Comentarios. El campo **Comentarios** tiene una área de visión grande y puede utilizar la barra de desplazamiento vertical para ver más texto del que cabe en el cuadro de lista.

Ficha Estadísticas

La figura 4.29 muestra la página Estadísticas. Esta información le dice cuándo se creó su archivo, y cuándo se modificó, se accedió a él o se imprimió;

Item	Descripción
Nombre de archivo	En la parte superior de la ficha, el nombre muestra asimismo un icono que representa el tipo de programa.
Tipo:	Tipo de archivo. Este ejemplo muestra Documento Microsoft Word.
Localización:	Nombre de carpeta que contiene al archivo.

Item	Descripción
Tamaño:	El tamaño del archivo se muestra tanto en KB (kilobytes) como en bytes.
Nombre MS-DOS:	El nombre convertido basado en los seis primeros caracteres del nombre del archivo, un punto, y la extensión del tipo de archivo. Utilizará este nombre si copia el archivo a un sistema DOS:
Creado:	La fecha en que creó (o copió por primera vez) el archivo.
Modificado:	La última vez que guardó el archivo.
Usado:	Si está viendo el archivo en este momento, la fecha de hoy.
Atributos:	No puede cambiar estos atributos desde el comando del programa Archivo, Propiedades, pero sí puede modificarlos si introduce las propiedades a través del Explorador.
Atributo Sólo Lectura	Puede leer el archivo pero no guardarlo con el mismo nombre.
Atributo Archivo	El archivo ha sido modificado desde la última copia de seguridad
Atributo Oculto	El archivo está normalmente oculto de las listas de archivos.
Atributo Sistema	El archivo es un archivo del sistema necesario para hacer funcionar el sistema operativo y no debe ser modificado.

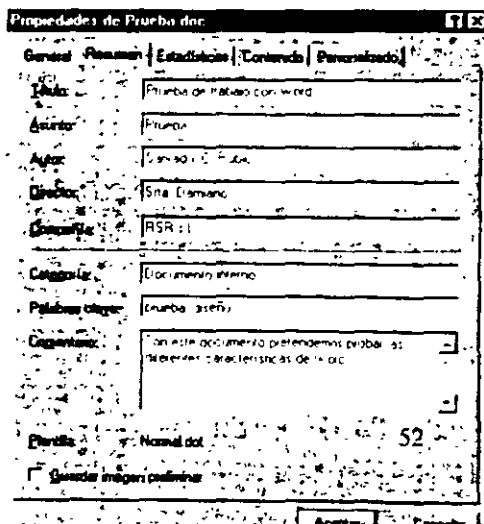
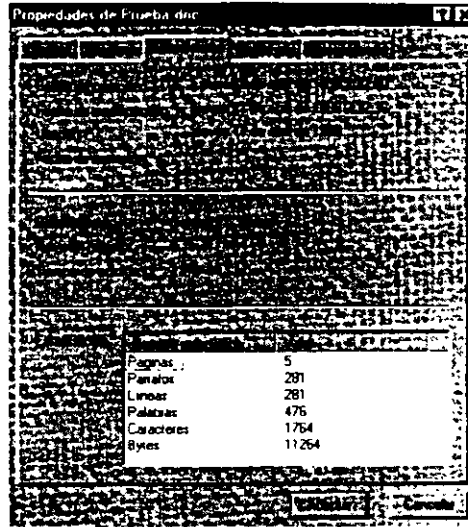


Fig. 4.28 Rellene la página Resumen de la hoja Propiedades para que le ayude a encontrar o administrar su archivo.

Fig. 4.29
La página Estadísticas muestra información acerca de fechas, horas, edición y longitud de documento.

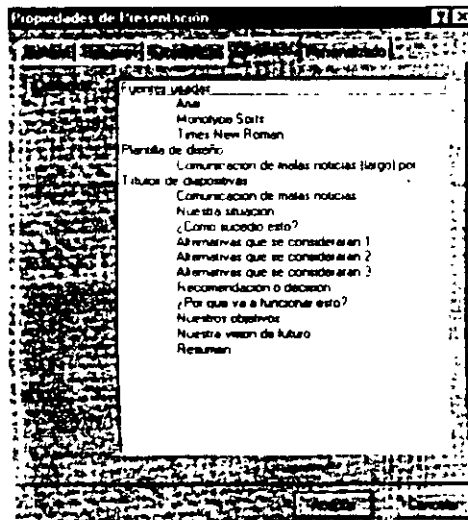


quién lo guardó por última vez; cuántas veces ha sido revisado, y cuál es el tiempo total de edición del mismo. Asimismo, Word incluye la longitud del documento en páginas, párrafos, líneas, palabras, caracteres y bytes.

Ficha Contenido

La información en la página Contenido depende del tipo de archivo que tenga. Si el archivo es un archivo Excel, aparecerán los nombres de la hoja de cálculo o de los módulos. Si el archivo es un archivo PowerPoint, aparecerán los nombres de cada diapositiva así como información sobre el formato (véase la fig. 4.30). Un archivo Word muestra sólo el Título de la página Resumen.

Fig. 4.30
La página Contenido muestra cada título de diapositiva para archivos PowerPoint.



Ficha Personalizada

Si quiere crear sus propias propiedades, dispone de mucha flexibilidad con la página Personalizada. Puede añadir sus propios nombres de campo y valores para cualquier propiedad que desee (véase la figura 4.31).

Para agregar sus propias propiedades, siga estos pasos:

1. Elija **Archivo, Propiedades**.
2. Pulse sobre la ficha Personalizada.
3. Escriba el nombre de su campo en el cuadro de texto Nombre o elija entre uno de los más de 20 nombres de campos del cuadro de lista desplegable. Incluidos en la lista desplegable hay ítems como Comprobado por, Departamento, Origen y Registrado por.
4. Elija el tipo de dato de la lista desplegable Tipo/. Los tipos incluyen Texto, Fecha, Número, y Sí o No.
5. Escriba un Valor en el área Valor o elija **Sí** o **No** (si el tipo de dato es Sí o No, verá únicamente opciones Sí, No). El valor tiene que coincidir con el tipo de dato.

Nota

Si activa la casilla de verificación **Vincular Contenido** de la página de Personalizado, el cuadro de **Valor** tendrá una lista desplegable de donde podrá elegir ítems dentro del documento a vincular, tales como un marcador en Word o una celda con nombre en Excel.

6. Elija el botón de comando **Agregar**.
7. Repita los pasos 3-6 para más campos.

SOLUCIÓN DE PROBLEMAS

No veo cinco fichas en mi hoja de Propiedades.

Asegúrese de que está usando el archivo original, y no un nombre abreviado, cuando mire las propiedades de archivo.

Necesito asociar mis documentos con números de cuenta.

Abra el archivo y elija **Propiedades de Archivo** y pulse sobre la ficha Personalizado. Pulse sobre el cuadro de texto Nombre y escriba Número de Cuenta. Elija un **Tipo** y escriba el número en el campo Valor.

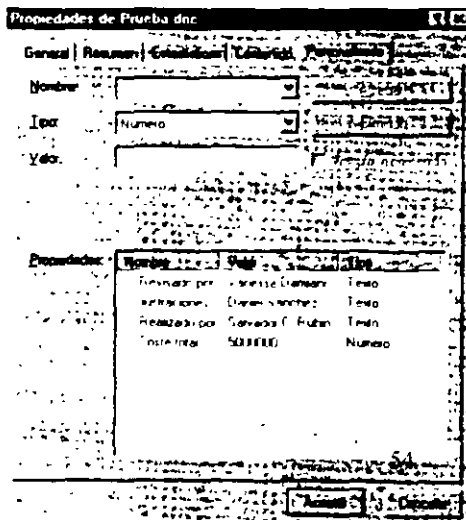


Fig. 4.31
Un conjunto personalizado de propiedades Word puede incluir la información adicional que necesita para gestionar sus documentos.

Cómo imprimir documentos

Para imprimir o ver una presentación preliminar del documento en curso, puede utilizar comandos de menú, botones de barra de herramientas, o teclas de método abreviado. Como se mencionó en un apartado anterior, puede imprimir uno o más archivos pulsando con el botón derecho del ratón en un nombre de archivo del cuadro de diálogo Abrir y eligiendo después el comando **Imprimir**.

Cómo imprimir todo el documento o parte del documento



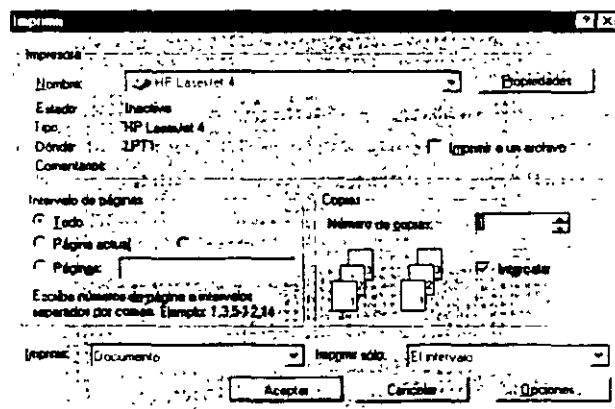
Para imprimir el documento en curso, elija el comando **Archivo, Imprimir**, presione <Control> + <P> o pulse sobre el botón [Imprimir] de la barra de herramientas Estándar.

Si usa el botón [Imprimir] o la tecla de método abreviado, se imprimirá el documento entero. Si utiliza el comando del menú, aparecerá un cuadro de diálogo similar a la figura 4.32, presentando más opciones.

Las opciones en el cuadro de diálogo Imprimir le permiten imprimir el documento entero, la página en curso, páginas específicas o texto seleccionado. También puede especificar el número de copias a imprimir. En el cuadro de texto **Páginas** en Word o en el cuadro de texto **Diapositivas** en PowerPoint, puede saltarse páginas (puede teclear 1-2, 4-7 o sencillamente 13- para imprimir desde la página 13 al final del documento).

Fig. 4.32

El cuadro de diálogo Imprimir le permite especificar lo que quiere imprimir con más detalle.



Cómo modificar las opciones de impresión

Si quiere realizar opciones de impresión adicionales, use el cuadro de diálogo Preparar Página, mostrado en la figura 4.33. Las opciones en este cuadro de diálogo le permiten fijar márgenes, imprimir encabezados y pies de página, especificar la orientación de la impresión y modificar los parámetros de la impresora.

Aplicación	Comando de Menú
Word	Archivo, Preparar Página
Excel	Archivo, Preparar Página
PowerPoint	Archivo, Preparar diapositiva

Para modificar márgenes, tamaño de papel y otras funciones, use el siguiente comando:

Para cambiar la impresora, elija una impresora de la lista desplegable en la sección Impresora del cuadro de diálogo Imprimir.

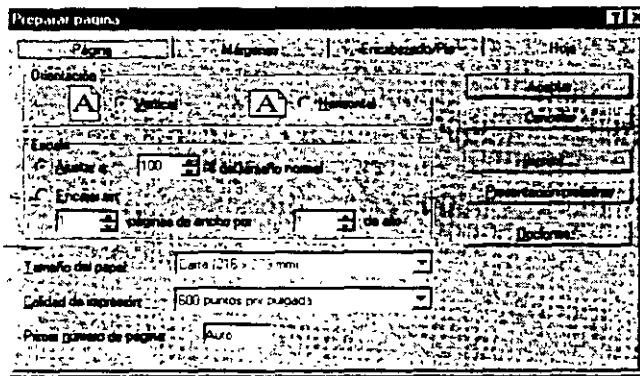


Fig. 4.33
Puede especificar opciones de impresión adicionales en el cuadro de diálogo Preparar Página de Excel

Tabla 4.5 Opciones de Presentación preliminar

Opción	Word	Excel
Modificar márgenes		X
Ver Regla (para cambiar márgenes, sangrías y tabulaciones)	X	
Aumentar/Zoom	X	X
Zoom en diferentes tamaños	X	
Páginas múltiples	X	
Ajustar a	X	X
Imprimir	X	X
Edición	X	
Siguiente (ir a la página siguiente)		X
Anterior (ir a la página anterior)	X	
Preparar Página (ir al cuadro de diálogo Preparar Página)	X	

Cómo usar la Presentación preliminar

Aunque su pantalla muestra lo que verá en la página impresa, Word y Excel tienen una opción **Presentación preliminar** que le permite ver la página entera (o más de una página), incluidos los encabezados y pies de página, números de página y márgenes. Para entrar en **Presentación preliminar** elija el comando **Archivo, Presentación preliminar**. La tabla 4.5 muestra las funciones disponibles mientras se está en modo de presentación preliminar.

La figura 4.34 muestra la Presentación preliminar de Word, y la 4.35 la Presentación preliminar de Excel.

Para modificar márgenes, pulse sobre el botón **Márgenes** o [Ver Regla], mueva el puntero del ratón a la parte superior o lateral hasta que cambie a una doble flecha negra y arrastre el margen. Para aumentar, pulse sobre el botón [Aumentar] (Word) o **Zoom** (Excel) y pulse sobre el documento en donde quiera aumentar. Para desactivar el aumento, pulse sobre el documento otra vez. En Word, para editar el documento, pulse sobre el botón [Aumentar] para desactivar el aumento y activar la edición. En Excel, para usar Ajustar a, elija el botón **Preparar Página**, pulse sobre la ficha **Página** y elija **Encajar** en y rellene los cuadros de texto Ancho por Alto de la **Página(s)**.

Fig. 4.34

La Presentación preliminar de Word puede mostrar más de una página a la vez

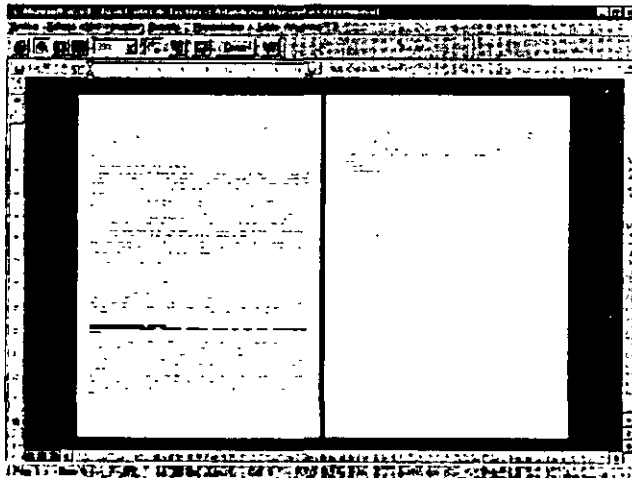
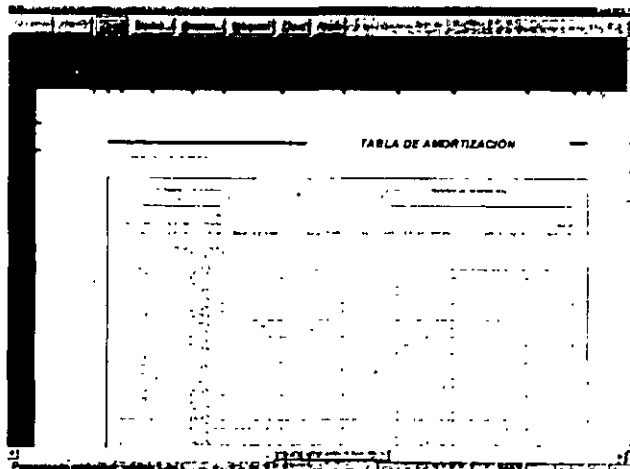


Fig. 4.35

Puede modificar los márgenes y visualizar un documento en la Presentación preliminar de Excel



Conmutar entre documentos

Cuando abre más de un archivo a la vez, tiene una ventana para cada archivo. puede conmutar entre documentos abiertos de cualquiera de las siguientes maneras:

- Elija el documento abierto en la parte inferior del menú **Ventana**.
- Pulse <Control> + <F6> para desplazarse por los documentos abiertos.
- Si partes de un documento son visibles en pantalla, pulse sobre el que quiera ver. Si un documento está minimizado, tal y como se muestra en la figura 4.36, realice una doble pulsación en su icono.

◀ VÉASE “Conmutar entre programas”, pág. 38

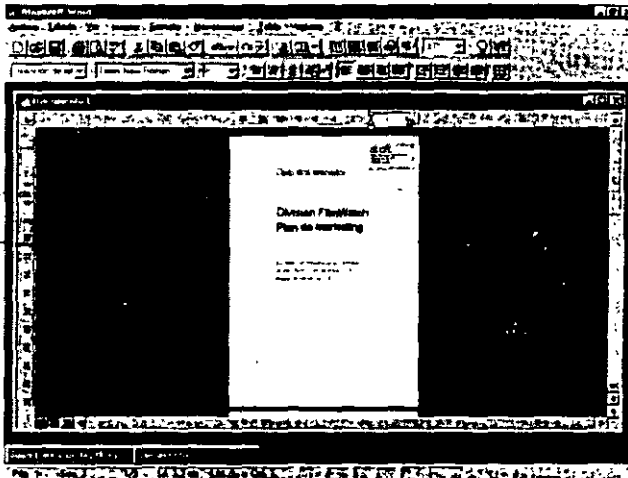


Fig. 4.36 En Word u otra aplicación cualquiera, pulse sobre el botón Minimizar para reducir el documento a un icono; realice una doble pulsación en el icono para abrir el documento.

Si quiere copiar información de un documento a otro, puede mantener visibles dos o más ventanas. En Word, Excel y PowerPoint puede utilizar la función arrastrar y colocar para mover o copiar ítems de una ventana de documento a otra. Para mover, sólo pulse y arrastre el ítem a otra ventana; para copiar, mantenga presionada la tecla <Control> mientras arrastra.

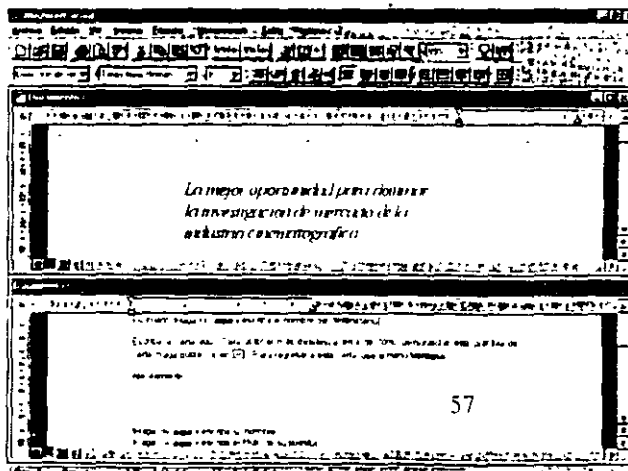
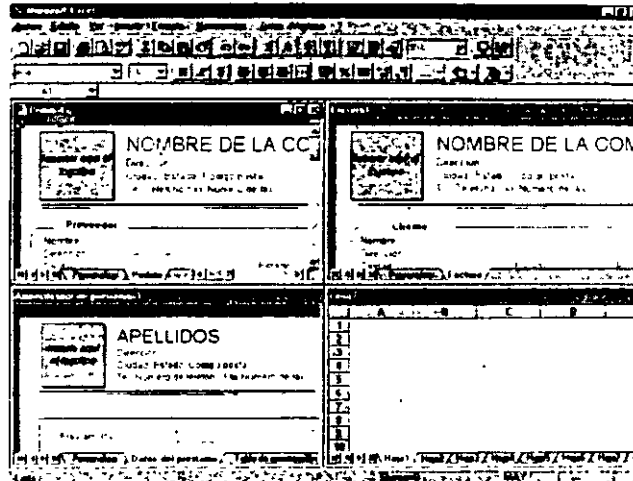


Fig. 4.37 En Word, elija **Ventana, Organizar Todo** para organizar las ventanas de los documentos en torma horizontal.

Fig. 4.38

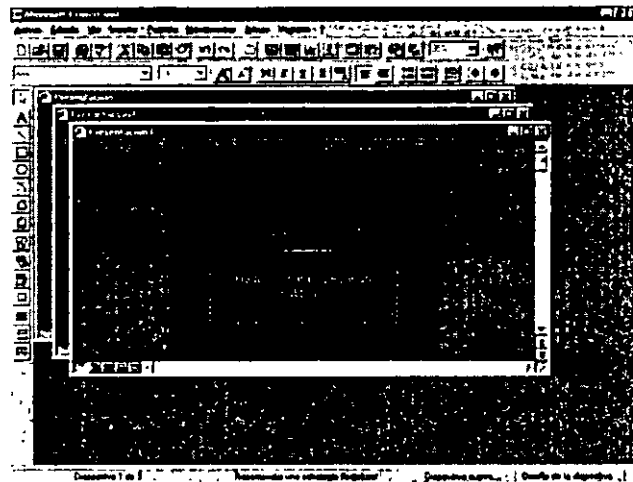
En Excel, elija el comando **Ventana, Organizar, En Mosaico** para presentar las ventanas en cuadrados de igual tamaño.



- En Word, elija el comando **Ventana, Organizar todo**, para visualizar los documentos tal y como se muestra en la figura 4.37.

Fig. 4.39

En PowerPoint, elija el comando **Ventana, Cascada** para apilar las ventanas de documento.



- En Excel, elija el comando **Ventana, Organizar**. Aparecerá un cuadro de diálogo, preguntándole cómo quiere organizar las ventanas. La opción en **Mosaico** presenta las ventanas en pequeños rectángulos (véase la fig. 4.38). **Horizontal** presenta las ventanas en filas (como Word). **Vertical** las muestra en columnas. **Cascada** apila las ventanas, dejando ver las barras de título de cada una de las ventanas.
- En PowerPoint, elija el comando **Ventana, Organizar Todo** para presentar los documentos en forma de **Mosaico**. También puede elegir el comando **Ventana, Cascada** para apilar las ventanas, mostrando las barras de título de cada una de las ventanas.

Cuando varios documentos están visibles, una barra de título oscura indica qué ventana está activa. Para activar otra ventana, pulse sobre ella. Para mostrar una de las ventanas a pantalla completa, pulse sobre el botón Maximizar o realice una doble pulsación en la barra de título.

Cómo iniciar Visual Basic para Aplicaciones

por Jeff Bankston

Las aplicaciones de Microsoft Office le ofrecen una serie de entornos variados y divergentes para llevar a cabo tareas comerciales. El cuaderno que unifica todos estos entornos es OLE 2.0 (vinculación e incrustación de objetos) y el programa que controla OLE 2.0 es Visual Basic para Aplicaciones (VBA). VBA es un entorno de programación que será finalmente el lenguaje de macro para todas las aplicaciones de Microsoft Office. OLE es una extensión del entorno Windows que permite vincular una aplicación al archivo de datos de otra aplicación directamente en su propio archivo de datos. La combinación de archivos de datos es una poderosa herramienta que podrá utilizar en VBA.

En este capítulo aprenderá a:

- Comprender Visual Basic para Aplicaciones
- Comprender la diferencia entre Visual Basic para Aplicaciones y otros lenguajes Basic
- Utilizar Visual Basic para Aplicaciones para automatizar tareas
- Crear pequeños programas en el lenguaje Visual Basic para Aplicaciones

Cómo comprender Visual Basic para Aplicaciones

Visual Basic para Aplicaciones es un entorno de programación basado en el lenguaje de programación Visual Basic para Windows. VBA ocupa el lugar de los rudimentarios lenguajes de programación de macro que se encuentran generalmente en las aplicaciones.

Además de ser una poderosa herramienta para desarrollar aplicaciones, VBA busca conciliar la diferencia entre la facilidad de los lenguajes de macro y la necesidad de lenguajes de programación orientados a objetos. Los nuevos usuarios pueden grabar sus acciones y crear macros sin necesidad de aprender las complejidades del lenguaje. De hecho, grabar acciones y observar luego el código grabado es la forma más fácil de empezar a aprender VBA por su cuenta.

En la actualidad, en Microsoft Office para Windows 95, VBA se encuentra por completo desarrollada únicamente en Excel. Word contiene su propio lenguaje de programación, llamado Word Basic, que es un poco distinto de VBA. PowerPoint no contiene un lenguaje de programación.

Microsoft planea integrar a largo plazo VBA en todas sus aplicaciones principales, incluyendo MS Access para Windows 95. Microsoft ha marcado ya el comienzo con la difusión actual de sus productos. Dado que VBA se implementa en otras aplicaciones, podrá usar los procedimientos desarrollados aquí en las otras aplicaciones con muy pocas o casi ninguna modificación.

Comprender los objetos

La diferencia entre Visual Basic para Aplicaciones y otros lenguajes de programación Basic radica en que VBA está orientada para objetos. Ya ha visto objetos en acción dentro de Microsoft Office. En el Capítulo 30, "Cómo trabajar con Asistentes, múltiples documentos, cortar, copiar y pegar", ha incrustado gráficos (objetos gráfico) y rangos de Excel (objetos rango) en los documentos de Word. Estos objetos saben cómo manipularse y mostrarse a sí mismos, incluso cuando están incrustados en un documento de otra aplicación. Visual Basic para Aplicaciones aprovecha estas capacidades para controlar estos objetos.

Visual Basic para Aplicaciones utiliza un modelo de programación orientado al objeto. Si comprende los programas orientados a objeto (OOP), le resultará sencillo entender el uso que VBA hace de ellos. Si no sabe cómo funcionan estos programas, no se preocupe, ya que el concepto no es tan complicado como parece.

Los *objetos* VBA representan una forma conveniente de almacenar y ocultar datos y codificar un programa. En lugar de escribir un programa para manipular algunos valores de datos, encapsula los datos y el código que los manipula dentro de un objeto. A partir de ello, sólo deberá acceder al objeto para utilizar o ver los datos.

Puede ver esta capacidad en acción cuando incrusta un objeto de una aplicación en otra. El objeto incrustado se autogobierna y el objeto en el que se incrusta tan sólo debe ofrecer al objeto incrustado un lugar para desplegarse. Cuando adjunta un botón de Visual Basic a una hoja de cálculo, por ejemplo, la hoja de cálculo no necesita saber cómo hacer funcionar el botón cuando pulsa -el botón se ocupa de ello.

En código, hace casi lo mismo. No intente manipular los datos de un objeto directamente, envíe mensajes al objeto y deje que éste haga el trabajo.

Los objetos VBA incluyen botones, ítems de menú, rangos de celdas de hoja de cálculo e incluso una hoja de cálculo, es decir casi todo lo que puede ver en pantalla mientras una aplicación se está ejecutando en un objeto.

► VÉASE "Cómo vincular una hoja de cálculo de Excel a un documento de Word" pág. 90.

► VÉASE "Cómo vincular datos entre hojas de cálculo de Excel", pág. 90.

Comprender objetos en Visual Basic para Aplicaciones

dores (véase la figura 35.1). El contenedor más grande es el objeto Aplicación, que es el programa que está ejecutando actualmente, como por ejemplo, Excel. En Excel, el objeto Aplicación contiene objetos Menús, objetos Control, objetos Libro de Trabajo, etc. Dentro de los objetos Libro de trabajo se encuentran los objetos Hojas (hojas de cálculo, hojas de macro, hojas de módulos, hojas de diálogo, etc.) y dentro de los objetos Hoja se encuentran los objetos Rango (rangos de celda). Otras aplicaciones contienen objetos similares para cubrir sus necesidades específicas. Véase la documentación y ayuda en pantalla para las diferentes aplicaciones de Microsoft Office para acceder a una lista de los objetos que contiene cada aplicación.

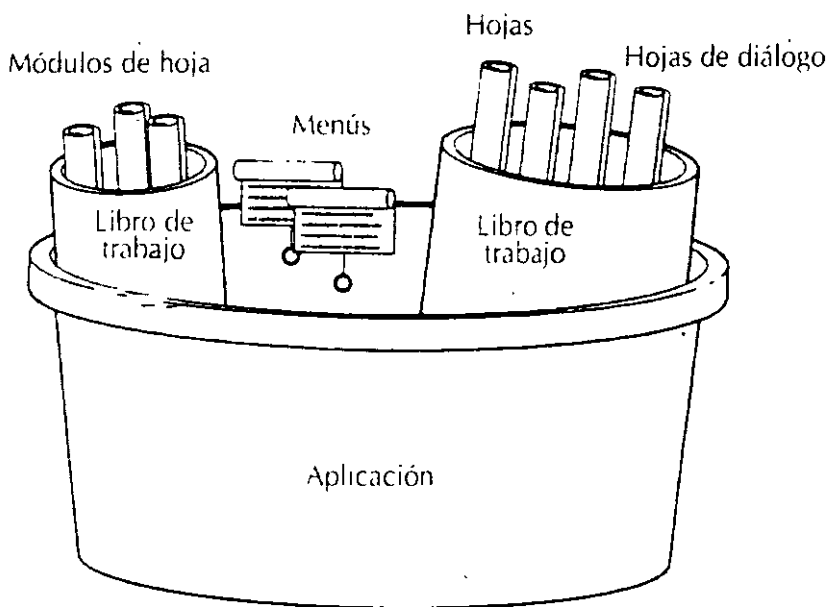


Fig. 35.1
El modelo objeto Visual Basic para Aplicaciones, contenedores dentro de contenedores.

Nota

Las celdas individuales no son objetos en Excel, pero se accede a ellas como objetos Rango que a su vez contienen una celda o un rango de celdas.

Cómo acceder a objetos

Para tener acceso a un objeto específico de Visual Basic para Aplicaciones, deberá comenzar con el objeto contenedor más externo, seguido por un punto, seguido por el siguiente objeto contenedor del interior, seguido por un punto, etc., hasta alcanzar el objeto deseado. Para acceder a la celda B5 en una hoja de cálculo de Excel cuyo nombre es Hoja3 en un libro de trabajo denominado Libro2, por ejemplo, podría utilizar la siguiente referencia:

```
Application.Workbooks("Libro2").Worksheets("Hoja3").Range("B5")
```

Dado que Visual Basic para Aplicaciones trabaja con objetos, cada aplicación que registra su objeto con el sistema operativo de Windows pone dicho objeto a disposición de Visual Basic. Por eso, incluso si está ejecutando Visual Basic para Aplicaciones en otra aplicación, podrá acceder a un objeto en Excel casi de la misma forma en que lo haría si estuviera trabajando en Excel. La única diferencia radica en que deberá incluir el nombre que utilizan las otras aplicaciones cuando se registran para especificar el objeto a usar. Por ejemplo, puede utilizar la siguiente referencia en Project para acceder a una celda en Excel:

```
Excel.Application. Workbooks("Libro2").Worksheets("Hoja3").Range("B5")
```

Esta referencia puede resultar demasiado larga, por lo que Visual Basic para Aplicaciones realiza una suposición que le permitirá omitir algunos contenedores. Para cada contenedor no incluido en la parte izquierda de la referencia a un objeto (como por ejemplo la referencia al libro de trabajo o a la hoja de cálculo), Visual Basic para Aplicaciones presupone que el objeto actualmente activo de este tipo es el que está siendo referenciado.

Por eso, podrá casi siempre omitir la aplicación, así como también el libro de trabajo. Sin embargo, deberá tener cuidado y asegurarse de cuáles son los objetos activos antes de omitirlos en la especificación. Omitir los contenedores agiliza sus procedimientos. Si omite todo pero conserva el objeto Rango, su código siempre se aplicará a la hoja actualmente activa, por lo que no deberá cambiar el nombre de la hoja para aplicar su código a una hoja diferente. Además de las hojas y libros de trabajo nombrados, podrá usar los objetos **ActiveWorkbook**, **ActiveWorksheet**, **ActiveCell** para referenciar los objetos actualmente activos sin necesidad de conocer sus nombres.

Nota

Recuerde que si utiliza nombres de libros de trabajo y hojas específicos en sus procedimientos, éstos funcionarán sólo en esos libros de trabajo nombrados y procedimientos. Al omitir parte de una especificación de objeto, hace que el código pueda aplicarse a todos los objetos de la misma clase que ha omitido.

Comprender clases y colecciones

Una *clase* de objetos es una referencia a un tipo o clasificación general de objetos. En Visual Basic para Aplicaciones, por ejemplo, cada celda o rango de celdas en una hoja de cálculo es un objeto Range, que es un ejemplo de la clase Range.

Si combina todos los objetos de una clase específica en un grupo, ese grupo se denomina *colección*. Por eso, todos los libros de trabajo en el objeto Application están en la colección Workbooks, y todas las hojas de cálculo dentro de un libro de trabajo, se encuentran en la colección Worksheets. De esta forma, todas las hojas de cálculo están en la colección Sheets, lo que incluye todos los tipos de hojas (hojas de cálculo, de gráfico, de módulo y de diálogo) dentro de un libro de trabajo.

Cómo acceder a colecciones

Las *colecciones* le permitirán acceder a la mayoría de los objetos. Para acceder a miembros específicos de una colección, coloque después del nombre de la colección ya sea una cadena que contenga el nombre del objeto o un número entero entre paréntesis. Por lo tanto, **Worksheets("Hoja 1")**, se refiere a la hoja denominada **Hoja 1**, mientras que **Worksheets(2)** se refiere a la segunda hoja de cálculo de la colección compuesta por todas las hojas de cálculo del libro de trabajo activo. Si desea acceder a la celda B5 en la tercera hoja de cálculo dentro de un libro de trabajo denominado **Libro2**, podrá utilizar la siguiente referencia:

Workbooks("Libro2").Worksheets(3).Range("B5")

Si omite el número y el paréntesis cuando referencia una colección, la referencia se establece para todos los miembros de la colección.

Atención

Tenga cuidado cuando utiliza números para seleccionar objetos en una colección. Si agrega o elimina miembros de una colección, la numeración de todos los otros miembros de la colección puede cambiar y su número podría seleccionar un objeto diferente.

Comprender las propiedades

La información de un objeto es lo que se denomina *propiedad* de un objeto. La mayoría de las propiedades son legibles, aunque no todas pueden escribirse o modificarse. Véase el apartado acerca de la ayuda en pantalla de Visual Basic para Aplicaciones para obtener una descripción de cada una de las propiedades. En la descripción de cada objeto encontrará una lista de las propiedades aplicables al mismo.

Para un objeto del tipo Range (una o más celdas de hoja de cálculo), la fuente, el color, el tamaño de fuente, el contenido, etc. son propiedades de lectura/escritura, pero su ubicación es sólo de lectura (las celdas no se desplazan). Las propiedades pueden referirse a la información directa contenida en un objeto, como por ejemplo el valor de una celda o a los valores de datos que controlan el aspecto y comportamiento de un objeto, como por ejemplo el color.

Los valores de propiedad pueden ser cadenas de texto, de números, cadenas lógicas (**True o False**) o listas enumeradas. Una *lista enumerada* es una lista de opciones numerada, en la que el número se utiliza para seleccionar una opción específica. Por ejemplo, la propiedad **Color** de la mayoría de los objetos es una lista enumerada en la que 0 es igual a ningún color, 1 es negro, 2 es blanco, 3 es rojo, 4 es verde, 5 es azul, etc.

Para las listas enumeradas, VBA y las otras aplicaciones en Microsoft Office contienen listas de constantes predefinidas para utilizar en lugar de números. Usar las constantes resulta mucho más informativo que utilizar los números. Las constantes aplicables a una propiedad se encuentran listadas en la descripción de las propiedades en la ayuda en pantalla. Podrá

Consejo

Cada objeto tiene propiedades de tamaño, forma y color.

acceder a una lista de constantes buscando *constantes* o *variables* en la ayuda en pantalla y seleccionando los temas "Resumen de palabras clave de variables y constantes" y "Constantes de Visual Basic".

Cómo acceder a las propiedades

La manera más sencilla para saber qué propiedades establecer y a qué valores, es iniciar la Grabadora de macros, realizar los cambios, detener la grabadora y copiar en su programa los cambios de propiedad grabados. Tanto Excel como Word cuentan actualmente con grabadoras de macro.

La siguiente es la sintaxis que le permite acceder a las propiedades de un objeto:

objeto.propiedad

En este ejemplo, "**objeto**" es el objeto cuyas propiedades desea modificar o ver y "**propiedad**" es el nombre de la propiedad. Si la construcción anterior está en el lado derecho de la instrucción, está leyendo el valor de la propiedad desde el objeto. Si la construcción se encuentra en el lado izquierdo de la instrucción, está estableciendo el valor de la propiedad. Para establecer el valor de la propiedad **Formula** (el contenido de la celda) de la celda B5 en =ABS(B4) cuando B5 está en la hoja de cálculo **Hoja1** (dentro del libro de trabajo **Libro2**), podrá utilizar la siguiente instrucción:

```
WorkBooks("Libro2").Worksheets("Hoja1").Range("B5").Formula = "=ABS(B4)"
```

Para leer la misma propiedad desde la celda y almacenarla desde la misma celda en la variable **miFormula**, podrá utilizar la instrucción:

```
miFormula = WorkBooks("Libro2").Worksheets("Hoja1").Range("B5").Formula
```

Aquí se aplican las reglas que conciernen a la omisión de objetos de los contenedores (descritas en "Cómo acceder a los objetos" anteriormente en este capítulo). Dado que debe incluir un objeto con la propiedad, no podrá omitir el objeto **Range** para obtener la fórmula en cualquiera que sea la celda activa. Para este caso u otros similares que traten objetos, algunas propiedades especiales devuelven el objeto actualmente activo o seleccionado. La tabla 35.1 lista estas propiedades especiales.

Tabla 35.1: Propiedades especiales que devuelven los objetos activos

Propiedad	Descripción
ActiveCell	La celda activa en la ventana activa
ActiveChart	El gráfico activo en un libro de trabajo
ActiveDialog	La hoja de diálogo activa en un libro de trabajo
ActiveSheet	La hoja de cálculo, gráfico, módulo u hoja de diálogo activa en un libro de trabajo
ActiveWorkbook	El libro de trabajo activo en una aplicación
Selection	El objeto actualmente seleccionado en la hoja actualmente seleccionada

Atención

*Asegúrese de que un objeto del tipo esperado sea el objeto activo antes de intentar usar las propiedades activas, como por ejemplo **ActiveSheet**, en un procedimiento. Si un objeto del tipo especificado no es un objeto activo, estas propiedades no devolverán nada y es posible que el procedimiento que las utilice fracase.*

Para obtener la fórmula contenida en la celda activa de la hoja de cálculo **Hoja3**, por ejemplo, podrá utilizar la siguiente instrucción:

```
miFormula =  
Workbooks("Libro2").Worksheets("Hoja1").Range("B5").ActiveCell.Formula
```

Si **Libro2** y **Hoja2** son el libro de trabajo y la hoja de cálculo activos actualmente, podrá usar la instrucción:

```
miFormula = ActiveCell.Formula
```

Si **Libro2** es el libro de trabajo activo pero **Hoja3** no es necesariamente la hoja de cálculo activa, podrá utilizar la siguiente instrucción:

```
miFormula = Worksheets("Hoja3").ActiveCell.Formula
```

Si desea acceder a la celda B5 en cualquier hoja de cálculo activa en **Libro2**, podrá utilizar la siguiente instrucción:

```
miFormula = Workbooks("Libro2").ActiveSheet.Range("B5").Formula
```

Todo lo que se encuentra a la izquierda del último punto debe ser considerado como parte de un objeto o una colección de objetos.

Comprender los métodos

Los *métodos* de Visual Basic para Aplicaciones son bloques de código almacenados en un objeto que saben cómo manipular los datos del objeto. Para un objeto **Range**, por ejemplo, el método **Calculate** hace que se vuelvan a calcular las fórmulas de las celdas seleccionadas y el método **Clear**, borra el contenido de la celda. Los métodos actúan sobre los objetos y los datos que éstos contienen, en oposición a las propiedades que establecen valores. Para aprender más acerca de los detalles específicos de los diferentes métodos y para descubrir qué método se aplica a cada objeto, véase el apartado de ayuda en pantalla de Visual Basic para Aplicaciones.

Cómo acceder a los métodos

Consejo

Los métodos llevan a cabo la tarea de un objeto.

Accede o ejecuta los métodos de un objeto casi de la misma forma en que accede a las propiedades de un objeto. La diferencia principal radica en que siempre accede a una propiedad como parte de una fórmula, pero un método deberá ser parte de una fórmula sólo si devuelve un valor. El método **Rows**, por ejemplo, devuelve una colección que contiene todas las filas del rango. Para usar este método a fin de establecer la propiedad **RowHeight** de todas las filas del rango actualmente seleccionado en 20, utilice una fórmula como la siguiente:

Selection.Rows.RowHeight = 20

Para obtener el número de filas de la selección actual, podrá usar el método **Rows** para devolver una colección y la propiedad **Count** para devolver el número de ítems en la colección, tal y como se muestra a continuación:

numFilas = Selection.Row.Count

Algunos métodos requieren argumentos para que funcionen. Por ejemplo, el método **Insert**, cuando se aplica a un objeto de tipo **Range**, necesita de un argumento que le indique cómo mover las celdas que ya están en la selección. El método **Rows** necesita un número índice para seleccionar una sola fila en la colección de filas. Dado que el método forma parte de una fórmula, el argumento deberá ir entre paréntesis. Para obtener el alto de fila de la segunda fila en la colección de filas, podrá utilizar la siguiente instrucción:

AltoDeFila = Selection.Rows(2).RowHeight

Si el método está sólo siendo ejecutado y no forma parte de una fórmula, coloque los argumentos a la derecha de la referencia al método. Para utilizar el método **Insert** a fin de insertar celdas vacías en la selección actual y desplazar la selección hacia abajo para crear espacio, podrá usar esta sentencia:

Selection.Insert x1Down

El argumento es en realidad un número entero, pero la constante incorporada se utiliza aquí para que el código sea más legible. Puede obtener las constantes incorporadas que se aplican a un método en la descripción del método de la ayuda en pantalla o puede buscar *constantes* o *variables* en

la ayuda en pantalla y seleccionar los temas "Resumen de palabras clave de variables y constantes" y "Constantes de Visual Basic".

Cómo crear procedimientos con la grabadora de macros

La forma más conveniente de aprender a usar Visual Basic para Aplicaciones es crear procedimientos con VBA, usando la Grabadora de macros. Cuando activa la Grabadora de macros y crea una hoja de cálculo, la grabadora escribe el código de Visual Basic que realiza las mismas acciones que hace a mano. Examinado el código, podrá aprender a utilizar VBA para acceder y modificar los objetos de una aplicación.

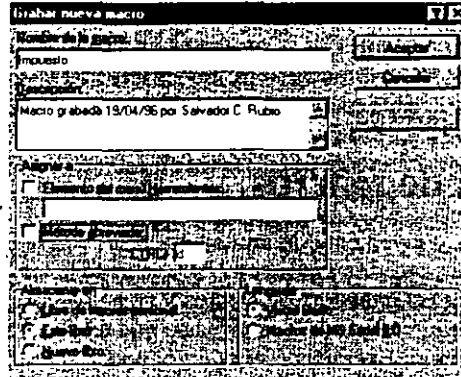
En los siguientes apartados, creará una simple hoja de cálculo que calcula el impuesto sobre el costo de un ítem y luego calcula el costo total. La hoja de cálculo tiene una celda de entrada y dos celdas de salida calculadas. La celda de entrada acepta un costo y las celdas de salida muestran el impuesto y el costo total.

Cómo iniciar la Grabadora

Para preparar la hoja de cálculo y abrir el cuadro de diálogo Grabar nueva macro, siga estos pasos:

1. Si aún no lo ha hecho, Inicie Excel
2. Abra un nuevo libro de trabajo eligiendo **Archivo, Nuevo**, seleccione Libro de trabajo y pulse en [Aceptar.]
3. Elija **Herramientas, Grabar macro, Grabar nueva macro** y luego elija el botón **Opciones**. Aparecerá el cuadro de diálogo Grabar nueva macro (véase la figura 35.2).
4. En el campo **Nombre de la macro**, escriba **ImpuestoCifra**.
5. En el campo **Descripción**, escriba esta línea:
Crear una hoja de cálculo para calcular el impuesto sobre un ítem
6. Deje los otros campos con los valores por defecto que se muestran en la figura 35.2 y elija [Aceptar.]

Fig. 35.2
El cuadro de diálogo Grabar nueva macro le permite establecer el nombre y otras opciones para un nuevo procedimiento



El botón [Detener macro] aparece en una barra de herramientas flotante y la Grabadora de macros graba lo que hace, registrando todas las pulsaciones de tecla y pulsaciones del ratón hasta que pulse en el botón [Detener macro.]

Cómo grabar un procedimiento

Para crear un procedimiento basta con crear la hoja de cálculo como suele hacerlo. Para crear una hoja de cálculo, siga estos pasos:

1. Seleccione la celda B4 y escriba **Costo**.
2. Seleccione la celda B5 y escriba **Impuesto**.
3. Seleccione la celda B6 y escriba **Total**.
4. Seleccione la celda C4 y escriba **12.43**.
5. Seleccione la celda C5 y escriba **=C4*0.0825**
6. Seleccione la celda C6 y escriba **=C4 + C5**
7. Seleccione la celda C4, elija **Formato, Celdas** y seleccione la ficha **Número**. Seleccione el tipo de formato **Moneda**, seleccione el formato **\$1243,10** y elija [Aceptar]. Repita este paso con la celda C6. Repita este paso con la celda C5, pero utilice el formato **Número**.
8. Seleccione la celda C5, elija **Formato, Celdas** y seleccione la ficha **Bordes**.
9. En el cuadro de diálogo **Bordes**, en su parte izquierda pulse en **Inferior** para el emplazamiento del borde y en la zona derecha, pulse en el cuadro inferior izquierdo para el estilo. Pulse en [Aceptar] para continuar.

▶ VÉASE "Cómo introducir datos", pág. 305.

▶ VÉASE "Cómo seleccionar celdas y rangos", pág. 309.

▶ VÉASE "Cómo formatear números", pág. 141.

La hoja de cálculo deberá parecerse ahora a la que muestra la figura 35.3

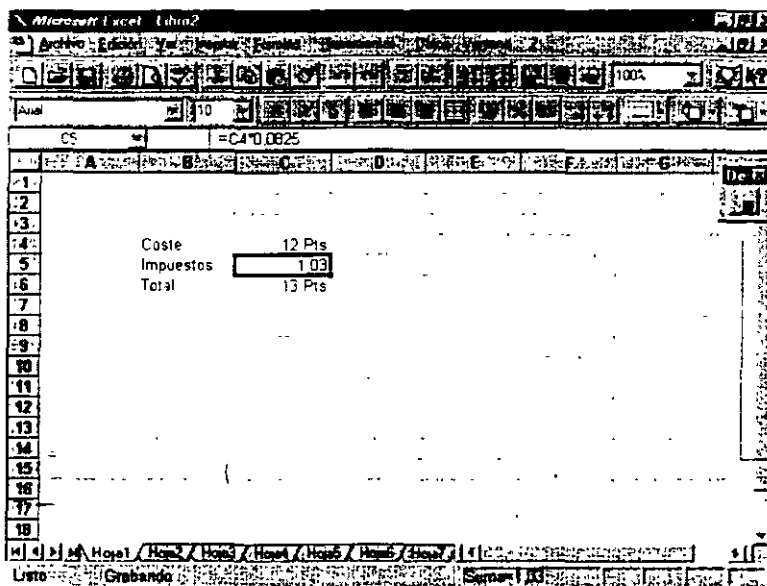


Fig. 35.3

La hoja de cálculo terminada antes de desactivar la Grabadora de macros. Observe en el lado derecho el botón [Detener grabación].

Detener grabación

La barra de estado indica que la macro se está grabando

Cómo detener la Grabadora

Detener la grabadora es una tarea sencilla: simplemente pulse en el botón Detener macro.

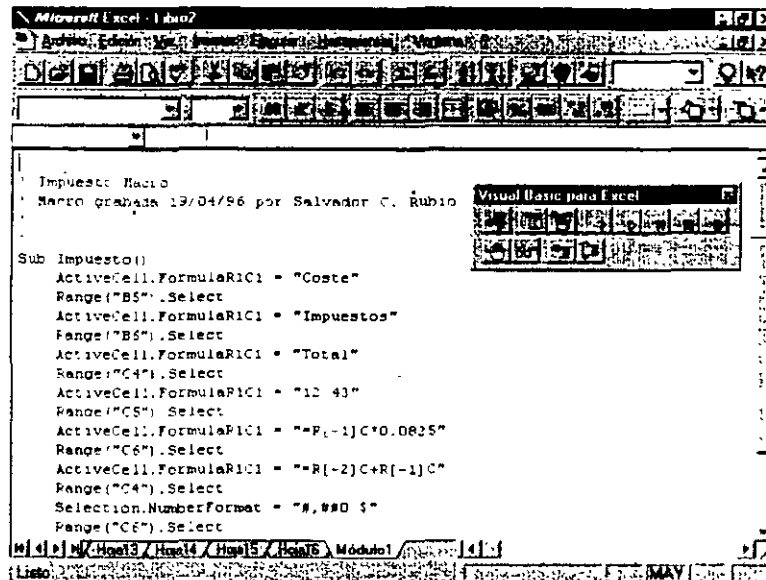


Cómo examinar un procedimiento

Para examinar el procedimiento creado recientemente, busque la ficha Modulo 1 situada en la parte inferior de la pantalla y pulse en ella. Su procedimiento aparecerá en pantalla y deberá parecerse al de la figura 35.4.

Fig. 35.4

La grabadora de macros de Excel coloca los comandos grabados en un módulo, tal y como se muestra aquí para el procedimiento



Impuesto
Cifra

Consejo

El código para cada módulo y procedimiento se almacena dentro de la hoja de cálculo y se guarda con él. Si tiene problemas para comprender algún código, busque en Edición Especial de cómo Usar Visual Basic de la editorial Prentice Hall.

La lista de los procedimientos se muestra en la lista 35.1

Lista 35.1 El procedimiento ImpuestoCifra

- ImpuestoCifra Macro
- Crear una hoja de cálculo para calcular el impuesto sobre un ítem.

```
Sub ImpuestoCifra()
    Range("B4").Select
    ActiveCell.FormulaR1C1 = "Costo"
    Range("B5").Select
    ActiveCell.FormulaR1C1 = "Impuesto"
    Range("B6").Select
    ActiveCell.FormulaR1C1 = "Total"
    Range("C4").Select
    ActiveCell.FormulaR1C1 = "12.43"
    Range("C5").Select
    ActiveCell.FormulaR1C1 = "=R[-1]C*0.0825"
    Range("C6").Select
    ActiveCell.FormulaR1C1 = "=R[-2]C + R[-1]C"
    Range("C4").Select
    Selection.NumberFormat = "$#,##0.00"
End Sub
```



```
Range("C6").Select
Selection.NumberFormat = "$#,##0.00"
Range("C5").Select
Selection.NumberFormat = "0.00"
Selection.Borders(xlLeft).LineStyle = xlNone
Selection.Borders(xlRight).LineStyle = xlNone
Selection.Borders(xlTop).LineStyle = xlNone
With Selection.Borders(xlBottom)
    .Weight = xlThick
    .ColorIndex = xlAutomatic
End With
Selection.BorderAround LineStyle:=xlNone
End Sub
```

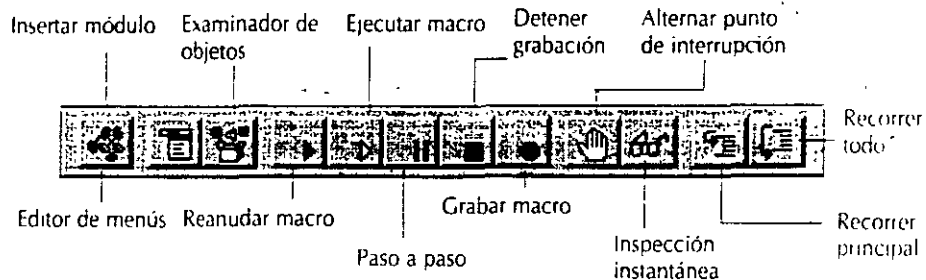
Si observa esta lista y la compara con los pasos anteriores, podrá ver que cada paso corresponde a una o más líneas de código insertadas en el procedimiento. Existen también muchas líneas adicionales en el procedimiento, son las que establecen los parámetros que no ha establecido de forma explícita al crear su hoja de cálculo. Estas líneas adicionales aparecen cuando elige [Aceptar] en un cuadro de diálogo que establece varios parámetros. Aunque modifique sólo un parámetro en el cuadro de diálogo, al cerrar el cuadro se ajustan todos los parámetros que se muestran en el cuadro y se insertan las líneas correspondientes en el procedimiento que está grabando.

Nota

En el cuadro de diálogo **Bordes**, por ejemplo, ajusta sólo la opción **Inferior** (paso 9), pero se insertan 8 líneas en el procedimiento, que establecen los valores de todas las otras opciones. En muchos casos, podrá eliminar las líneas adicionales del procedimiento sin modificar sus resultados.

El procedimiento aparece en color, con los comentarios en verde, las palabras clave en azul y todo lo demás en negro. La barra de herramientas de Visual Basic aparece flotando en la parte derecha de la ventana del procedimiento. La barra de herramientas puede aparecer flotando o anclada en cualquiera de los lados. La figura 35.5 muestra la función de cada botón de la barra de herramientas de Visual Basic. Los primeros tres botones, de izquierda a derecha, son Insertar módulo, Examinador de objetos y Editor de menús. Los siguientes cinco botones sirven para ejecutar y manipular módulos y los cuatro últimos para depurar módulos.

Fig. 35.5
La barra de herramientas de Visual Basic contiene controles para ejecutar y depurar un programa



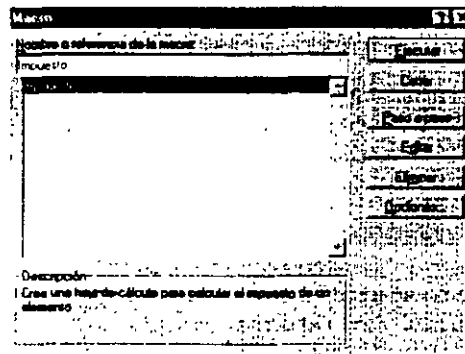
Cómo ejecutar un procedimiento



Para ejecutar este procedimiento, siga estos pasos:

1. Seleccione una hoja de cálculo no utilizada. Asegúrese de que está sin utilizar o no contiene nada útil en el rango B4:C6 porque el procedimiento sobrescribe ese área.
2. Elija **Herramientas, Macro**. Aparecerá el cuadro de diálogo Macro mostrado en la figura 35.6 y podrá observar en él todos los procedimientos disponibles en esta hoja y en la hoja general (actualmente ninguno)
3. En el cuadro de diálogo, seleccione el procedimiento **ImpuestoCifra** y elija **Ejecutar**. Aparecerá la hoja de cálculo y el procedimiento se ejecutará, estableciendo el contenido y formateando las celdas de la hoja de cálculo. La hoja de cálculo terminada es idéntica a la que ha creado manualmente.

Fig. 35.6
El cuadro de diálogo Macro le permite seleccionar y ejecutar procedimientos. Este cuadro de diálogo le ofrece además una forma sencilla de localizar, editar o eliminar procedimientos.



Comprender el procedimiento

Ahora observe de nuevo y en detalle la lista de procedimientos seleccionando la ficha Módulo 1. Las primeras líneas del procedimiento son comentarios. Un procedimiento en ejecución ignora los comentarios y éstos pueden contener cualquier texto.

Todo texto a continuación de las comillas simples es un comentario. Estos pueden aparecer al principio de la línea o a la derecha de cualquier instrucción válida de VBA.

```
' ImpuestoCifre Macro  
' Crear una hoja de cálculo para calcular el impuesto sobre un ítem.
```

A continuación de los comentarios aparece el encabezado del procedimiento, que le da un nombre:

```
Sub ImpuestoCifre()
```

A continuación del encabezado del procedimiento, se encuentran 12 instrucciones que seleccionan de forma alterna cada celda en el rango B4:C6 e insertan texto, un valor numérico o una fórmula. Las últimas cuatro líneas insertan fórmulas, pero aparecen escritas en estilo R1C1 y no en el estilo A1 de localización de celdas. Este es el método por defecto para guardar fórmulas insertadas, independientemente del método que haya utilizado para crear la hoja de cálculo mientras la grabadora estaba en ejecución. Cuando se reproduce este procedimiento, las fórmulas adoptan automáticamente el estilo por defecto de la hoja de cálculo activa. Las 12 instrucciones son como estas:

```
Range("B4").Select  
ActiveCell.FormulaR1C1 = "Costo"  
Range("B5").Select  
ActiveCell.FormulaR1C1 = "Impuesto"  
Range("B6").Select  
ActiveCell.FormulaR1C1 = "Total"  
Range("C4").Select  
ActiveCell.FormulaR1C1 = "12.43"  
Range("C5").Select  
ActiveCell.FormulaR1C1 = "=R[-1]C*0.0825"  
Range("C6").Select  
ActiveCell.FormulaR1C1 = "=R[-2]C + R[-1]C"
```

Las dos instrucciones siguientes seleccionan el rango C4 y aplican el formato moneda a esa celda. Lo mismo ocurre con la celda C6.

```
Range("C4").Select  
Selection.NumberFormat = "$#,##0.00"  
Range("C6").Select  
Selection.NumberFormat = "$#,##0.00"
```

Las últimas nueve líneas del procedimiento son el resultado de seleccionar la celda C5 y hacer modificaciones en el cuadro de diálogo Bordes. Observe que si bien ha seleccionado sólo el cuadro Inferior, el procedimiento establece valores para todas las opciones del cuadro de diálogo. Los valores `xlNone`, `xlBottom`, `xlThin`, etc., son constantes incorporadas en Excel. Véase la ayuda en pantalla para acceder a una lista de las constantes disponibles para utilizar con las diferentes propiedades y métodos. Las líneas son como estas:

```
Range("C5").Select
Selection.Borders(xlLeft).LineStyle = xlNone
Selection.Borders(xlRight).LineStyle = xlNone
Selection.Borders(xlTop).LineStyle = xlNone
With Selection.Borders(xlBottom)
    .Weight = xlThick
    .ColorIndex = xlAutomatic
End With
Selection.BorderAround LineStyle:=xlNone
```

Nota

La estructura **With, End With** en el procedimiento, es un método para disminuir el tamaño de un procedimiento y ordenar conjuntamente las instrucciones que se aplican a un objeto específico. La instrucción **With** funciona insertando el objeto que sigue a la palabra **With** antes del punto en cada línea debajo de la misma. Por lo tanto, los dos siguientes bloques de código son equivalentes:

```
With Selection.Borders(xlBottom)
    .Weight = xlThick
    .ColorIndex = xlAutomatic
End With

Selection.Borders(xlBottom).Weight = xlThin
Selection.Borders(xlBottom).ColorIndex = xlAutomatic
```

Sugerencia

Para aprender una nueva técnica lo más conveniente por lo general es tener un proyecto en mente. Piense en alguna tarea que recita a menudo en Excel que siempre haya deseado automatizar y tendrá el objetivo de su aplicación.

La última instrucción en el procedimiento es el pie de página del procedimiento, que marca el final del mismo:

```
End Sub
```

Cómo crear una aplicación

Ahora que ya conoce los objetos, las propiedades y los métodos, puede empezar a combinar la información para crear una aplicación. La aplicación que va a crear es un creador de recibos, como el que podría utilizar en los sitios en donde no hay una caja registradora normal. El programa introduce datos a través de un cuadro de diálogo, inserta los

datos en un formulario en una hoja de cálculo de Excel e imprime el formulario.

La estructura básica de este programa -datos de entrada, su almacenamiento, hacer algo con ello- compone la base de muchos programas, como programas de inventario o un organizador personal. Deberá ser capaz de adaptar los métodos descritos aquí para diferentes aplicaciones.

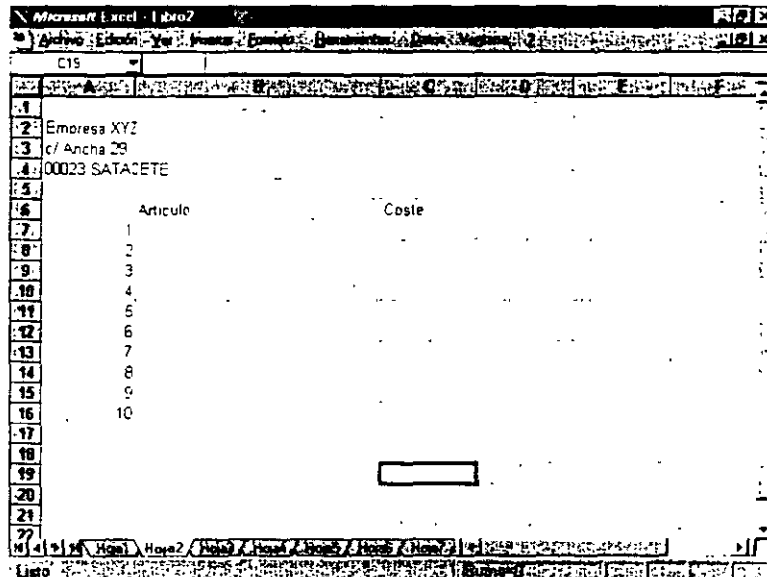
El programa utiliza una hoja de cálculo y un módulo. La hoja de cálculo contiene el formulario de recibo que almacena los datos hasta que esté listo para imprimir. El módulo contiene el código que recoge los datos, los almacena en la hoja y los imprime. En primer lugar, cree la hoja de cálculo. No se preocupe si no comprende lo que hace el código, se explica en los siguientes apartados.

Para crear una hoja de cálculo, siga estos pasos.

1. Seleccione una hoja de cálculo no utilizada. Elija **Formato, Hoja, Cambiar** nombre o realice una doble pulsación en la ficha en la parte inferior de la hoja y cambie el nombre de la hoja por **Recibo** en el cuadro de diálogo y pulse en [Aceptar].
2. Seleccione la celda A2 y escriba **Papeleras XYZ**
3. Seleccione la celda A3 y escriba **Calle Alguna parte, 1127**.
4. Seleccione la celda A4 y escriba **Aquí mismo, 28000 Madrid**.
5. Seleccione la celda B6 y escriba **Item**.
6. Seleccione la celda C6 y escriba **Costo**.
7. Seleccione la celda A7 y escriba **1**.
8. Seleccione la celda A8 y escriba **2**.
9. Seleccione las celdas A7:A8, seleccione el cuadro de selección en la esquina inferior derecha y arrástrelo hacia abajo hasta A16 para crear la serie de números de 1 a 10.
10. Seleccione C16, elija **Formato, Celdas**, seleccione la ficha Bordes, seleccione un estilo de línea grueso, pulse en el cuadro Inferior y elija [Aceptar].
11. Seleccione las celdas C7:C19, elija **Formato, Celdas**, seleccione la ficha Número, seleccione el tipo moneda, seleccione el formato **\$1,234,19** y elija [Aceptar].
12. Seleccione la barra entre los encabezados de columna B y C y arrástrela hasta que el ancho de columna tenga un valor de 27.00 caracteres.
13. Seleccione la celda B7 y escriba **TopOfList**, usando **Insertar, Nombre, Definir**.
14. Seleccione las celdas A1:D20 y asígneles el nombre **PrintRange**.
15. Elija **Archivo, Preparar página**, seleccione la ficha Encabezado/Pie y establezca en "ninguno" el encabezado y el pie de página. Seleccione a ficha Hoja, desactive la selección de todas las casillas de verificación y elija [Aceptar].

La hoja de cálculo debe parecerse a la de la figura 35.7

Figura 35.7
Diseño de la
hoja de cálculo
Recibo



El siguiente paso consiste en crear los procedimientos en un módulo y adjuntarlos a dos botones en el formulario.

Para crear los procedimientos, siga estos pasos:

1. Elija **Insertar, Macro, Módulo**, para insertar un nuevo módulo.
2. Seleccione el módulo y cámbiele el nombre por **Creador de Recibos** eligiendo **Edición, Hoja, Cambiar nombre** o realizando una doble pulsación en la ficha de nombre del módulo.
3. Escriba en el módulo los procedimientos que se muestran en la lista 35.2

Lista 35.2 El programa Creador de recibos

```
'Creador de recibos
```

```
'Un programa para crear recibos
```

```
Option Explicit 'Force the declaration of all variables.
```

```
Dim laHoja As Object 'Pointer to the Worksheet.
```

```
Dim RangoSalida As Object 'Pointer to the list of items.
```

```
Const MaxNumItems = 10 'Maximum number of items allowed.
```

```
'Get entries.
```

```
Sub GetEntries()
```

```
Dim elItem As String, elCosto As Currency
```

```
Dim lafila As Integer, NumItem As Integer
```

```
'Define the pointer to the worksheet.
```

```
Set laHoja = Application.Workbooks("Ejemp.xls").
```

```

Worksheets("Recibo")
'Define the pointer to the top of the tables items.
Set RangoSalida = laHoja.Rango("ParteSuperiorLista")
ClearRange RangoSalida 'Clear the table of items.
NumItems = 1
Ask for the name and cost for up to 10 items.
Do While NumItems <= MaxNumItems
'Get the name.
ElItem =InputBox("Nombre Item:", "Crear Recibo")
' If the user didn't enter anything, _ he must be done, so quit.
If elItem = "" Then Exit Do
    Get the cost.
    ElCosto = Val(InputBox("Costo Item:", "Crear Recibo"))
    ' Insert the items name and cost on the worksheet.
    OutputRange.Cells(NumItems, 1).Formula = elItem
    OutputRange.Cells(NumItems, 2).Formula = Str(elCosto)
    NumItems = NumItems + 1 ' Increment the number of items.
Loop
Totalizar ' Calcular e imprimir totales.
' Ejecutar procedimientos Totalizar y prueba
' attached to the worksheet.
LaHoja.OnEntry = "Totalizar" ' Volver a totalizar si el usua-
rio realiza modificaciones.

End Sub
'
' Calcular subtotal y total
'
Sub Totalizar()
    Dim laFila As Integer
    Dim SubTotal As Currency, ImpuestoItem As Currency
    Dim elTotal As Currency
    SubTotal = 0
    'Calculate the total by extracting the values from the works-
    heet.
    For laFila = 1 to MaxNumItems
        SubTotal = SubTotal + Val(OutputRange.Cells(laFila,
        2).Value)
    Next laFila
    'Insert the subtotal, tax, and total on the worksheet.
    With RangoSalida
        .Cells(MaxNumItems + 1, 1).Formula = "Subtotal"
    End With

```

```

.Cells(MaxNumItems + 1, 2).Formula = Str(Subtotal)
.Cells(MaxNumItems + 2, 1).Formula = "Impuesto"
.ImpuestoItem = elImpuesto(SubTotal) 'Calculate the tax.
.Cells(MaxNumItems + 3, 1).Formula = "Total"
.Cells(MaxNumItems + 3, 2).Formula = Str(elTotal)
End With
End Sub
'
'Clear the output range.
'
Sub ClearRange(elRango As Object)
Dim laFila As Integer
For laFila = 1 To MaxNumItems + 3
'Clear the cells. Use ClearContents to only clear the va-
lues
'and not the formatting.
elRango.Cells(laFila, 1).ClearContents
elRango.Cells(laFila, 2).ClearContents
Next laFila
End Sub
'
'Imprimir el recibo.
'
Sub ImprimirRecibo()
laHoja.OnEntry = "" 'Turn off the automatic retotaling
laHoja.Rango("ImprimirRango").PrintOut 'Print the worksheet.
End Sub
'
'Calculate the tax on an item.
'
Function elImpuesto(Costo As Currency) As Currency
Const Tasalmpuesto = 0,0825
elImpuesto = Costo * Tasalmpuesto
End Function

```



4. Conmute a la hoja de cálculo Recibo y despliegue la barra de herramientas **Dibujo** eligiendo **Ver, Barra de herramientas**.
5. Pulse en la herramienta Crear botón en la barra de herramientas Dibujo y dibuje el botón Crear recibo, tal y como se muestra en la figura 35.8. Cuando aparezca el cuadro de diálogo Asignar macro, seleccione **Ge-fEntries** para la macro y elija [Aceptar.]
6. El botón deberá permanecer seleccionado, de lo contrario, pulse en la herramienta Dibujar selección en la barra de herramientas Dibujo y seleccione el botón. Seleccione el texto sobre el botón y e introduzca **Crear Recibo**

7. Cree el botón Imprimir recibo, adjúntelo a la macro **ImprimirRecibo** y asígnele el título **Imprimir recibo**.
8. Cierre la barra de herramientas dibujo realizando una pulsación en X en la esquina superior derecha de la barra de herramientas.
9. Guarde el libro de trabajo como **EJEMPS.XLS**.

La hoja de cálculo deberá parecerse a la de la figura 35.8. Si ha hecho todo correctamente, podrá utilizar el programa para crear un recibo.

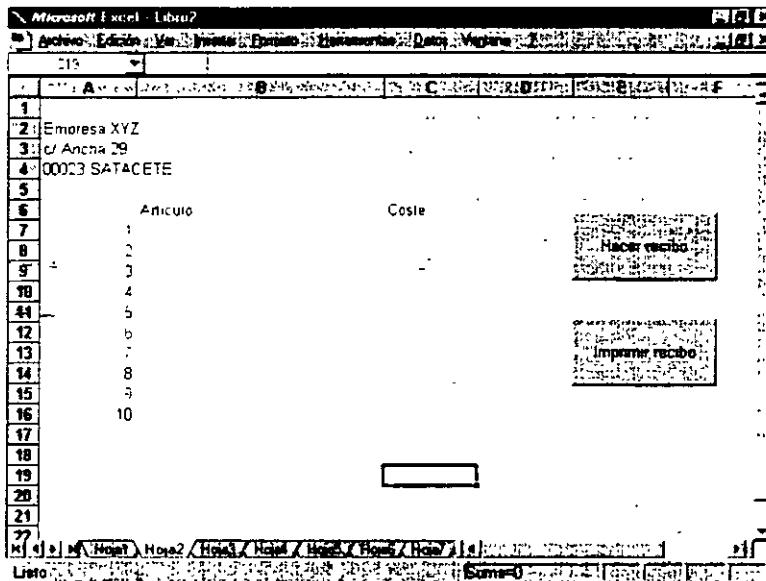


Fig. 35.8
El programa Creador de recibos terminado

1. Conmute a la hoja de cálculo Recibo y pulse en el botón [Crear recibo].
2. En el primer cuadro de diálogo que aparece, escriba el nombre de un ítem y pulse <Intro>.
3. En el segundo cuadro de diálogo, escriba el costo del ítem y pulse <Intro>.
4. Siga introduciendo nombres y costos hasta escribir todos los ítems que desea usar en el recibo.
5. Para finalizar la lista, pulse <Intro> o pulse en el botón [Cancelar] en el cuadro de diálogo Nombre del ítem. Los totales se calcularán y se mostrarán en el recibo.
6. Pulse en el botón [Imprimir] recibo para imprimir el recibo.

Cómo ramificarse y tomar decisiones

A medida que lee la lista de algunos programas, es posible que encuentre la siguiente línea:

```
If elItem = " " Then Exit Do
```

La línea es una instrucción **If** que examina los valores de **elItem** para verificar si contiene una cadena vacía. La variable **elItem** contiene el nombre de texto del ítem que se agregará al recibo. Si la variable está vacía, el usuario ha terminado de introducir datos y ejecuta la instrucción **Exit Do** para salir de la estructura **Do/Loop**. Por lo tanto, la instrucción **If** controla una ramificación de la estructura del programa.

Estructuras If de bloque

Una estructura **If** de bloque le permite utilizar una ecuación de condición lógica para decidir qué bloque de código ejecutar. La siguiente es una estructura **If** de bloque:

```

If condición1 Then
    instrucciones1
Elseif condición2
    instrucciones2
Else
    instrucciones3
End If

```

Cuando se ejecuta la instrucción **If**, se examina **condición1** y si el resultado es **True**, el bloque **instrucciones1** se ejecutará. Si **condición1** es **False**, se examina **condición2** y si el resultado es **True**, el bloque **instrucciones2** se ejecutará. La estructura puede tener múltiples cláusulas **Elseif** y cada una de ellas se examina individualmente, en busca de aquella cuya condición de como resultado **True**. Si ninguna de las condiciones es **True**, la cláusula **Else** se ejecutará. Se ejecutará sólo el bloque de instrucciones que sigue a la primera condición que da como resultado **True**, se omitirán todos aquellos que están debajo hasta llegar a la instrucción **End If**, aunque sus condiciones se evalúen como **True**.

Fórmulas lógicas

Las condiciones usadas en las instrucciones **If** son valores lógicos, fórmulas que dan como resultado un valor lógico o fórmulas numéricas cuyo resultado es cero (**False**) o no-cero (**True**). Comúnmente, las fórmulas lógicas se crean comparando dos valores utilizando uno de los operadores de comparación que se muestran en la tabla 35.2. Podrá combinar expresiones lógicas con los operadores Booleanos listados en la tabla 35.3. Para más información, busque *operadores de comparación y lógicos* en el apartado de ayuda en pantalla de Visual Basic para Aplicaciones.

Tabla 35.2 Operadores de comparación

Operador	Descripción
=	Igual a
<>	Distinto de
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

Tabla 35.3 Operadores lógicos

Operador	Descripción
And	And lógico
Eqv	Equivalencia lógica
Imp	Implicación lógica
Not	Negación lógica
Or	Or lógico
Xor	Or exclusivo lógico

Select case

La estructura **Select Case** ejecuta una función similar a la de la estructura **If** de bloque, en el sentido que una expresión se utiliza para seleccionar un bloque particular de instrucciones. En **Select Case**, la expresión devuelve un valor y ese valor se utiliza para determinar qué bloque de instrucciones ejecutar. La sintaxis de **Select Case** es la siguiente:

```

expresión Select Case
Case lista1
    instrucciones1
Case Else
    instrucciones2
End Select

```

Cuando se ejecuta la instrucción **Select Case**, se evalúa una expresión. Después de una instrucción **Select Case** aparecen una o más instrucciones **Case**. El valor de **expresión** se compara con la lista de valores delimitadas por comas en **lista1**. Si uno de los valores es afín, se ejecutará el bloque **instrucciones1**. De lo contrario, se ignorará el bloque y se revisará la siguiente instrucción **Case** para encontrar afinidad. Si no se encuentra ninguna instrucción **Case afín**, se ejecutará el bloque **instrucciones2** que sigue a **Case Else**. Al igual que con la estructura **If** de bloque, se ejecutará sólo uno de los bloques de instrucciones.

Cómo acceder a las celdas de la hoja de cálculo

En el ejemplo Creador de recibos, después de que el usuario introduce el nombre y el costo del ítem, esa información se inserta en las celdas de la hoja de cálculo. A continuación se presenta el código que ejecuta la inserción:

```
Set laHoja = Application.Workbooks ("Ejemp.xls").  
Worksheets("Recibo")
```

```
'Define the pointer to the top of the table of items.
```

```
Set OutputRange = laHoja.Range ("ParteSuperiorLista")
```

```
'Insert the item's name on the worksheet.
```

```
OutputRange.Cells(NumItems, 1).Formula = elItem
```

```
OutputRange.Cells(NumItems, 2).Formula = Str(elCosto)
```

Deberá utilizar una referencia de rango y una instrucción **Set** para definir una variable de objeto que señala a la celda denominada **ParteSuperiorLista** en la hoja de cálculo. Esa referencia de rango se utiliza con el método **Cells** para seleccionar las celdas de la hoja de cálculo en posiciones relativas a la celda **ParteSuperiorLista**. Después de obtener una referencia de rango para una celda, utilice las propiedades **Value** y **Formula** para obtener o establecer el valor de una celda. Observe que la propiedad **Valor** contiene el valor de una celda -es decir, el número que aparece en la hoja de cálculo- mientras que la propiedad **Formula** contiene lo que ha escrito en la celda. En este caso, dado que ninguna de las celdas contiene una fórmula, las propiedades **Value** y **Formula** son iguales.

Nota

La referencia de rango de la variable **OutputRange** se refiere a una celda con nombre. Es preferible emplear esta referencia que referenciar una celda explícita ya que podrá reorganizar de esta manera la hoja de cálculo y el código seguirá obteniendo el valor de la celda correcta.

Podrá referenciar las celdas usando una referencia de celda o por fila o columna, usando **Cell**, que también devuelve un objeto **Rango**. Las siguientes tres instrucciones referencian el contenido de la celda C7 en el ejemplo:

```
elCosto = laHoja.Rango("C7").Value
```

```
elCosto = laHoja.Rango("ParteSuperiorLista").Cells (1, 2).Value
```

```
elCosto = laHoja.Cells(7, 3).Value
```

El procedimiento **Totalizar** verifica la lectura y modificación de los valores de las celdas de la hoja de cálculo. La primera parte del procedimiento obtiene el valor en cada una de las celdas de la columna C que contie-

ne un valor y calcula un subtotal. Esto se muestra en el siguiente fragmento de código:

```
SubTotal = 0
'Calculate the total by extracting the values form the worksheet.
For laFila = 1 To MaxNumItems
SubTotal = SubTotal + Val(OutputRange.Cells(laFila, 2).Value)
Next laFila
```

Las instrucciones **For** y **Next** forman un bucle que ejecuta la instrucción **SubTotal** para cada fila en el rango de datos del recibo. El valor de la celda se obtiene y totaliza en la variable **SubTotal**.

Nota

Obsérvese cómo se aplica la función **Val** () al contenido de la celda para asegurar que un número ha pasado a la variable. Sin la función (), el usuario puede introducir en una celda una cadena que podría romper su código si intentara almacenarlo en una variable numérica. La función **Val** () convierte una cadena en un número y evita ese problema potencial.

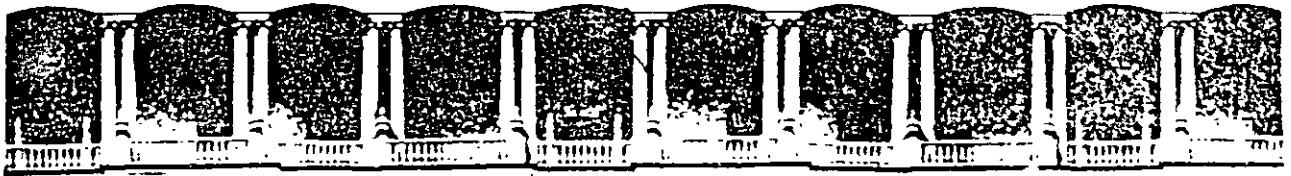
La segunda mitad del procedimiento escribe los rótulos y valores para el subtotal, el impuesto y el total en la hoja de cálculo:

```
With OutputRange
.Cells(MaxNumItem + 1, 1).Formula = "Subtotal"
.Cells(MaxNumItem + 1, 2).Formula = Str(SubTotal)
.Cells(MaxNumItem + 2, 1).Formula = "Impuesto"
ImpuestoItem = elimpuesto(SubTotal) "Calculate the tax.
.Cells(MaxNumItem + 2, 2).Formula = Str (ImpuestoItem)
elTotal = SubTotal + ImpuestoItem
.Cells(MaxNumItem + 3, 1).Formula = "Total"
.Cells(MaxNumItem + 3, 2).Formula = Str(elTotal)
End With
```

Este bloque de instrucciones también demuestran el uso de la instrucción **With** para unir las celdas en bloque y ahorrar la tarea de escritura. Las instrucciones están lógicamente unidas en bloque porque referencian al mismo objeto y permiten ahorrar escritura dado que debe introducir el objeto sólo una vez. El objeto que sigue a la cláusula **With (OutputRange)** se adjunta antes del punto a todas las instrucciones situadas entre las cláusulas **With** y **End**.

Cómo llamar a los procedimientos

El procedimiento **Totalizar** es un ejemplo de un subprocedimiento que es llamado por otros procedimientos. El procedimiento **GetEntries** necesita



**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA**

CURSOS INSTITUCIONALES

SOPORTE A MICROSOFT OFFICE PARA WINDOWS 95

Del 26 al 30 de mayo de 1997.

*Apuntes Generales
Complemento*

Palacio de Minería
1997.

Cómo acceder a las celdas de la hoja de cálculo

En el ejemplo Creador de recibos, después de que el usuario introduce el nombre y el costo del ítem, esa información se inserta en las celdas de la hoja de cálculo. A continuación se presenta el código que ejecuta la inserción:

```
Set laHoja = Application.Workbooks ("Ejemp.xls").
```

```
Worksheets("Recibo")
```

```
'Define the pointer to the top of the table of items.
```

```
Set OutputRange = laHoja.Range ("ParteSuperiorLista")
```

```
'Insert the item's name on the worksheet.
```

```
OutputRange.Cells(NumItems, 1).Formula = elItem
```

```
OutputRange.Cells(NumItems, 2).Formula = Str.(elCosto)
```

Deberá utilizar una referencia de rango y una instrucción **Set** para definir una variable de objeto que señala a la celda denominada **ParteSuperiorLista** en la hoja de cálculo. Esa referencia de rango se utiliza con el método **Cells** para seleccionar las celdas de la hoja de cálculo en posiciones relativas a la celda **ParteSuperiorLista**. Después de obtener una referencia de rango para una celda, utilice las propiedades **Value** y **Formula** para obtener o establecer el valor de una celda. Observe que la propiedad **Valor** contiene el valor de una celda -es decir, el número que aparece en la hoja de cálculo- mientras que la propiedad **Formula** contiene lo que ha escrito en la celda. En este caso, dado que ninguna de las celdas contiene una fórmula, las propiedades **Value** y **Formula** son iguales.

Nota

La referencia de rango de la variable **OutputRange** se refiere a una celda con nombre. Es preferible emplear esta referencia que referenciar una celda explícita ya que podrá reorganizar de esta manera la hoja de cálculo y el código seguirá obteniendo el valor de la celda correcta.

Podrá referenciar las celdas usando una referencia de celda o por fila o columna, usando **Cell** que también devuelve un objeto Rango. Las siguientes tres instrucciones referencian el contenido de la celda C7 en el ejemplo:

```
elCosto = laHoja.Rango("C7").Value
```

```
elCosto = laHoja.Rango("ParteSuperiorLista").Cells (1, 2).Value
```

```
elCosto = laHoja.Cells(7, 3).Value
```

El procedimiento **Totalizar** verifica la lectura y modificación de los valores de las celdas de la hoja de cálculo. La primera parte del procedimiento obtiene el valor en cada una de las celdas de la columna C que contie-

ne un valor y calcula un subtotal. Esto se muestra en el siguiente fragmento de código:

```
SubTotal = 0
'Calculate the total by extracting the values form the worksheet.
For laFila = 1 To MaxNumItems
SubTotal = SubTotal + Val(OutputRange.Cells(laFila, 2).Value)
Next laFila
```

Las instrucciones **For** y **Next** forman un bucle que ejecuta la instrucción **SubTotal** para cada fila en el rango de datos del recibo. El valor de la celda se obtiene y totaliza en la variable **SubTotal**.

Nota

Obsérvese cómo se aplica la función **Val ()** al contenido de la celda para asegurar que un número ha pasado a la variable. Sin la función **Val ()**, el usuario puede introducir en una celda una cadena que podría romper su código si intentara almacenarlo en una variable numérica. La función **Val ()** convierte una cadena en un número y evita ese problema potencial.

La segunda mitad del procedimiento escribe los rótulos y valores para el subtotal, el impuesto y el total en la hoja de cálculo:

With OutputRange

```
.Cells(MaxNumItem + 1, 1).Formula = "Subtotal"
.Cells(MaxNumItem + 1, 2).Formula = Str(SubTotal)
.Cells(MaxNumItem + 2, 1).Formula = "Impuesto"
ImpuestoItem = elimpuesto(SubTotal) 'Calculate the tax.
.Cells(MaxNumItem + 2, 2).Formula = Str (ImpuestoItem)
elTotal = SubTotal + ImpuestoItem
.Cells(MaxNumItem + 3, 1).Formula = "Total"
.Cells(MaxNumItem + 3, 2).Formula = Str(elTotal)
```

End With

Este bloque de instrucciones también demuestra el uso de la instrucción **With** para unir las celdas en bloque y ahorrar la tarea de escritura. Las instrucciones están lógicamente unidas en bloque porque referencian al mismo objeto y permiten ahorrar escritura dado que debe introducir el objeto sólo una vez. El objeto que sigue a la cláusula **With (OutputRange)** se adjunta antes del punto a todas las instrucciones situadas entre las cláusulas **With** y **End**.

Cómo llamar a los procedimientos

El procedimiento **Totalizar** es un ejemplo de un subprocedimiento que es llamado por otros procedimientos. El procedimiento **GetEntries** necesita

poder calcular y mostrar los totales en la hoja de cálculo y para ello llamará al procedimiento **Totalizar**. Ambos procedimientos pueden encontrarse en la hoja de cálculo **Creadorderecibos**.

Al principio del procedimiento **GetEntries**, deberá eliminar toda entrada anterior del recibo. Se llamará al procedimiento **ClearRange** para que ejecute esta acción:

```
Sub ClearRange (elRango As Object)
    Dim laFila As Integer
    For laFila = 1 To MaxNumItems + 3
        'Clear the cells. Use Clear contents to only clear
        'the values and not the formatting.
        elRango.Cells(laFila, 1).ClearContents
        elRango.Cells(laFila, 2).ClearContents
    Next laFila
End Sub
```

El procedimiento necesita de un objeto para un argumento denominado **elRango** en el procedimiento **ClearRange**. En el procedimiento **GetEntries**, el procedimiento **ClearRange** es llamado con la siguiente instrucción:

```
ClearRange OutputRange 'Clear the table of items.
```

Dado que esta llamada de procedimientos no forma parte de una fórmula, no será necesario colocar el argumento entre paréntesis.

El procedimiento de la función **elImpuesto** también es otro procedimiento que es llamado desde otro punto del programa:

```
'Calculate the tax on an item.
```

```
Function elImpuesto(Costo As Currency) As Currency
    Const TasalImpuesto = 0,0825
    elImpuesto = Costo * TasalImpuesto
End Function
```

Observe que en este procedimiento, el argumento es una variable de tipo de moneda denominada **Costo**. La función es reclamada como parte de una fórmula de **Totalizar**:

```
ImpuestoItem = elImpuesto(SubTotal) 'Calculate the tax
```

En **Totalizar**, la función **elImpuesto** pasa a la variable **subTotal** como un argumento. Esta variable señala una localización de memoria y ésta pasa a la función **elImpuesto**, donde se la denomina **Costo**. Ambos nombres señalan la misma localización de memoria, por lo que si el valor de **Costo** se modifica en **elImpuesto**, el valor de **SubTotal** cambiará en **Totalizar** cuando la función termine de ejecutarse.

En algunos casos, es posible que desee asegurarse de que un procedimiento no modificará un argumento, por lo que deberá pasar el argumento co-

mo un valor en lugar de pasarlo como una ubicación de memoria. Podrá hacer esto en el encabezado del procedimiento o en el programa de llamada. En un encabezado de procedimiento, por ejemplo, deberá anteponer al argumento la palabra clave **ByVal**, tal y como se muestra a continuación:

Function *elImpuesto*(**ByVal** *Costo* **As** *Currency*) **As** *Currency*

La otra posibilidad es transformar el argumento en el programa de llamada en una fórmula. La ubicación en la que se almacena el resultado de la fórmula se envía al procedimiento en lugar de las ubicaciones de cualquiera de las variables. Transforma una variable en fórmula, colocando el nombre de la variable entre paréntesis, tal y como se muestra a continuación:

Impuesto*Item* = *elImpuesto* ((*SubTotal*)) **'Calculate the tax**

El procedimiento **Totalizar** es un procedimiento evento adjunto a la hoja de cálculo. Esto se realiza al final del procedimiento **GetReceipt**, tal y como sigue:

**'Make the TotalIt procedure an event procedure attached
'to the worksheet.**

**LaHoja.OnEntry = "Totalizar" 'Volver a calcular el total si el
usuario realiza modificaciones.**

Más adelante en el programa, el procedimiento **Totalizar** aparece separado de la hoja de cálculo en el procedimiento **PrintReceipt**, tal y como sigue:

LaHoja.OnEntry = "" 'Turn off the automatic retotaling.

La propiedad **OnEntry** de una hoja de cálculo contiene el nombre de un procedimiento que debe ejecutarse en el momento en que el usuario modifica el contenido de una celda de la hoja de cálculo. Convirtiendo el procedimiento **Totalizar** en un procedimiento evento, el usuario podrá realizar cambios en el recibo antes de imprimirlo y esos cambios se incluirán de inmediato en los totales situados en la parte inferior del recibo.

Cómo usar bucles

El procedimiento **Totalizar** debe buscar entre las celdas de la hoja de cálculo y agregar contenido para calcular el subtotal. Si desea escribir referencias de rango para las 10 celdas de la hoja de cálculo, podrá hacerlo, pero probablemente no quiera perder su tiempo escribiendo la misma instrucción una y otra vez. Para solucionar estos problemas, utilice bucles.

For/Next

El bucle más común es el de **For/Next**, que ejecuta un bloque de instrucciones un determinado número de veces. La sintaxis del bucle **For/Next** es la siguiente:

```
For loopvariable = start To end Step stepval
.
.
instrucciones
.
.
Next loopvariable
```

Consejo

Planifique sus bucles en un papel antes de hacerlo en código. Le permitirá ahorrar horas de trabajo depurando infinitos bucles

En el bucle **For/Next**, **loopvariable** es una variable estándar. La primera vez que se ejecuta el bucle, **loopvariable** tiene el valor **start** y se ejecutarán todas las instrucciones situadas por debajo de la instrucción **Next**. La segunda vez que se ejecuta el bucle, se agrega **stepval** a **loopvariable** y ese valor se compara con **end**. Si **loopvariable** es mayor que **end**, el bucle finaliza. De lo contrario, las instrucciones dentro del bucle se vuelven a ejecutar. La cláusula **Step stepval** puede omitirse, en cuyo caso el **stepval** es 1. Si **stepval** es negativo, el bucle cuenta hacia abajo en lugar de hacia arriba hasta que **loopvariable** es menor que **end**.

El procedimiento **Totalizar** utiliza el bucle **For/Next** para seleccionar todas las celdas que contengan valores en la hoja de cálculo, a continuación el procedimiento calcula un total para todos los valores hallados:

```

For laFila = 1 To MaxNumItems
    SubTotal = SubTotal + Val(outputRange.Cells(laFila, 2).Value)
Next laFila

```

En este ejemplo, **laFila** es una variable de bucle y su rango va desde 1 hasta **MaxNumEntries**. Cada vez que se ejecuta el bucle, se selecciona una celda diferente de la hoja de cálculo, usando el método **Cells**.

Do/Loop

El bucle **Do/Loop** emplea una condición para determinar cuantas veces debe ejecutar el bucle. Puede comprobar la condición al comienzo o al final del bucle, y éste podrá continuar mientras la condición sea **True** o hasta que la condición de como resultado **True** (mientras sea **False**). Por eso, la sintaxis presenta cuatro variaciones, tal y como se observa a continuación:

Do While Condición

Do until condición

.instrucciones

.instrucciones

Loop

Loop

Do

Do

instrucciones

instrucciones

Loop While Condición

Loop While Condición

El procedimiento **GetEntries** utiliza una estructura **Do/Loop** para realizar el bucle de las 10 líneas de entrada permitidas en el recibo. En este caso,

la construcción **Do While** se utiliza con una condición que permanece **True** hasta que el valor de **NumItems** es menor o igual que **MaxNumItems**:

```
Do While NumItems <= MaxNumItems .
```

```
  'Get the name.
```

```
  ElItem = InputBox("Nombre del Item:", "Crear Recibo")
```

```
  'If the user didn't enter anything, he must be done, so quit.
```

```
  If elItem = "" Then Exit Do
```

```
  'Get the cost.
```

```
  ElCosto = Val(InputBox("Costo del Item:", "Crear Recibo"))
```

```
  'Insert the item's name and cost on the worksheet.
```

```
  OutputRange.Cells(NumItems, 1).Formula = elItem
```

```
  OutputRange.Cells(NumItems, 2).Formula = Str(elCosto)
```

```
  NumItems = NumItems + 1 'Increment the number of items.
```

Loop

Todas las instrucciones que se encuentran entre la cláusula **Do** y la cláusula **Loop** se ejecutarán hasta que la condición sea **False**. Si el usuario pulsa Cancelar y la instrucción **If** dentro del bucle tiene una condición **True**, ocurre una excepción:

```
If elItem = "" Then Exit Do
```

Si la condición es **True**, se ejecutará la instrucción **Exit Do**, terminando inmediatamente el bucle y comenzando la ejecución en la instrucción que sigue a la instrucción **Loop**. Existe también una instrucción **Exit For** para salir de un bucle **For/Next** en una fase temprana.

For Each

El bucle **For Each** se utiliza para realizar algunas acciones para todos los elementos de una matriz o colección. La sintaxis es la siguiente:

```
For Each elemento In grupo
    instrucciones
Next elemento
```

El bucle **For Each** se aplica sólo a matrices y colecciones. El bucle se ejecuta una vez para cada elemento de la matriz o colección. Este bucle es útil cuando no sabe (o no le importa) cuántos elementos hay en una colección. El *elemento* variable bucle es el mismo tipo de información que los elementos en la colección *grupo*. Cada vez que se calcula el bucle, el *elemento* asume el valor de otro miembro de la colección.

Cómo acceder a los archivos de disco

Si ha jugado con los ejemplos, es posible que haya observado que cada vez que crea un nuevo recibo, desaparecen todos los valores de datos. Lo

que falta es una forma de guardar los datos para que pueda recuperarlos y utilizarlos. Es posible que, quiera saber cuantos ítems y de qué tipo se han vendido, o que desee calcular el total de recibos por día para compararlos con los recibos de la caja.

Cuenta con dos opciones: puede almacenar los datos en otra hoja de cálculo para que se guarden en un libro de trabajo, o puede abrir un archivo de disco y almacenar los datos inmediatamente. Estos métodos ofrecen diferentes ventajas, dependiendo de lo que desee hacer con los datos. Si guarda los datos en una hoja de cálculo, podrá aplicarles todas las funciones de base de datos de Excel. Si los guarda en un archivo de disco, otros programas podrán abrirlos directamente. En este ejemplo va a guardar los datos en un archivo de disco. Agregue código al procedimiento **PrintRange** para añadir los datos al final de un archivo de datos, tal y como se muestra (las líneas agregadas aparecen en negrita) en la siguiente estructura:

'Print the receipt

Sub PrintReceipt ()

Dim laFila As Integer

laHoja.OnEntry = "" 'Turn off the automatic retotaling.

laHoja.Range("ImprimirRango").PrintOut 'Print the worksheet.

'Save the data from the receipt

Open "C:\data\ejemps.dat" For Append As # 1

For laFila = 1 To MaxNumItems

Write # 1, OutputRange.Cells(laFila, 1).Value,

Write # 1, OutputRange.Cells(laFila, 2).Value

Next laFila

'Write the subtotal, tax, and total.

With OutputRange

Write # 1, "SubTotal",

Write # 1, .Cells(MaxNumItems + 1, 2).Value

Write # 1, "Impuesto",

Write # 1, .Cells(MaxNumItems + 2, 2).Value

Write # 1, "Total",

Write # 1, .Cells(MaxNumItems + 3, 2).Value

End With

Close #1

End Sub

Consejo

Por lo general, no es aconsejable colocar archivos en el directorio raíz de cualquier disco duro.

En las líneas agregadas, el archivo se abre para utilizarlo como apéndice, usando un número de archivo de 1. El apéndice coloca cada nueva entrada al final del archivo. El bucle copia después los datos de la hoja de cálculo y los escribe en el archivo. El programa parece seguir funcionando igual, pero ahora los datos se guardan cada vez que se pulsa en el botón

Imprimir recibo. Después de introducir los datos y pulsar en el botón [Imprimir recibo], aparece el siguiente texto en EJEMPS.DAT:

```
"Lapices",2,35
"Cuaderno de cuentas",1,8
"Papel de impresora - caja",18,95
,
,
,
,
,
,
"Subtotal",23,1
"Impuesto",1,9058
"Total",25,0058
```

Obsérvese que la instrucción **Write** delimita los datos en el archivo colocando las cadenas de texto entre comillas y comas entre los ítems que se escriben en el disco. Estos delimitadores facilitan el uso de la instrucción **Input** para que el programa vuelva a leer los datos para su tratamiento posterior. Si utiliza **Print** en lugar de **Write**, el texto y las cadenas se escriben en el archivo sin delimitadores, creando un archivo de texto más adecuado para impresión que para ser leído en otro programa.

Para más información acerca de leer y escribir archivos, busque *input* en el apartado ayuda en pantalla de Visual Basic para Aplicaciones y seleccione el tema "Resumen de palabras claves de entrada y salida"

Cómo utilizar los cuadros de diálogo incorporados

Visual Basic para Aplicaciones ofrece dos cuadros de diálogo incorporados que podrá utilizar en sus programas para enviar y obtener datos del usuario. Estos dos cuadros de diálogo han sido creados con las funciones **MsgBox()** e **InputBox()**. Ya ha utilizado la función **InputBox()** para obtener datos del usuario en el programa Creador de recibos. La función **MsgBox** abre un cuadro de diálogo que contiene un mensaje y uno o más botones en los que podrá pulsar para cerrar el cuadro de diálogo. Ambas funciones recogen uno o más argumentos para establecer el texto de aviso, el título del cuadro, el número y tipo de botones, etc. Véase la ayuda en pantalla para obtener una lista completa de los argumentos.

Además de estos dos cuadros de diálogo, puede utilizar dos cuadros de diálogo de Excel para realzar sus programas cuando abre o guarda archivos. Los dos cuadros de diálogo se abren con los métodos **GetSaveAsFilename** y **GetOpenFilename**.

El método **GetSaveAsFilename** abre el cuadro de diálogo estándar Guardar como y obtiene un nombre de archivo del usuario. En realidad, el cua-

dro de diálogo no guarda nada, simplemente le da una vía de acceso y un nombre de archivo para usar. Después, deberá utilizar la instrucción **Open** para crear realmente el archivo y guardar algo en él. El método **GetOpenFilename** opera de la misma manera, pero en su lugar abre el cuadro de diálogo Abrir.

Cómo crear cuadros de diálogo personalizados

Además de los cuadros de diálogo incorporados, podrá crear los suyos personalizados y adjuntarlos al programa Visual Basic para Aplicaciones. Por ejemplo, el programa Creador de recibos podría utilizar un formulario de entrada de datos reemplazando los dos cuadros de diálogo necesarios para introducir una sola entrada:

Para realizar esta modificación, siga estos pasos:

1. Abra el libro de trabajo EJEMPS.XLS y guárdelo como EJEMPS2.XLS.
2. Elija **Insertar, Macro, Diálogo**. Su hoja de cálculo deberá parecerse a la de la figura 35.9, con un cuadro de diálogo personalizado en blanco y la barra de herramientas Diálogo. Esta barra de herramientas puede aparecer flotando o anclada en la parte superior o inferior de la página. Las herramientas de esta barra se listan en la tabla 35.4

Fig. 35.9
Un cuadro de diálogo personalizado antes de la edición. La barra de herramientas Diálogo aparece flotando en la parte superior de la pantalla

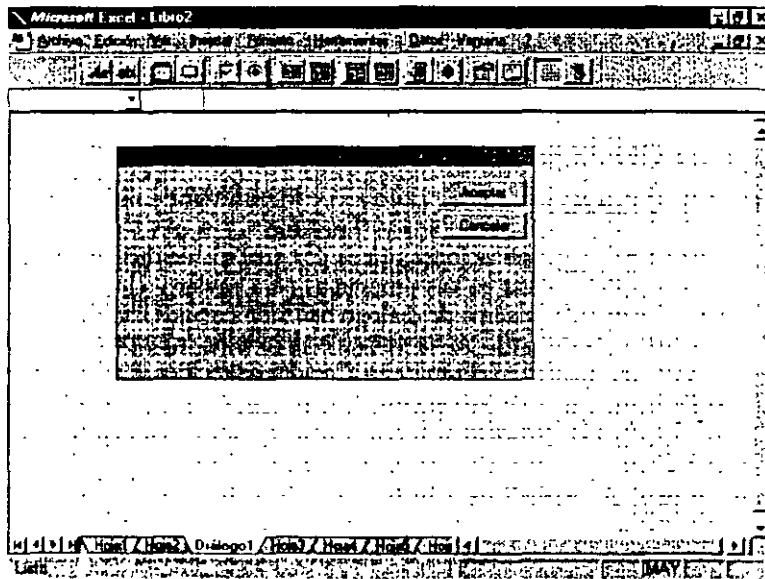






Tabla 35.4 Herramientas de la barra de Diálogo

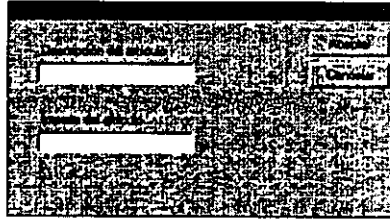
Herramienta	Nombre de la herramienta	Descripción
	Crear rótulo	Crea un rótulo de texto.
	Cuadro de edición	Crea un cuadro de edición para introducir datos.
	Cuadro de grupo	Crea un marco de grupo para agrupar visualmente otros controles y agrupar funcionalmente botones de opción.
	Crear botón	Crea un botón de comando que puede ejecutar un procedimiento.
	Crear casilla de verificación	Crea una casilla de verificación con un rótulo para seleccionar opciones no exclusivas.
	Crear botón de opción	Crea un botón de opción con un rótulo para establecer opciones exclusivas. (Crea un grupo de botones de opción dentro del marco Grupo).
	Cuadro de lista	Crea un cuadro de lista para seleccionar una valor de una lista de valores en un cuadro por el que podrá desplazarse.
	Crear desplegable	Crea un cuadro de lista desplegable para seleccionar una valor.
	Crear cuadro combinado de lista	Crea un cuadro combinado de lista (combinación entre cuadro de lista y cuadro de edición).
	Crear cuadro combinado desplegable	Crea un cuadro combinado desplegable (combinación entre cuadro de lista desplegable y cuadro de edición).
	Crear barra de desplazamiento	Crea una barra de desplazamiento para introducir un valor desplazando un marcador o para usar como indicador de un valor.
	Crear controles de giro	Crea un control de giro para pasar rápidamente de un paso a otro en una lista de valores enteros.

Herramienta	Nombre de la herramienta	Descripción
	Formato objeto	Abre el cuadro de diálogo Formato objeto para establecer las propiedades de control.
	Editar código	Busca el procedimiento asignado al control seleccionado.
	Alternar cuadrícula	Activa o desactiva una cuadrícula para simplificar la alineación de los controles en un cuadro de diálogo.
	Ejecutar diálogo	Activa el cuadro de diálogo por lo que cambiando valores o realizando una pulsación en los botones se ejecutarán los procedimientos adjuntos.



3. Seleccione el título de diálogo, escriba **Creador de recibos** y pulse en cualquier área en blanco de la hoja de diálogo.
4. Elija **Formato, Hoja, Cambiar nombre** y cambie el nombre de la hoja por **DialogItem**. Pulse en [Aceptar].
5. Usando el botón [Crear rótulo] en la barra de herramientas diálogo, dibuje dos rótulos en el cuadro de diálogo, tal y como se muestra en la figura 35.10. Seleccione el título del primero y escriba **Nombre de Item**. Seleccione el segundo título y escriba **CostoItem**.

Fig. 35.10
Diseño del
cuadro de
diálogo Creador
de recibos



6. Usando el botón [Cuadro de edición], dibuje dos cuadros en el cuadro de diálogo, tal y como se muestra en la figura 35.10
7. Seleccione el cuadro de edición próximo al rótulo Nombre de Item y cambie el nombre por **CuadroNombreItem**.

Nota

Para cambiar el nombre de un control en la hoja de diálogo, seleccione el control. Luego pulse en el cuadro de nombre en la parte izquierda de la barra de edición, escriba el nuevo nombre y pulse <Intro>

8. Seleccione el cuadro de edición próximo al rótulo Costo del Item y asígnele el nombre **CuadroCostoItem**.
9. Elija **Herramientas, Orden de tabulación**. Seleccione **CuadroNombreItem** y desplácelo a la parte superior de la lista. Seleccione **CuadroCostoItem** y desplácelo debajo de **CuadroNombreItem**. Este procedimiento hace que **CuadroNombreItem** se seleccione en primer lugar cuando se abre el cuadro de diálogo. **CuadroCostoItem** se selecciona en segundo lugar, cuando el usuario pulsa la tecla <Tab>. Elija [Aceptar] para completar los cambios en el orden de la tabulación.
10. Conmute al módulo **Creadorderecibos**, realice las siguientes modificaciones en el procedimiento **GetEntries** (los cambios aparecen en negrita) y guarde el libro de trabajo:

```
'Get entries
```

```
Sub GetEntries()
```

```
Dim elItem As String, elCosto As Currency
```

```
Dim laFila As Integer, NumItems As Integer
```

```
Dim elDialogo As Object
```

```
'Define the pointer to the worksheet.
```

```
Set laHoja = Application.Workbooks("Ejemps2.xls")  
Worksheets(Recibo)
```

```
'Define the pointer to the top of the table of items.
```

```
Set OutputRange = laHoja.Range("ParteSuperiorLista")
```

```
Set elDialogo = Application.Workbooks("Ejemps2.xls")  
DialogSheets("DialogoItem")
```

```
ClearRange OutputRange 'Clear the table of items.
```

```
NumItems = 1
```

```
'Ask for the name and cost for up to 10 items.
```

```
Do While NumItems <= MaxNumItems
```

```
Clear the edit boxes
```

```
elDialogo.EditBoxes("CuadroNombreItem").Text = ""
```

```
elDialogo.EditBoxes("CuadroCostoItem").Text = ""
```

```
elDialogo.Show
```

```
'Get the name
```

```
elItem = elDialogo.EditBoxes("CuadroNombreItem").Text
```

```
'If the user didn't enter anything, he must be done, so quit.
```

```
If elItem = "" Then Exit Do
```

```
'Get the cost
```

```
elCosto = Val(elDialogo.EditBoxes("CuadroCostoItem").Text)
```

```
'Insert the item's name and cost on the worksheet.
```

```
OutputRange.Cells(NumItems, 1).Formula = elItem
```

```

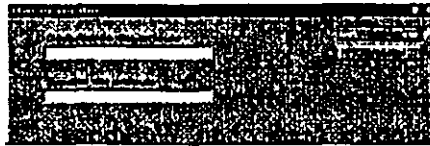
OutputRange.Cells(NumItems, 2).Formula = Str(elCosto)
NumItems = NumItems + 1 'Increment the number of items.
Loop
TotalIt 'Calculate and print the totals.
'Retotal it if the user makes changes.
LaHoja.OnEntry = "Totalizar"
End Sub

```

La primera modificación en el procedimiento define un nuevo objeto denominado **elDialogo** que referencia a la hoja de diálogo. Luego ese objeto se utiliza con la colección **EditBoxes** para despejar los dos cuadros de edición. El método **Show** se utiliza para abrir el cuadro de diálogo. Después de que el usuario elige el botón [Aceptar], el contenido de los dos cuadros de edición es devuelto al procedimiento y procesado igual que antes.

Cuando ejecuta el programa realizando una pulsación en el botón [Crear] recibo en la hoja de cálculo, aparece el cuadro de diálogo, tal y como se muestra en la figura 35.11. Escriba el nombre del ítem, pulse <Tab>, escriba el costo del ítem y pulse <Intro>. El primer ítem se insertará en el recibo y reaparecerá el cuadro de diálogo. Para finalizar una entrada, pulse <Intro> sin escribir en el cuadro de diálogo. Se calcularán los totales.

Fig. 35.11
El cuadro de
diálogo Crear
recibo.



Cómo utilizar las herramientas de depuración

Los fallos de programación son algo muy común en la vida de los programadores de ordenadores. Independientemente de lo cuidadosos que sean, casi siempre aparecen fallos. Debe encontrarlos y eliminarlos de sus programas. Los fallos más sencillos son los *errores de sintaxis*, en los que los parámetros de llamada son incorrectos. VBA encuentra normalmente los errores de sintaxis tan pronto como los comete. Otros fallos son los *errores de tiempo de ejecución*, que surgen cuando se utiliza un tipo de variable equivocada o cuando se ejecuta un cálculo numérico incorrecto (por ejemplo, la raíz cuadrada de -1). VBA también detecta estos errores tan pronto como se ejecuta la instrucción incorrecta. Finalmente, existen los *errores lógicos*, por los cuales un programa no realizará lo que ordena. Los errores lógicos son los más difíciles de detectar porque el código de programa compila, pero el programa simplemente no realiza lo que había planeado que hiciera.

Visual Basic para Aplicaciones cuenta con una serie de poderosas herramientas de depuración que le ayudarán a detectar y corregir los fallos de programación. Puede colocar puntos de interrupción en cualquier lugar del programa para forzarlo a detener la ejecución en ese punto. Después de detener el programa, elija **Herramientas, Inspección instantánea** para ver los valores de cada variable o expresión. Luego podrá seguir ejecutando un programa o pasar una instrucción por vez hasta encontrar el problema. También podrá establecer puntos de inspección que interrumpan automáticamente el programa cuando una variable o expresión alcance un valor determinado.

Modo Interrupción

El modo Interrupción es el punto en el que se detiene un programa en ejecución con todas sus variables intactas. Por lo general, cuando finaliza un programa, se pierde el contenido de todas las variables. No obstante, el modo interrupción es una pausa durante la ejecución del programa, por lo que el contenido de las variables a los que se ha asignado valores durante la ejecución siguen estando disponibles.

Nota

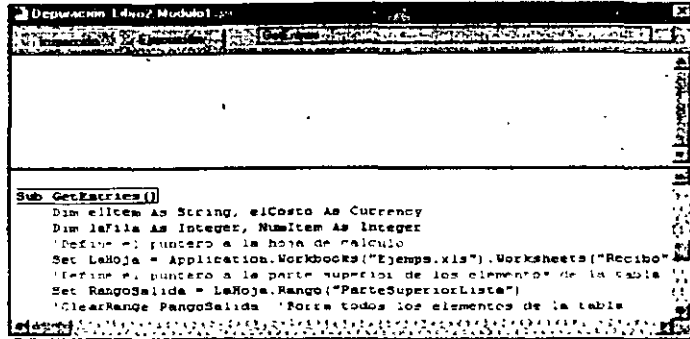
Un programa en ejecución entra en el modo interrupción cuando pulsa <Control> + <Inter>, al encontrarse con un error o un punto de interrupción o de inspección.

Cuando un programa entra en el modo interrupción, por encontrar un error o porque ha pulsado <Control> + <Inter>, aparece el cuadro de diálogo Error de macro, que le ofrece la posibilidad de salir, continuar o abrir la ventana Depuración.

Puntos de interrupción y puntos de inspección

Los puntos de interrupción y los puntos de inspección también hacen que un programa entre en el modo interrupción. Un *punto de interrupción* es un marcador en una línea de código que obliga a un programa a detener la ejecución cuando Visual Basic para Aplicaciones intenta ejecutar la línea marcada. Un *punto de inspección* es un marcador en el valor de una variable o de una simple fórmula. Cuando el valor de un punto de inspección cambia de una determinada manera, el programa se detiene y entra en el modo interrupción. Véase en la figura 35.12 el punto de interrupción **Sub GetEntries()**. Cuando se introduce este procedimiento, se establece el modo interrupción. Dado que se ha introducido un punto de interrupción, no existen datos en el Panel de inspección. ¡No hay datos para inspeccionar!

Fig. 35.12
La ventana
Depuración con
un punto de
interrupción
establecido



— —

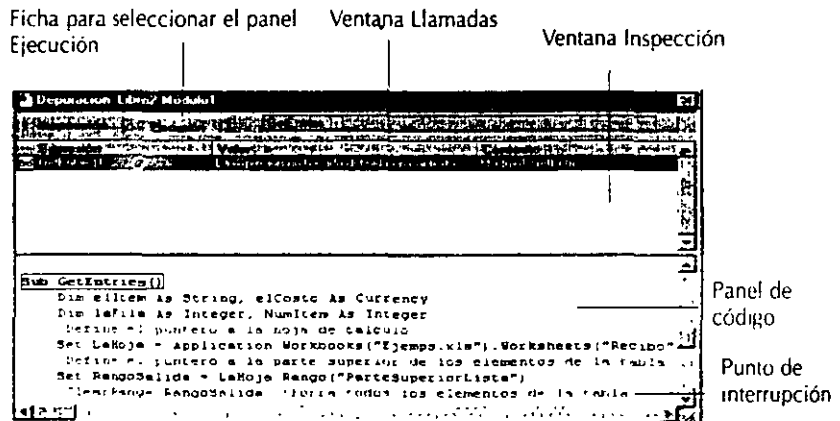
— —

Para colocar un punto de interrupción, abra el módulo que contiene su procedimiento y seleccione la línea de código en la que desea que el programa se interrumpa. Elija **Ejecutar, Alternar punto de interrupción** para introducir un punto de interrupción. Elija nuevamente el comando para eliminar un punto de interrupción seleccionado o elija **Ejecutar, Eliminar** todos los puntos de interrupción para quitarlos todos. Luego ejecute su código. Cuando alcance el punto de interrupción, el código se detendrá y entrará en modo interrupción. Cuando un programa entra en el modo interrupción por haber encontrado un punto de interrupción o inspección, abra directamente la ventana de depuración, que se describe en el siguiente apartado.

La ventana Depuración

Si elige **Depurar** en el cuadro de diálogo Error de macro, o se encuentra con un punto de interrupción o inspección, aparece la ventana Depuración. (véase la figura 35.13). La ventana Depuración es una ventana dividida, que presenta el procedimiento que se está ejecutando en su mitad inferior y el panel Ejecución o panel Inspección en la parte superior. En la mitad inferior de la ventana, podrá seleccionar líneas de código, agregar o quitar puntos de interrupción y seleccionar código para los puntos de inspección.

Fig. 35.13
La ventana
Depuración para
un código
detenido en un
punto de
inspección.



La ventana Depuración que se muestra en la figura 35.13 presenta el programa detenido en un punto de interrupción establecido en la instrucción **If** destacada en el Panel de código. El panel Inspección muestra los valores actuales de **elItem**, **elCosto** y **elDiálogo.EditBoxes("CuadroCostoItem").Text** como variables de inspección. Observe que **elCosto** aún no tiene valor porque todavía no ha pasado el valor en **elDiálogo.EditBoxes("CuadroCostoItem").Text**. En este punto, puede continuar la ejecución del procedimiento, establecer o eliminar puntos de inspección, examinar el valor de las variables o ejecutar el procedimiento instrucción por instrucción.

El panel Ejecución

En el panel Ejecución de la ventana Depuración, podrá escribir o ejecutar casi todos los comandos de Visual Basic para Aplicaciones. La única restricción radica en que el comando deberá tener una extensión máxima de una línea. El panel Ejecución recibe también todos los valores impresos generados por la instrucción **Debug.Print**, que se utilizan para imprimir los valores de un programa que está siendo ejecutado.

El panel Inspección y expresiones de inspección

El panel Inspección muestra el valor actual de los puntos de inspección y expresiones de inspección. Puntos de inspección y expresiones de inspección aparecen en el panel Inspección y muestran continuamente el valor actual de las variables y expresiones. La diferencia entre ambos radica en que aunque los dos muestran un valor, un punto de inspección puede detener su programa si el valor seleccionado se modifica de alguna manera. Utiliza Inspección instantánea para ver el valor actual de una variable o expresión sin necesidad de colocarla en el panel Inspección.

La figura 35.14 muestra el resultado de seleccionar la variable **elItem** en la ventana Depuración y elegir **Herramientas, Inspección instantánea**. Si elige el botón [Agregar], la variable Inspección instantánea se convierte en una expresión de inspección y es añadida al panel Inspección.

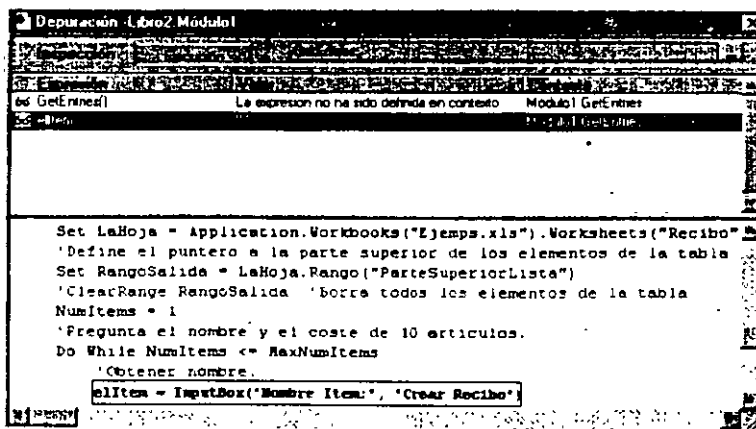


Fig. 35.14
Un panel de
Inspección
instantánea.

Los comandos paso



Llegado este punto, puede utilizar dos comandos paso para ejecutar una línea de su programa y detenerlo otra vez en el modo interrupción: **Ejecutar, Recorrer todo** y **Ejecutar, Recorrer principal**. El comando **Ejecutar, Recorrer todo** hace que el programa ejecute una línea a la vez. Si el programa se encuentra con un procedimiento, el siguiente paso tiene lugar dentro de ese procedimiento llamado.



El comando **Ejecutar, Recorrer principal** es similar, pero cuando encuentra una de procedimiento, ejecuta el procedimiento por completo antes de detenerse y entrar nuevamente en el modo interrupción. Por eso, el comando **Recorrer principal** aparece para recorrer las llamadas de procedimientos en el procedimiento que está ejecutando.

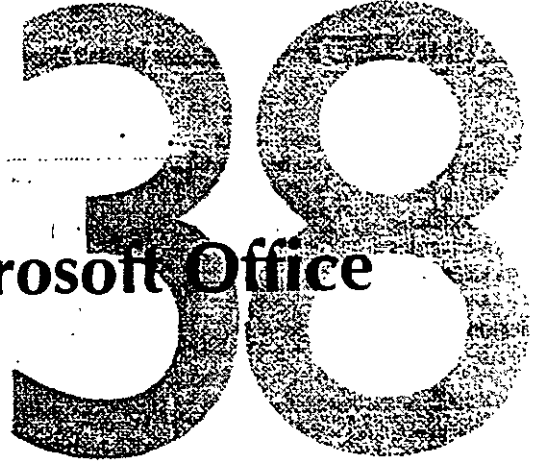
La ventana Llamadas



La ventana situada en la parte superior derecha de la ventana Depuración que se muestra en la figura 35.14, presenta el procedimiento introducido. Si pulsa en el cuadro con puntos suspensivos en el extremo derecho de la ventana **DEPURACIÓN**, justo debajo de la **X**, se abrirá la ventana **Llamadas**. Esta ventana muestra el nombre del procedimiento que contiene el punto actual de ejecución. Si selecciona la ventana **Llamadas**, ésta se amplía y lista todos los procedimientos activos en este programa. Procedimientos activos son aquellos que o bien están ejecutando la tarea actual o bien pueden haber llamado a otro procedimiento para realizar otra tarea.

Cómo usar Microsoft Office en una red

por Robert Garrison



John Donne escribió: "Ningún hombre es una isla en sí mismo."

Como usuario de Microsoft Office, tampoco es un usuario en sí mismo. Después de todo, ¿tendría un ordenador para no compartirlo *nunca* con nadie?

Cuando usa Microsoft Office en una red, existe una mejora potencial en su productividad. Obtiene este aumento de productividad trabajando en equipo.

En este capítulo aprenderá a:

- Usar diccionarios compartidos en Word
- Anotar documentos de Word
- Usar revisiones de documentos de Word
- Programar reuniones con Schedule+
- Utilizar Uso compartido de archivos en Excel
- Utilizar Presentación remota en PowerPoint

Cómo establecer diccionarios compartidos

En el capítulo 3 aprendió a usar diccionarios para verificar la ortografía en sus documentos. Si se encuentra en una red trabajando con otros usuarios, es posible que desee compartir diccionarios comunes. También es posible que en su oficina o industria se emplee un diccionario de términos específicos. Puede añadir hasta ocho diccionarios para agilizar su trabajo. El diccionario personalizado por defecto se llama *Personalizado*. Cuando agrega palabras durante una revisión ortográfica, éstas se añaden al diccionario personalizado. En este apartado usará Word para aprender más cosas sobre los diccionarios.

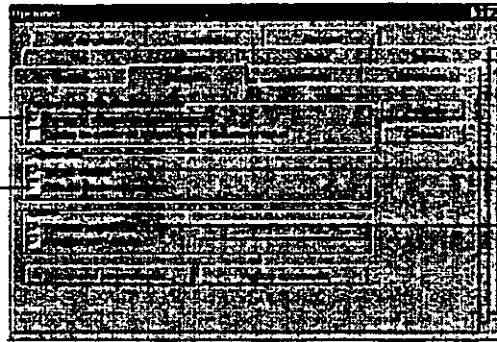
Para crear un diccionario compartido, siga estos pasos:

1. Elija **Herramientas, Opciones**.
2. Pulse en la ficha **Ortografía**. Aparecerá el cuadro de diálogo **Opciones**, tal y como se muestra en la figura 38.1.

◀ VÉASE "Cómo revisar la ortografía", pág. 90

Fig. 38.1
La ficha Ortografía le permite personalizar las directrices para usar los diccionarios.

A medida que escribe, se verifica la ortografía de las palabras



Sugiere ortografías posibles

Impide que el corrector ortográfico muestre palabras con todas las letras en mayúsculas

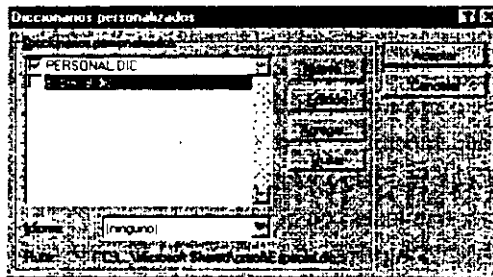
Consejo

Las palabras escritas de forma incorrecta aparecen indicadas con una línea ondulada roja (si se establece esta opción).

Marque aquí cuando no desee sugerencias de los diccionarios personalizados

3. Para añadir un diccionario personalizado, pulse en **Diccionarios Personalizados**. Aparecerá el cuadro de la figura 38.2.

Fig. 38.2
Al trabajar con diccionarios personalizados, Office aprende sus palabras específicas



Use los botones de opción de Diccionarios Personalizados para hacer lo siguiente:

- **Nuevo** le permite crear un diccionario personalizado.
- **Edición** le permite editar el diccionario seleccionado. El diccionario es un documento de Word.
- **Agregar** le permite hacer que un diccionario existente, probablemente localizado en una unidad de disco de red, sea verificado para la ortografía.
- **Quitar** elimina un diccionario de su lista de diccionarios personalizados.
- **Ruta** le muestra la ubicación del diccionario personalizado seleccionado.

4. Pulse en **Nuevo** para crear un nuevo diccionario. Aparecerá el cuadro de diálogo Crear diccionario personalizado (véase la figura 38.3)
5. Desplácese por el cuadro de texto **Guardar en** para colocar el diccionario personalizado en un directorio de red compartida. Así, otros

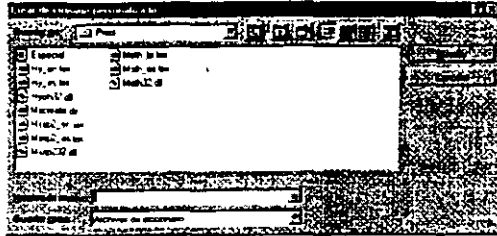


Fig. 38.3 Word puede almacenar sus palabras específicas si le proporciona un nuevo diccionario.

◀ VÉASE "Cómo Usar Cuadros de Diálogo Archivo", pág. 98

- podrán usar su diccionario si agregan el diccionario a su lista de diccionarios personalizados.
6. En el cuadro de texto **Nombre de archivo**, escriba el nombre del nuevo diccionario. Por ejemplo, escriba **Especial** y pulse en **Guardar**. Se creará un documento estándar de Word.
 7. Dado que se trata de un nuevo diccionario, no encontrará palabras en él. Puede elegir **Edición** para agregar palabras manualmente o agregarlas durante una revisión ortográfica.
 8. Cuando haya terminado, realice una doble pulsación en [Aceptar] para volver al documento.

Consejo

Cuando escriba el nombre de su diccionario, no escriba un punto ni una extensión. Word añade la extensión DIC automáticamente.

Nota

Si tiene un lenguaje específico o términos técnicos que usa en su trabajo, podrá colocarlos en un diccionario personalizado y usarlos cuando los necesite. Eliminar diccionarios personalizados puede reducir el tiempo de revisión de un documento si no utiliza ese tipo de palabras en el diccionario todo el tiempo.

Nota

¿Para qué compartir diccionarios? Si en su red hay un número elevado de usuarios, cada uno de los cuales agrega las mismas palabras a diccionarios personalizados individuales, el gasto de espacio es muy alto. Además, es posible que no todos conozcan la ortografía correcta. El uso de un diccionario compartido le ayudará a cumplir con los estándares de la compañía de forma coherente.

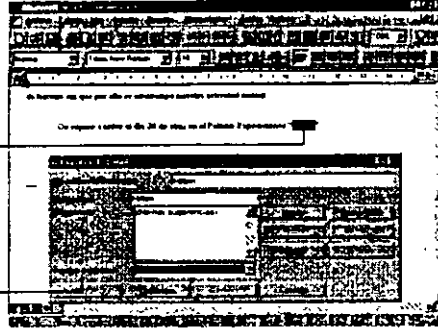
Para agregar palabras a su diccionario personalizado durante una revisión ortográfica, siga estos pasos:

1. Elija **Herramientas, Ortografía** para iniciar la revisión ortográfica. Aparecerá el cuadro de diálogo Ortografía (véase la figura 38.4).
2. Para agregar nuevas palabras a su diccionario especial, elija **ESPECIAL.DIC** de la lista desplegable **Agregar palabras a:**
3. Pulse en **Agregar** para agregar la palabra no reconocida al diccionario especial.

Fig. 38.4
Ahora puede verificar la ortografía del documento y agregar al diccionario las palabras especiales.

La palabra que puede ser reemplazada

Le permite seleccionar su diccionario especial



Cómo usar anotaciones

◀ VÉASE "Cómo trabajar con correo distribuido", pág. 730

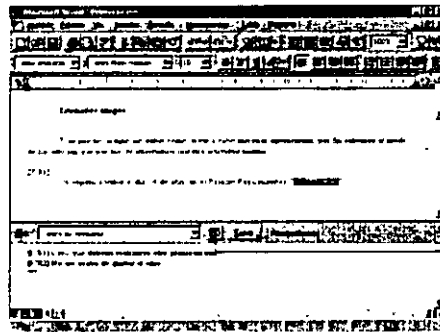
— —

Anotaciones permite a los autores hacer comentarios en un documento sin modificarlo. Un uso particular de las anotaciones puede llevarse a cabo dentro de un documento que está siendo distribuido por Microsoft Exchange.

Para hacer anotaciones en un documento, siga estos pasos:

1. Abra el documento Word.
2. Coloque el punto de inserción en donde desea insertar una anotación.
3. Elija **Insertar, Anotación**. Se abrirá la ventana *Anotaciones*, tal y como se muestra en la figura 38.5.

Fig. 38.5
La ventana *Anotaciones* contiene comentarios y sugerencias acerca del documento que esta revisando



Insertar Objeto de sonido le permite incluir anotaciones sonoras si tiene una tarjeta de sonido en su sistema

Consejo

Si las iniciales no son correctas, cámbielas contando la ficha **Información del usuario** después de elegir **Herramientas, Opciones**.

4. Escriba su comentario en la ventana *Anotaciones*. El cursor se colocará automáticamente allí al elegir **Insertar, Anotación**. Fíjese que cada anotación comienza con las iniciales del autor y que va numerada secuencialmente.
5. Pulse en [Cerrar] cuando termine con esta anotación.
6. Si desea ver todas sus anotaciones, o las de otras revisiones, elija **Ver, Anotaciones**. La ventana *Anotaciones*, mostrada en la figura 38.5, se mantendrá abierta en la pantalla.

Cómo usar marcadores de revisión

Los marcadores de revisión son diferentes de las anotaciones. Mientras que las anotaciones son comentarios que no alteran el documento, los marcadores de revisión indican los cambios hechos en el documento. Revisiones como la inserción, eliminación o modificación del texto pueden aparecer en colores y fuentes de texto diferentes. Las revisiones de cada autor pueden marcarse en un color distinto. Hasta ocho autores pueden revisar un documento sin que se repita los colores. Puede ver cómo aparecerán las revisiones en su documento eligiendo **Herramientas, Opciones** y realizando una pulsación en la ficha Revisiones.

Puede hacer revisiones en su documento siguiendo estos pasos:

1. Elija **Herramientas, Revisiones**. Se abrirá el cuadro de diálogo Revisiones (véase la figura 38.6).

◀ VÉASE "Cómo modificar las opciones de revisión", pág. 219

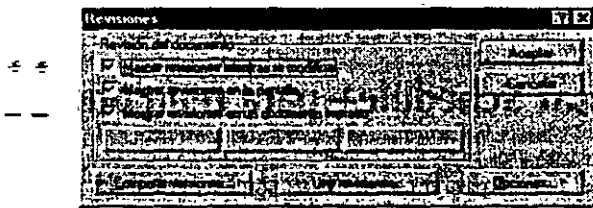


Fig. 38.6
Use el cuadro de diálogo Revisiones para indicar las modificaciones en el texto.

Las siguientes opciones están disponibles:

- **Comparar Versiones** compara el documento en curso con otro en disco. Todos los cambios se marcarán automáticamente.
 - **Unir revisiones** aplica los cambios de un documento revisado al documento original.
 - **Opciones** le permite fijar el color y el estilo del texto insertado, eliminado o modificado.
2. Pulse en el cuadro **Marcar** revisiones mientras se modifica y elija [Aceptar]. Sus revisiones aparecerán en pantalla a medida que las vaya realizando.
 3. Ahora haga algunos cambios. Por ejemplo, primero seleccione una frase y córtela. Después, inserte esa frase al comienzo del párrafo.

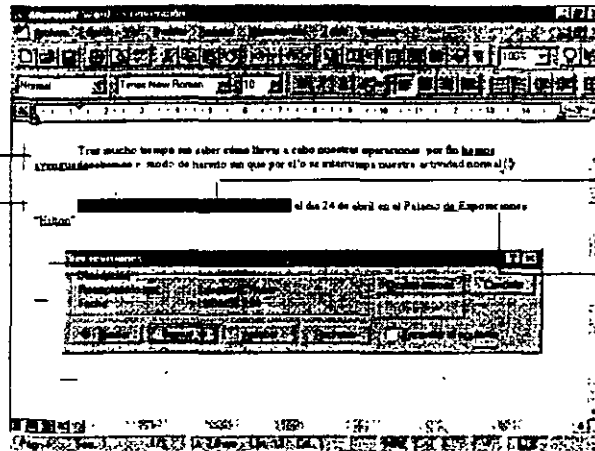
Su documento mostrará revisiones similares a las de la figura 38.7.

Cuando termine con las revisiones, puede guardar el archivo o enviarlo a la siguiente persona si el archivo le fue distribuido. Una vez que todos han hecho todos los comentarios y revisiones necesarios, alguien deberá decidir cuáles son las revisiones que hay que aceptar para que formen parte del documento. Para ello, siga estos pasos:

1. Elija **Herramientas, Revisiones**.
2. Pulse en **Revisar**, y se abrirá el cuadro de diálogo Ver revisiones (véase la figura 38.7).

Fig. 38.7
 Cuando se activan los marcadores de revisión, Word muestra qué texto ha sido modificado.

Todo el texto que presenta marcadores de revisión viene indicado por barras verticales en los márgenes



El texto insertado aparece con subrayado doble

El texto eliminado aparece tachado

Use el cuadro de diálogo Ver revisiones para aceptar o rechazar las revisiones propuestas en su documento.

Cómo usar Schedule+ para reuniones

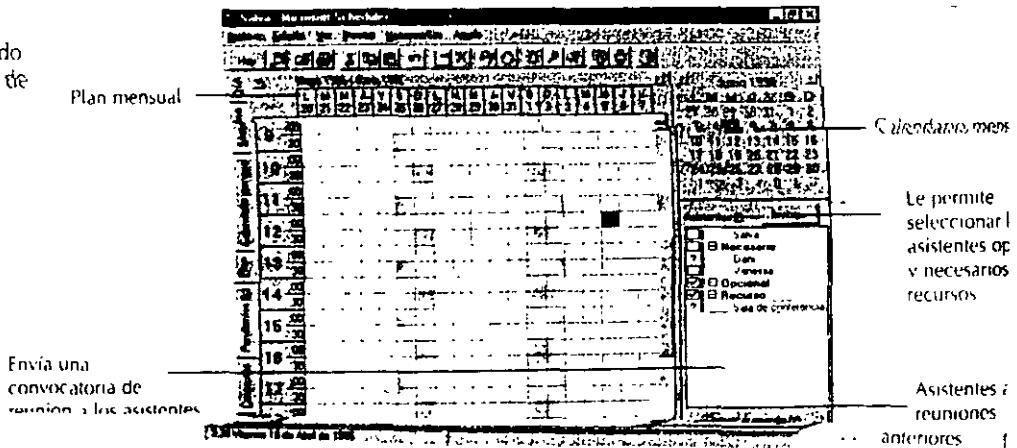
Uno de los mayores retos que plantean los lugares de trabajo hoy en día, es encontrar el tiempo para reunirse, hacer planes y tomar decisiones. Si todos sus colaboradores están usando Microsoft Office (y, por supuesto, lo están haciendo) puede utilizar Schedule+ para programar esas reuniones a la conveniencia de todos los asistentes.

► VÉASE "Cómo programar su tiempo", pág. 834

Para programar una reunión con colaboradores, siga estos pasos:

1. Conmute a Schedule+ y pulse en la ficha Plan: La ventana Plan de su agenda aparecerá, tal y como se muestra en la figura 38.8. Fíjese que los participantes de cualquier reunión anterior aparecerán en el cuadro Asistentes para facilitar la tarea de planificación.

Fig. 38.8
 Planificando reuniones de grupo



Plan mensual

Calendario mensual

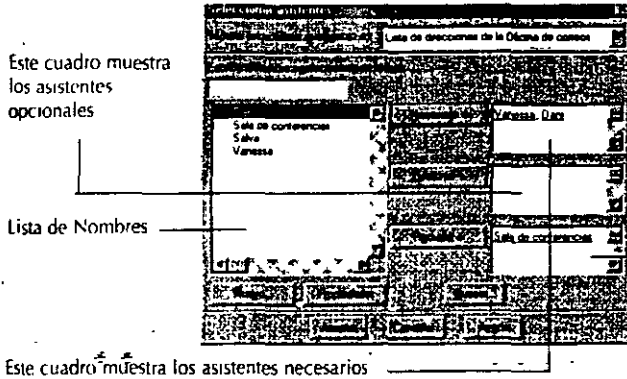
Le permite seleccionar asistentes o v necesarios recursos

Envía una convocatoria de reunión a los asistentes

Asistentes a reuniones anteriores

2. Seleccione una hora para la reunión manteniendo presionado el botón del ratón y arrastrando éste hasta el horario de finalización de la reunión.
3. Pulse en **Invitar** para seleccionar a los asistentes. Aparecerá el cuadro de diálogo Asistentes, tal como se muestra en la figura 38.9.

► VÉASE "Cómo crear un mensaje", pág. 717



Recursos necesarios para la reunión. Los recursos vienen determinados por el administrador del sistema, y pueden incluir habitaciones o equipo.

Fig. 38.9
Invitando a su personal a una reunión.

4. Seleccione la libreta de direcciones apropiada usando la lista desplegable **Mostrar los nombres** que figuran en, normalmente la lista de direcciones del servidor de correo.
5. Seleccione un nombre de persona o de recurso en la lista de nombres.
6. Pulse en **Necesario** para agregar una persona a la lista de asistencia necesaria.
7. Pulse en **Opcional** si la asistencia de la persona no es obligatoria.
8. Pulse en **Recurso** si el nombre de la lista es un recurso.
9. Repita los pasos 6-8 todas las veces que sea necesario. Pulse en [Aceptar] cuando termine (véase la figura 38.10).
10. Pulse y arrastre cubriendo el intervalo de duración de la reunión.
11. Pulse en **Convocar reunión** y se abrirá una ventana de mensaje Microsoft Exchange. Rellene la el cuadro de texto Asunto y escriba el

Consejo

Para eliminar un nombre de una de las tres listas, pulse en el nombre y pulse <Supr>. Asegúrese de borrar los signos de punto y coma que separan los nombres al eliminar un nombre

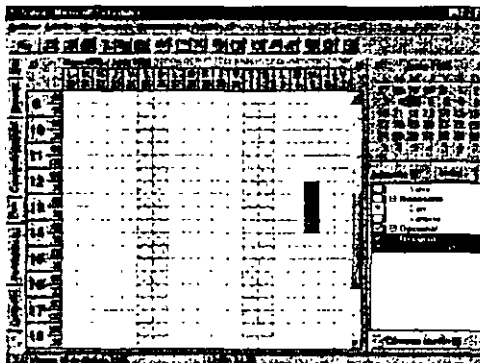


Fig. 38.10
El programa de reunión completo.

texto del mensaje. Pulse en [Enviar] para que Microsoft Exchange envíe por correo la convocatoria de reunión.

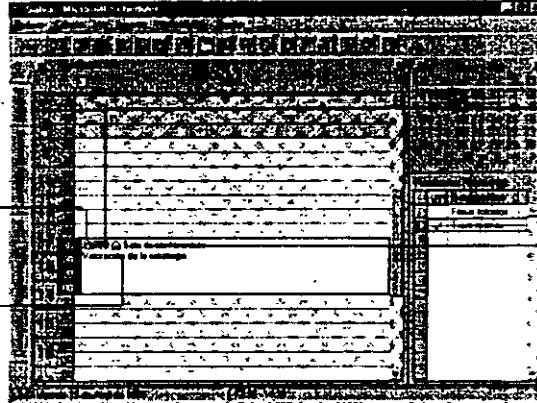
12. Pulse en la ficha **Día** y verá su agenda personal para ese día (véase la figura 38.11).

Fig. 38.11

Vease la reunión programada a partir de las citas del día, del mensaje de convocatoria de la reunión

El asunto de la reunión es el mismo que aparece en el cuadro de texto Asunto

Aviso de alarma



El icono Reunión indica una reunión programada en lugar de una cita

El lugar de la reunión aparece tras el icono casa

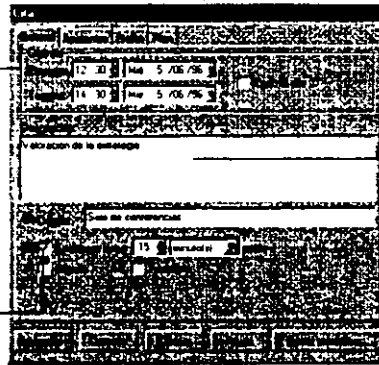
13. Si desea información adicional sobre la reunión o desea cambiar la hora del aviso, realice una doble pulsación en uno de los iconos. Se abrirá la ventana Citas (véase la figura 38.12).

Fig. 38.12

La ventana Cita muestra los detalles de la reunión programada.

La ficha General muestra la hora de la reunión

La hora de aviso, que puede ser ajustada

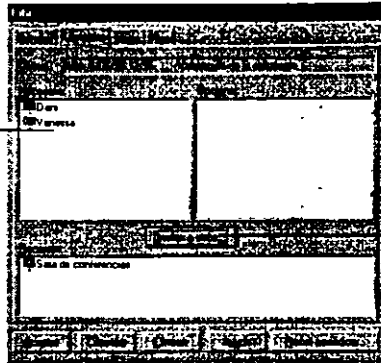


Una descripción que puede ser editada

14. Pulse en la ficha **Asistentes** para acceder a información adicional (véase la figura 38.13).

También puede crear notas sobre la reunión para su uso personal. Las notas originales son el texto del mensaje de convocatoria. Pulse en la ficha **Notas**. La ficha **Plan** muestra una mini-vista del Plan.

El signo ? delante del nombre indica que el invitado no ha respondido



Le permite añadir más personas a la reunión

Fig. 38.13
Cómo comprobar quién más asistirá a la reunión.

Cómo utilizar Uso compartido de archivos en Excel

Cuando esté trabajando en una red con otros usuarios, querrá utilizar la función Uso compartido de archivos de Excel para algunas de sus hojas de cálculo. Uso compartido de archivos permite que múltiples usuarios puedan acceder y modificar una hoja de cálculo en una unidad de disco compartida. Si un archivo se guarda cuando esta opción está activada, no se podrán modificar los formatos de celda ni las fórmulas.

Para utilizar Uso compartido de archivos, siga estos pasos:

1. Pulse en **Archivo, Compartir** y después pulse en la ficha Edición. Aparecerá el cuadro de diálogo Uso compartido de archivos (véase la figura 38.14).

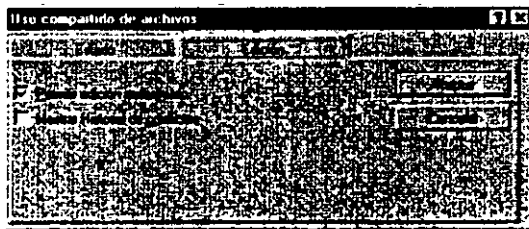


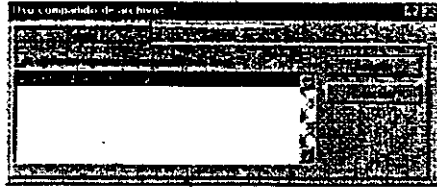
Fig. 38.14
A través de Uso compartido de archivos, otros usuarios en red podrán acceder simultáneamente a una hoja de cálculo.

2. Pulse en la ficha Estado del cuadro de diálogo mencionado para ver quién tiene el archivo abierto (véase la figura 38.15).

Si efectúa un cambio en la hoja de cálculo e intenta guardarlo mientras otro usuario está usandola, se le preguntará si desea actualizar el archivo compartido. Para poder conocer los cambios realizados, utilice **Mostrar historial**

de conflictos. Cuando se activa la opción **Mostrar historial de conflictos** para un archivo, se agrega una hoja de cálculo al libro de trabajo que proporciona un seguimiento de los cambios efectuados.

Fig. 38.15
Excel mantiene el seguimiento de los usuarios que tienen un archivo abierto cuando se activa la opción **Uso compartido de archivos para ese archivo.**



Cómo utilizar una presentación remota

El comando **Herramientas, Presentación remota** le permite revisar una presentación de PowerPoint durante su desarrollo o realizar la presentación final a través de la red. Los asistentes no tienen que desplazarse para ver la presentación, sino que permanecen sentados a sus mesas de trabajo.

Para llevar a cabo una conferencia, use el asistente para Presentaciones remotas para preparar la presentación. Como moderador de la conferencia puede usar **Notas de la reunión** para crear notas en el transcurso de la presentación, agregar actas y planes de actuación. También puede utilizar **Desplazamiento por diapositivas** para ver y bifurcar hacia sus diapositivas a medida que avanza en su presentación.

Para organizar la **Presentación remota** tiene que conocer los nombres de los ordenadores de los asistentes. Los participantes pueden obtener el nombre de sus ordenadores iniciando **Red** desde el **Panel de Control** y realizando una pulsación en la ficha **Identificación**.

Capítulo **39**

Ron Person ha escrito más de 18 libros para la Corporación Que, incluidos la Edición Especial de Cómo Utilizar Word 6 para Windows, la Edición Especial de Cómo Utilizar Excel para Windows 95, y fue el autor principal la Edición Especial de Cómo Utilizar Windows 95. Siendo uno de los socios asesores veteranos de Microsoft, Ron ha estado desarrollando las soluciones de Microsoft Office y preparando a los encargados del desarrollo corporativo durante siete años.

Cómo Construir Formularios y Cuadros de Diálogo Llenar

En el pasado, una de las labores de oficina que los procesadores de texto no podían realizar correctamente era rellenar formularios. Se necesitaba siempre una máquina de escribir para rellenar un formulario. Por otra parte, los almacenes y archivadores ocupaban demasiado espacio sólo para guardar inventarios y formularios mensuales, que en algunos casos se empleaban tan raramente que incluso vencían antes de ser retirados de los estantes.

Las nuevas funciones de formularios de Word constituyen un gran paso hacia la supresión de los formularios preimpresos. Ahora, con las funciones de autoedición de Word, muchas compañías diseñan formularios que guardan como *plantillas*, y que se imprimen cuando es necesario. El ahorro en el costo de impresión de gran cantidad de formularios puede ser inmenso.

En la actualidad, Word incluye (además de su capacidad de producir formularios de alta calidad a petición) funciones que facilitan la tarea de rellenar formularios. Usando los campos de for-

Técnicas de los Profesionales

por Ron Person

ulario de Word, podrá colocar cuadros de edición, casillas de verificación y listas desplegables directamente en sus documentos. El uso de los campos {fill in} y {ask} hace posible que un documento muestre cuadros de diálogo solicitando información.

En este capítulo aprenderá a:

- Crear una plantilla en la que podrá colocar cuadros de edición, casillas de verificación o listas desplegables
- Bloquear la plantilla para impedir que los usuarios cambien partes no autorizadas del documento
- Especificar listas de datos que aparecen en listas desplegables, o formatear los datos introducidos por un usuario en un cuadro de introducción de datos
- Usar un campo {fill in} para abrir un cuadro de diálogo
- Solicitar una sola vez un ítem de información y utilizarlo varias veces a lo largo del documento
- Crear una macro sencilla que controle cada cuadro de diálogo al abrirse

— Conceptos básicos de los formularios

Un *formulario* es un tipo especial de documento protegido que incluye campos en los que se puede escribir información. Cualquier documento que incluye campos de formulario es un formulario. Un *campo de formulario* es una ubicación en pantalla en la que puede hacer una de estas tres cosas: introducir texto, activar o desactivar una casilla de verificación o seleccionar de una lista desplegable.

Dado que las celdas de una tabla son el marco ideal para los rótulos de un formulario y para los campos de información, muchos formularios basan su estructura en una tabla. Podrá escribir rótulos en algunas celdas e insertar campos de formulario en otras. Las tablas también facilitan la tarea de agregar sombreados y bordes a un formulario. Podrá colocar un borde oscuro alrededor de un grupo de celdas seleccionadas en una tabla, por ejemplo, dejando el resto de las celdas sin borde. Una tabla no tiene por qué tener aspecto de tabla en absoluto, si las cuadrículas están desactivadas, por lo que constituye el marco ideal para un formulario.

Un formulario puede basarse en cualquier tipo de documento. Un contrato inmobiliario, por ejemplo, puede incluir varias páginas de párrafos descriptivos que contienen campos de formulario en los que podrá insertar información. El texto de los párrafos no cambiará ya que insertará la información únicamente en los campos de formulario.

Puede incluir tres tipos de campo de formulario en un formulario: texto, casilla de verificación y lista desplegable. Asimismo puede personalizar cada uno de estos tipos de campo de muchas maneras. Por ejemplo, podrá formatear un campo de texto para que acepte únicamente fechas e imprimir fechas del tipo 1 de enero, 1995, como 1/1/95, o en cualquier otro formato. Las figuras 39.1, 39.2 y 39.3 muestran ejemplos de los tres formularios que se pueden crear usando Word.

Libro	Pedido	Fecha pedido	Entregado	Fecha entrega	Cantidad	Pendencias
Libro estepona	12345	15/04	15/04	15/04	10	10
Libro estepona	12346	15/04	15/04	15/04	10	10
Libro estepona	12347	15/04	15/04	15/04	10	10
Libro estepona	12348	15/04	15/04	15/04	10	10
Libro estepona	12349	15/04	15/04	15/04	10	10
Libro estepona	12350	15/04	15/04	15/04	10	10
Libro estepona	12351	15/04	15/04	15/04	10	10
Libro estepona	12352	15/04	15/04	15/04	10	10
Libro estepona	12353	15/04	15/04	15/04	10	10
Libro estepona	12354	15/04	15/04	15/04	10	10
Libro estepona	12355	15/04	15/04	15/04	10	10
Total pedido	0			Total entregado	0	

Fig. 39.1
Ejemplo de un formulario de pedido/entrega.

Nombre de la compañía
 Estepona
 Dirección de la compañía
 Código postal Ciudad (Estado o provincia)
 (000) 000 0000 - Fax (000) 000 0000

FACTURA

Nº DE FACTURA: []
 FECHA: 19 de abril de 1996

Dirección de factura: []
 Dirección de entrega: []

REPRESENTANTE	Nº DE PEDIDO	FECHA DE ENVÍO	MODO DE ENVÍO	MODO DE PAGO	OTROS TÉRMINOS
[]	[]	[]	[]	[]	[]

Fig. 39.2
Ejemplo de un formulario de factura.

Podrá utilizar formularios de varias maneras a fin de ahorrar tiempo, esfuerzo y dinero. Puede crear sus propios formularios de negocios de uso corriente, como facturas de venta, hojas de pedido, fichas de personal, calendarios y contratos normalizados. Asimismo puede imprimir una copia de su formulario en blanco y reproducirlo en grandes cantidades, empleando color si lo desea. Después, imprima únicamente la información contenida en su formulario en los formularios preimpresos y la información se colocará correctamente.

También puede automatizar los formularios que no necesita imprimir. Distribuya al personal de su compañía plantillas de formularios, en lugar de

Fig. 39.3
Otro ejemplo de
formulario.

Microsoft Word - Alquila

Fecha: 23 Jun 1993

7 de junio de 1993

Nombre de la compañía:
Dirección:
Código postal, Ciudad (País):

Estimado Sr/Sra:

Como ya sabe el propietario de la oficina nº 3000 que ocupa su compañía en el edificio 3000 espera el 30/06. Abiendo lo abra un contrato para prorrogar la ocupación de la oficina durante otros 3 años.

El contrato incluye un alquiler mensual de 3000. Este alquiler incluye los gastos correspondientes al inmueble.

Le ruego por favor antes de aceptar. Para aceptar un nuevo propietario le agradeceré que me envíe antes de esta a nueva oferta a través de correo a través de la fecha de expiración del arrendamiento. La devolución una copia firmada para sus archivos.

formularios en papel. Puede simplificar el uso de los formularios incluyendo mensajes útiles en pantalla y macros automáticas.

Puede incluir cálculos en los formularios (por ejemplo, sumar los precios de los ítems de una factura de ventas, de modo que se muestre el importe total de la factura). Asimismo podrá agregar campos de formulario a documentos que no sean formularios. Cuando los campos aparecen sombreados, la gente reconoce fácilmente dónde debe introducir la información necesaria.

Los formularios más útiles están basados en plantillas, que se pueden usar una y otra vez. Cuando uno rellena un formulario de este tipo, lo que hace es rellena una copia, con lo que el original no cambia. (Obviamente puede crear un formulario como documento en vez de como plantilla si va a utilizarlo sólo una vez). Cuando alguien crea un nuevo documento basado en una plantilla de formulario hecha por ud., esa persona podrá escribir la información únicamente en los campos designados por ud. cuando creó y protegió el formulario.

Atención

A no ser que añada protección con una contraseña, cualquiera que use el formulario puede desprotegerlo y realizar cambios no sólo en los campos, sino también en otras partes del formulario. Más adelante • este capítulo, aprenderá a proporcionar la máxima protección a las plantillas de formulario.

Puede crear formularios usando dos herramientas importantes:

- **Insertar. Campo de Formulario.** que se usa para insertar y personalizar campos de formulario
- La barra de herramientas Formulario, que contiene herramientas para crear y personalizar formularios (véase la figura 39.4).



Puede desplegar la barra de herramientas Formulario eligiendo **Insertar, Campo de Formulario** y eligiendo **Mostrar Barra** o eligiendo **Ver, Barras de Herramientas** y seleccionando Formulario de la lista **Barras de Herramientas**. Como alternativa, puede pulsar con el botón derecho del ratón en la barra de herramientas Estándar y seleccionar Formulario de la lista desplegable de barras de herramientas.

Fig. 39.4
La barra de herramientas Formulario incluye herramientas para ayudarle a crear y personalizar formularios.

Cómo crear formularios

El proceso de creación de un formulario simple consta de tres partes. Primero: deberá crear una plantilla nueva; construir la *estructura del formulario* (el marco del formulario); añadir rótulos, formato, sombreado, bordes y cualquier otro elemento que no se modifique cuando los usuarios rellenen el formulario. Después, inserte los campos de formulario donde desee escribir la información al rellenar el formulario. Por último, proteja y guarde el formulario.

Cómo crear y guardar la estructura del formulario

Otra forma de crear un formulario es utilizando marcos. Cuando enmarca una tabla o texto seleccionado, puede colocarlos en cualquier lugar de la página que desee (véase la figura 39.5). De esta manera, puede separar la porción de un documento que contiene los campos en los que debe insertar información proveniente de otras partes del documento donde es posible que el texto no se modifique.

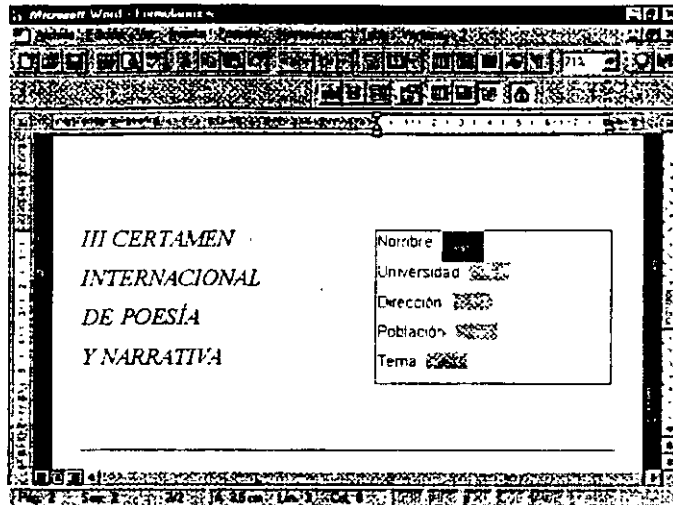
Nota

- Antes de empezar a diseñar un formulario en su ordenador, es aconsejable hacer un borrador en papel, especialmente si usa una tabla como estructura para el formulario. Gracias al borrador, sabrá cuántas filas y columnas necesita en su tabla, dónde escribir los rótulos y dónde insertar los campos de formulario. Incluso si cambia el formulario en el proceso, le resultará más fácil empezar con un plan.

Para crear y guardar la estructura del formulario como plantilla, siga estos pasos:

1. Elija **Archivo, Nuevo**. Aparecerá el cuadro de diálogo Nuevo.
2. En la lista **Plantilla**, seleccione la plantilla que desea usar como base para su formulario. En la mayoría de los casos, puede usar la plantilla Normal.
3. Seleccione **Plantilla** del grupo Nuevo.
4. Elija [Aceptar].
5. Establezca la estructura del formulario de alguna de las siguientes maneras:

Fig. 39.5
Enmarcando texto seleccionado que incluya campos de formulario, puede crear un formulario como éste.



Consejo

Si crea un nuevo formulario o como documento podrá seguir guardándolo como una plantilla. Guárdelo otra vez usando Archivo, Guardar Como. Seleccione Plantilla de Documento (*.DOT) en la lista desplegable Guardar como.

Inserte una tabla eligiendo **Tabla, Insertar Tabla**, o realizando una pulsación en el botón [Insertar tabla] de la barra de herramientas Estándar. Escriba los rótulos y cualquier otro texto que no vaya a cambiar en el formulario. Formatee la tabla con líneas, bordes y sombreado.

Cree un formulario basado en párrafos insertando campos de formulario donde los necesite a medida que escribe texto en su documento. Lea el siguiente apartado, "Cómo agregar campos de formulario", para aprender a insertar campos de formulario.

En la parte superior de su documento, inserte la tabla o tipo de texto que va a contener los campos de formulario. Seleccione, enmarque y coloque esta porción de su documento. Después escriba el resto del formulario, que incluirá el texto que no cambia cuando se rellena el formulario.

6. Elija **Archivo, Guardar** como para guardar la plantilla. Escriba el nombre de la plantilla en el cuadro Guardar como, y después elija [Aceptar]. Deje la plantilla abierta para poder agregar los campos de formulario.

Las plantillas normalmente se guardan en la carpeta PLANTILLA situada bajo la carpeta que contiene a Word. Si se instala Microsoft Office, las plantillas se almacenan en la carpeta MSOFFICEPLANTILLA.

Cómo agregar campos de formulario

Consejo

Si está creando un formulario que contiene muchos campos de formulario iguales ahorrará tiempo copiando un campo de formulario existente y pegándolo en una nueva ubicación.

Después de establecer la estructura de su formulario (tanto si es una tabla, un bloque de texto enmarcado o un párrafo), puede agregar los campos de formulario. Los campos de formulario permiten al usuario introducir datos. Como se ha mencionado anteriormente, los tres tipos de campos de formulario son texto, casilla de verificación y lista desplegable. Puede agregar campos de formulario a su plantilla usando un comando de menú o realizando una pulsación en los botones de la barra de herramientas Formulario. Para agregar campos de formulario a su documento usando un comando de menú, siga estos pasos:

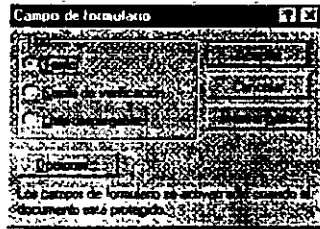


Fig. 39.6 El cuadro de diálogo Campo de Formulario le permite elegir el tipo de campo.

1. Coloque el punto de inserción donde desea que aparezca el campo de formulario.
2. Elija **Insertar, Campo de Formulario**. Aparecerá el cuadro de diálogo Campo de Formulario (véase la figura 39.6).
3. Seleccione **Texto**, **Casilla de verificación**, o **Lista desplegable** del grupo Tipo.
4. Elija [Aceptar]. El campo de formulario aparecerá en su documento (véase la figura 39.7).

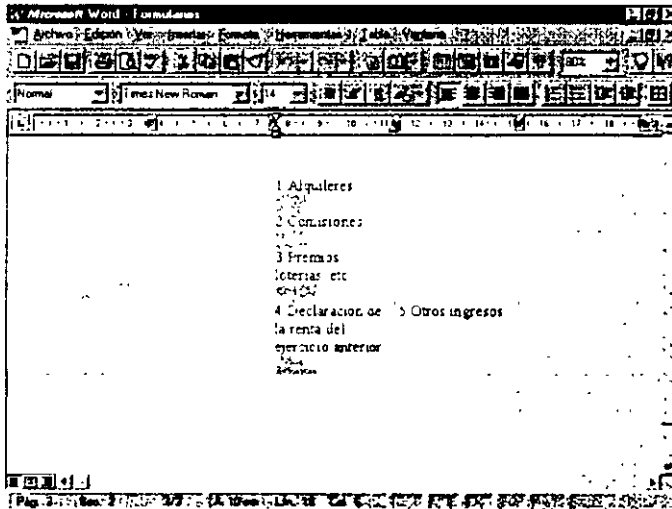


Fig. 39.7 Un campo de formulario aparece en su documento.

Para agregar campos de formulario a su documento usando la barra de herramientas Formulario, siga estos pasos:

1. Despliegue la barra de herramientas Formulario eligiendo **Insertar, Campo de formulario**, y realizando una pulsación en el botón **Mostrar barra**. Aparecerá la barra de herramientas Formulario (véase la figura 39.8).

Como alternativa, use otra técnica para desplegar la barra de herramientas Formulario.

◀ VÉASE "Cómo utilizar las barras de herramientas en aplicaciones de Microsoft Office", pág. 41

Fig. 39.8

Puede usar la barra de herramientas Formulario para simplificar la creación de un formulario.



2. Coloque el punto de inserción donde desee agregar un campo de formulario.

3. Pulse en una de las herramientas que aparecen a la izquierda de la barra de herramientas Formulario:

Para insertar un campo de texto, pulse en el botón [Campo con texto].

Para insertar una casilla de verificación, pulse en el botón [Campo con casilla de verificación].

Para insertar una lista desplegable, pulse en el botón [Campo con lista desplegable]. (Observe que una lista desplegable está vacía hasta que la personaliza agregándole ítems: véase el apartado "Cómo personalizar campos de formulario con lista desplegable", más adelante.)

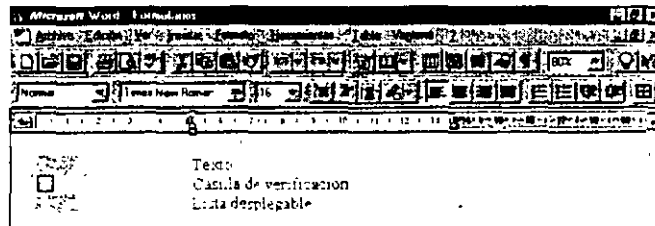
Aparecerá un campo de formulario en su documento.

4. Repita los pasos 2 y 3 para agregar campos de formulario a su documento.



Fig. 39.9

Los campos de formulario aparecen sombreados cuando pulsa en el botón de opción Sombreado de campo en la barra de herramientas Formulario.

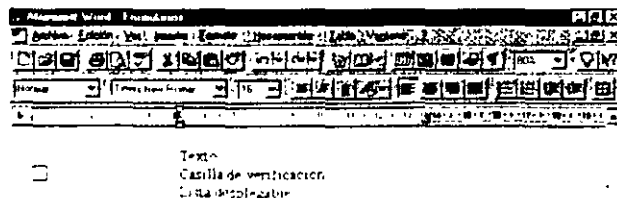


Obsérvese que la barra de herramientas Formulario contiene un botón con la opción Sombreado de campo. Si selecciona esta herramienta, los campos de formulario aparecerán como rectángulos sombreados en su pantalla (véase la figura 39.9). Si no selecciona esta herramienta, los campos con texto aparecerán sin sombreado o borde, los campos con casillas de verificación aparecerán con un contorno cuadrado, y los campos con lista desplegable aparecerán con un contorno rectangular (véase la figura 39.10).

Los pasos anteriores insertan cuadros de texto, casillas de verificación o listas, pero no podrá usar estos campos de formulario hasta que proteja su documento o plantilla, tal y como se describe en el siguiente apartado.

Fig. 39.10

Si no pulsa en el botón Sombreado de campo, los campos de formulario con texto no aparecerán en absoluto, mientras que los campos con casilla de verificación y lista desplegable se verán enmarcados.



Nota

Consulte el apartado "Cómo personalizar campos de formulario" más adelante en este capítulo para aprender a personalizar cada campo de formulario que agrega a su documento. Por ejemplo, puede personalizar un campo con texto de modo que la fecha del día aparezca automáticamente o personalizar un campo con lista desplegable para agregarle ítems.

Cómo proteger y guardar el formulario

Hasta que no proteja un documento que contiene campos de formulario, podrá editar cualquiera de sus partes, campos con texto o los campos de formulario. Una vez que el documento está protegido, podrá rellenar un campo de formulario, pero ya no podrá editar el documento.

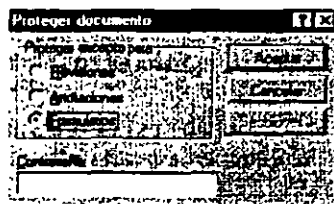
Un formulario protegido difiere de un formulario no protegido en varios aspectos. Por ejemplo, Cuando un documento protegido está en el modo de ver Diseño de página, no puede editar el documento, sólo podrá insertar una respuesta en un campo de formulario. No podrá seleccionar el documento entero, ni usar la mayor parte de los comandos, incluidos los comandos de formato. Las tablas y los marcos están fijos, y los campos con fórmulas reflejarán resultados en vez de las fórmulas.

Podrá desproteger un documento fácilmente cuando desee editarlo, a no ser que alguien lo haya protegido con una contraseña. Para aprender a usar la protección con contraseña, consulte el apartado "Cómo proteger y desproteger un formulario con una contraseña", más adelante. Para aprender a proteger sólo una parte del formulario, véase más adelante el apartado "Cómo proteger parte de un formulario".

Siempre y cuando designe como plantilla su nuevo documento, Word automáticamente lo guardará como plantilla (usando la extensión DOT) y pondrá guardarlo en la subcarpeta PLANTILLA, en donde *deberá* permanecer para que Word sea capaz de encontrarlo cuando cree un nuevo documento. Para utilizar su formulario como plantilla que aparece al elegir **Edición, Nuevo**, no cambie ni la carpeta ni la extensión de archivo. (podrá, sin embargo, especificar que desea almacenar todas las plantillas en una subcarpeta diferente eligiendo Herramientas, Opciones, seleccionando la ficha Archivos, y modificando las Plantillas personales.)

Para proteger y guardar su formulario, siga estos pasos:

1. Elija **Herramientas, Proteger documento**. Aparecerá el cuadro de diálogo Proteger documento (véase la figura 39.11).
2. Seleccione **Formularios**, y elija [Aceptar].

**Consejo**

Aunque no desee tener campos de formulario sombreados en su formulario en pantalla, use el sombreado mientras crea su formulario para facilitar la edición y visualización de los campos.

Fig. 39.11
Después de proteger un documento, no podrá editarlo.

Consejo

Alguien puede haber cambiado los nombres de su carpeta MSOFFICE y de su subcarpeta PLANTILLAS, acepte estos nombres cambiados si aparecen como nombres por defecto.

Fig. 39.12
Asegúrese de guardar su formulario como plantilla.

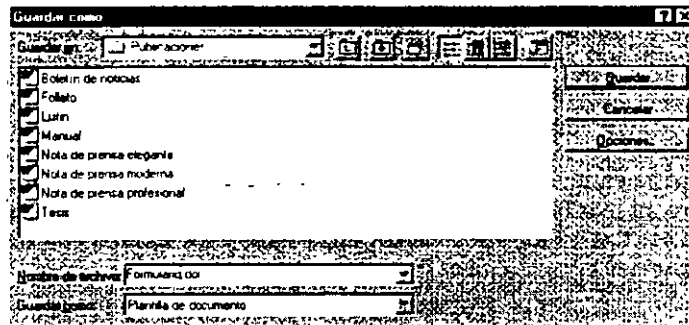
**Consejo**

Cuando da a alguien un formulario en pantalla para que lo use, asegúrese de que le entregue la plantilla. No olvide advertir a la persona que la copie en la subcarpeta PLANTILLAS bajo la carpeta MSOFFICE.

3. Elija **Archivo, Guardar como** (véase la figura 39.12). Escriba un nombre en el cuadro de texto **Nombre de archivo**, asegúrese de que **Plantilla de documento** está seleccionado en la lista **Guardar como** y que la subcarpeta **PLANTILLAS** y la carpeta **MSOFFICE** están seleccionadas en la lista de carpetas. La carpeta **PLANTILLAS** está ubicada bajo la carpeta **WINWORD** si no tiene Office 95.

4. Elija [Aceptar] para guardar el archivo como plantilla.

Para desproteger su formulario, elija **Herramientas, Desproteger documento**.



Si su formulario está protegido con una contraseña, deberá introducir la contraseña para desproteger el formulario. Véase la sección "Cómo proteger y desproteger un formulario con una contraseña", más adelante.

Para proteger y desproteger su formulario usando la barra de herramientas Formulario, pulse en el botón [Proteger formulario] de la barra de herramientas Formulario. Si el botón aparece presionado, el formulario está protegido; si aparece en relieve, el formulario está desprotegido.

Word tiene dos maneras de guardar formularios. Puede guardar el formulario completo, incluidos los campos, rótulos y la información introducida en el mismo. O bien puede guardar sólo la información introducida en un formulario, de modo que pueda usar estos datos con otro programa. Véase el apartado "Cómo guardar formularios en pantalla" más adelante en este capítulo, para detalles acerca del segundo método.

Cómo utilizar un formulario en pantalla

La gran ventaja que ofrecen los formularios es que en vez de editarlos, abre una copia vacía del formulario (conservando intacto el original) y después se desplaza de un campo a otro, introduciendo la información donde corresponda.

Los tres tipos de campos de formularios (texto, casilla de verificación y lista desplegable) no sólo presentan un aspecto distinto, sino que además cada

Realizando una pulsación en esta flecha verá los ítems en una lista desplegable

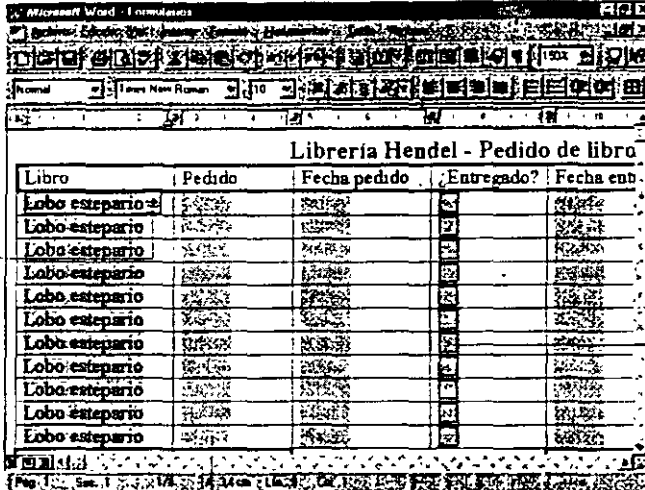


Fig. 39.13 Pulse en la flecha o pulse <Alt>+flecha hacia abajo para ver los ítems en una lista desplegable.

Seleccione o despeje una casilla de verificación pulsando la barra espaciadora

Responda a un campo de texto escribiendo texto o un número

uno requiere que utilice un enfoque específico (véase la figura 39.13)

Puede personalizar su formulario de varias maneras. Por ejemplo, puede personalizar un campo de texto de modo que contenga sólo fechas y que formatee la fecha que introduce de una determinada forma. Cualquier tipo de campo puede llevar adjunto un mensaje de ayuda de modo que cuando entre en el campo o pulse <F1> las instrucciones de uso del campo aparezcan en la barra de estado. En algunos campos, alguna respuesta particular puede producir algún tipo de acción en otra parte del formulario: por ejemplo, una respuesta positiva a un campo con casilla de verificación puede activar otro campo posterior en el formulario. Manténgase alerta sobre lo que ocurre en la pantalla mientras rellena su formulario. Para abrir un formulario, siga estos pasos:

1. Elija **Archivo, Nuevo**. Aparecerá el cuadro de diálogo Nuevo (véase la figura 39.14)
2. En la lista **Plantilla**, seleccione el nombre de su formulario.
3. Elija [Aceptar]. Una copia sin título del formulario aparecerá en la pantalla, con el primer campo del formulario seleccionado.

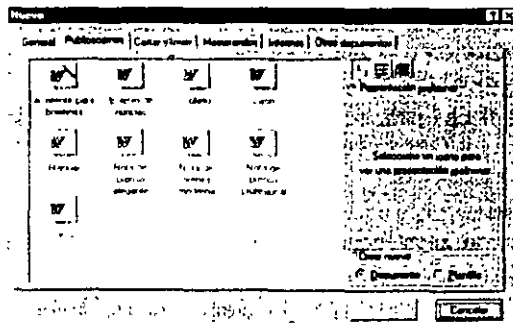


Fig. 39.14 Abra un formulario eligiendo Archivo, Nuevo.

◀ VÉASE "Cómo guardar, abrir y cerrar archivos", pág. 102

Aún en el caso de que su formulario no está basado en una plantilla, podrá usarlo abriéndolo como un archivo normal. Elija **Archivo, Nuevo**, localice y seleccione su formulario y elija [Aceptar]. Guárdelo con un nuevo nombre para conservar el original.

Cómo rellenar un formulario en pantalla

Cuando abre un formulario nuevo y protegido, el primer campo aparece seleccionado (destacado).

Para rellenar los campos en un formulario, siga estos pasos:

1. Responda al campo seleccionado según convenga.

En un campo con texto, escriba el texto o número requerido.

Realizando una pulsación en las casillas de verificación, éstas se activan o desactivan. Pulse una vez la barra espaciadora para marcar una casilla de verificación vacía; pulse la barra espaciadora una segunda vez para quitar la marca de la casilla. (Las casillas de verificación pueden estar marcadas por defecto; si este es el caso, pulse una vez la barra espaciadora para quitar la marca, y púlsela por segunda vez para volver a poner la marca).

En un campo con lista desplegable, pulse en la flecha para desplegar una lista de selecciones, y después pulse en el ítem que desee seleccionar. O bien, con el teclado, pulse <Alt>+flecha hacia abajo para desplegar la lista y pulse las teclas de flecha hacia arriba o hacia abajo para seleccionar un ítem de la lista.

2. Pulse <Tab> o <Intro> para aceptar su entrada. Pasará una de dos cosas:

Si su entrada es aceptable, Word seleccionará el siguiente campo.

Si su entrada es inaceptable, Word desplegará un mensaje de error y regresará al campo actual para que pueda hacer una entrada correcta. Por ejemplo, es posible que se encuentre con un mensaje de error, cuando escriba texto en un campo con texto que está formateado para contener números.

3. Continúe rellenando los campos hasta completar el formulario.

Si comete un error y desea volver al campo anterior, mantenga presionada la tecla <Mayús> mientras pulsa la tecla <Tab> o <Intro> hasta que se encuentre en ese campo. Para desplazarse al campo siguiente sin hacer una entrada, sencillamente pulse <Tab> o <Intro>. También puede moverse de un campo a otro pulsando las teclas de flecha hacia arriba o hacia abajo. Igualmente podrá desplazarse al inicio o al final de su formulario pulsando <Control>+<Inicio> o <Control>+<Fin>.

Para editar una entrada en un campo que ya ha dejado atrás, use el ratón o las teclas de flecha para colocar el punto de inserción junto al texto que desea editar; pulse <Retroceso> para borrar caracteres, o escriba los caracteres que desea insertar.

Si desea insertar un carácter tabulador en un campo sin desplazarse al siguiente campo, mantenga presionada la tecla <Control> mientras, pulsa <Tab>.

Consejo

Podrá usar las teclas de flecha hacia arriba y hacia abajo para moverse entre campos en un formulario, excepto en caso de que una lista desplegable este abierta.

Consejo

Busque en la barra de estado mensajes que le puedan ayudar a rellenar cada campo en un formulario. También podrá pulsar <F1> para obtener ayuda para un campo particular cuando ese campo este seleccionado.

Preste atención a la manera en que Word interpreta su respuesta a un campo. Por ejemplo, si un campo de texto está formateado para incluir números formateados sin decimales y responde deletreando un número (cuatro, por ejemplo), Word interpretará su respuesta como 0 (cero) porque lo que espera es un número y no letras. Regrese al campo y escriba la respuesta correcta.

Si el formulario no está protegido, el primer campo no estará seleccionado cuando abra el formulario, y no se podrá desplazar de un campo a otro pulsando <Tab> o <Intro>. Para rellenar un formulario desprotegido, utilice las técnicas usuales de Word para mover el punto de inserción de un campo a otro. O mejor aún, proteja el formulario eligiendo **Herramientas, Proteger documento** y seleccione la opción **Formulario**.

◀ VÉASE "Cómo usar los asistentes para plantillas", pág. 245

◀ VÉASE "Cómo trabajar con tablas", pág. 261

Cómo guardar un formulario en pantalla

Dado que la mayoría de los formularios en pantalla están basados en plantillas, no tienen nombres cuando los abre. Deberá guardar y dar un nombre a los formularios. (Si un formulario no está basado en una plantilla, use los siguientes pasos para guardar su formulario con un nombre único. De esta manera, conservará el original para usos futuros.)

Para guardar un formulario, siga estos pasos:

1. Elija **Archivo, Guardar como**. Aparecerá el cuadro de diálogo Guardar como.
2. Escriba un nombre en el cuadro **Nombre de archivo**, y seleccione la carpeta para el formulario de la lista de carpetas. Puede guardar el formulario en una unidad de disco diferente seleccionándola de la lista **Guardar en**.
3. Elija [Aceptar].

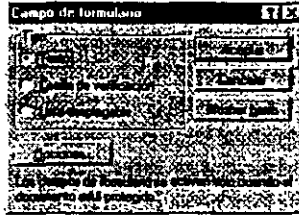
SOLUCIÓN DE PROBLEMAS

*He abierto un formulario nuevo y el primer campo no está seleccionado. El formulario no está protegido. Para rellenarlo, protéjalo eligiendo **Herramientas, Proteger documento**. (Es aconsejable abrir la plantilla de formulario y protegerla, para que la próxima vez que abra el formulario esté protegido.)*

Cómo personalizar campos de formulario

Hay muchas maneras de personalizar los campos de formulario para que sus formularios resulten más informativos, automatizados y fáciles de usar. Los campos de fecha y hora automáticos, por ejemplo, insertan la fecha u hora actual en su formulario. Las entradas por defecto sugieren una respuesta posible al campo. Los mensajes de ayuda dan pistas a los usuarios de cómo rellenar un campo determinado. Los controles impiden ciertos tipos de errores. Las fórmulas calculan los resultados en un campo. Las macros se ejecutan cuando los usuarios entran o salen de un campo particular. Los controles impiden cierto tipo de errores.

Fig. 39.15
Elija **Insertar, Campo de formulario** para personalizar campos de formulario a medida que los inserta.



También puede aplicar la mayoría de los tipos de formato a los campos de formulario. Por ejemplo, puede convertir en negrita un campo de formulario para que la respuesta se destaque. O bien, puede aplicar un borde a un campo de formulario para agregarle cuadros a un formulario que no esté basado en una tabla.

Fig. 39.16
Si selecciona **Texto** del grupo Tipo y después pulsa en el botón **Opciones**, aparecerá el cuadro de diálogo **Campo de formulario con texto**.



Puede personalizar campos de formulario mientras crea su formulario, personalizando campos a medida que los inserta, o después de haber creado su formulario, editando campos seleccionados. Para editar campos de formulario después de haberlos insertado, el documento deberá estar desprotegido. Para personalizar campos de formulario a medida que los inserta, siga estos pasos:



1. Elija **Insertar, Campo de formulario**. Aparecerá el cuadro de diálogo **Campo de formulario** (véase la figura 39.15).
2. Del grupo **Tipo** (**Texto**, **Casilla de verificación**, o **lista desplegable**), elija el tipo de campo de formulario que desea insertar y personalizar.
3. Pulse en el botón **Opciones**. Aparecerá un cuadro de diálogo conteniendo las opciones para personalizar el tipo de campo de formulario que haya seleccionado (véase la figura 39.16).

Para personalizar un campo de formulario existente, siga estos pasos:

1. Desproteja su documento, si está protegido, eligiendo **Herramientas, Desproteger documento** o realizando una pulsación en el botón [Proteger formulario] de la barra de herramientas **Formulario**.
2. Realice una doble pulsación en el campo de formulario que desea personalizar, abriendo el cuadro de diálogo **Campo de formulario con texto**. O bien, seleccione el campo de formulario que desea personalizar realizando una pulsación en él, colocando el punto de inserción por encima

o por debajo del mismo y pulsando la tecla de flecha hacia arriba o hacia abajo, o colocando el punto de inserción junto al mismo y manteniendo presionada la tecla <Mayús> mientras pulsa la tecla de flecha hacia la izquierda o hacia la derecha. Después, haga alguna de estas cosas:

Pulse en el botón [Opciones] de la barra de herramientas Formulario.

O bien, pulse en el campo de formulario que desea personalizar con el botón derecho del ratón para desplegar el menú contextual. Seleccione Opciones de campo de formulario.

3. Seleccione las opciones que desee, y elija [Aceptar].

Consejo

Si usa un campo de formulario repetidamente en un formulario, duplíquelo emplazándolo en Autotexto



Cómo personalizar campos de formulario con texto

Los campos con texto son probablemente los más personalizables de los tres tipos de campos de formulario. Puede personalizarlos por tipo (normal, número, fecha o cálculo, por ejemplo), por texto por defecto, por el tamaño del campo, por el número máximo de caracteres en la respuesta, o por el formato de la respuesta. Al igual que sucede con todos los tipos de campos de formularios, también podrá personalizar campos de texto agregando macros (véase más adelante el apartado "Cómo agregar macros a un formulario"), añadiendo texto de ayuda (véase el apartado "Cómo agregar Ayuda a un formulario"), asignando un nuevo nombre al marcador (véase "Cómo asignar nombre y encontrar campos en un formulario"), o inutilizando el campo para entradas (véase "Cómo inutilizar campos de formulario").

Para especificar restricciones en un campo de formulario con texto, siga estos pasos:

◀ VÉASE "Cómo usar Autotexto", pág. 198

Consejo

Especificar el tamaño de un campo es particularmente importante cuando se están usando formularios preimpresos.



Fig. 39.17 Los campos de formulario con texto presentan numerosas opciones de personalización.

1. Abra el cuadro de diálogo Campo de formulario con texto (véase la figura 39.17).
2. Seleccione de entre los siguientes tipos de opciones (véase las siguientes tablas para detalles sobre las opciones Tipo y Formato).

OPCION	DESCRIPCION
Tipo (véase la tabla 39.1)	Seleccione de entre seis tipos de entradas de texto: Texto normal, Número, Fecha, Fecha actual, Hora actual y Cálculo.
Texto predeterminado	Escriba el texto que desea que aparezca como entrada por defecto en este campo. Los usuarios pueden cambiar la entrada.
Largo máximo	Escriba o seleccione "ilimitado" o el número de caracteres o números que desea que acepte el campo (hasta un total de 255).
Formato del texto	Seleccione de entre diferentes tipos de formato de texto, numérico y de fecha, dependiendo de lo que haya seleccionado en la opción Tipo (véase la tabla 39.2).

3. Elija [Aceptar].

A menudo usará dos o más de estas opciones juntas. Por ejemplo, si selecciona Número como el **Tipo**, es posible que elija 0,00 como el **Formato** a fin de que aparezca una respuesta numérica con dos decimales.

Tabla 39.1. Opciones Tipo para el cuadro de diálogo Campo de Formulario con Texto

SELECCIONE ESTA OPCION:	CUANDO LOS USUARIOS DEBAN RESPONDER ESCRIBIENDO:
Texto Normal	Texto. Word formateará el texto de acuerdo con su selección en la lista Formato del texto.
Número	Un número. Word formateará el número de acuerdo con su selección en la lista Formato del número, y desplegará un mensaje de error si el usuario escribe texto.
Fecha	Una fecha. Word formateará la fecha de acuerdo con su selección en la lista Formato de fecha. Word desplegará un mensaje de error (Se requiere una fecha válida) si el usuario escribe texto o un número no reconocible como fecha y el usuario regresará al campo actual para buscar una respuesta apropiada. (No obstante, servirá casi cualquier respuesta parecida a una fecha).
Fecha actual	No se permite una respuesta por parte del usuario. Word introducirá la fecha actual (y actualizará la fecha al abrirse el documento*).

Hora actual	No se permite una respuesta por parte del usuario. Word introducirá la hora actual (y actualizará la hora al abrirse el documento*).
Cálculo	Introduzca una fórmula cuando inserte o edite este campo; no se permite una respuesta por parte del usuario. Word aplicará su fórmula, e imprimirá el resultado del cálculo en este campo. Por ejemplo, puede insertar una simple fórmula de SUMA para sumar los números de una columna si su formulario se haña en una tabla (Word actualizará el resultado al abrirse el documento*)

* puede especificar que Word actualice la fecha, hora o una fórmula cuando imprima su formulario, eligiendo el comando Herramientas, Opciones, seleccionando a continuación la ficha Imprimir y después Actualizar campos en las Opciones de impresión. También puede usar una macro de salida para actualizar los campos.

Tabla 39.2. Opciones formato del texto del cuadro de diálogo Campo de formulario con texto

TIPO DE OPCION DEL TEXTO	OPCION FORMATO	ASPECTO DE LA ENTRADA
Texto Normal Minúsculas	Mayúsculas	TODO MAYUSCULAS todo minúsculas
	Primera Mayúscula	La primera letra de la primera palabra es mayúscula
	Caja de Título	La Primera Letra De Cada Palabra es Mayúscula
Número	()	123456
	0,00	123456.00
	##0	123.456
	##0,00	123.456,00
	###0.00;(\$###0.00)	\$123.456,00
	0%	10%
	0,00%	10,00%
Fecha	M/d/aa	1/1/95
	dddd, MMMM d, aaa	Domingo, Enero 3, 1995
	d MMMM, aaaa	3 Enero, 1995
	d-MMM-aa	3-Ene-95
	MMM, aa	Ene, 95
	MM/dd/yy h:mm	01/03/95 2:15 PM
	AM/PM	
	MM/dd/yy h:mm:ss AM/PM	01/03/95 2:15:58 PM

◀ VÉASE "Trabajando con tablas", pág. 286

Hora	h:mm AM/PM	2:15 PM
	h:mm:ss AM/PM	2:15:58 PM
	H:mm	2:15
	H:mm:ss	2:15:58
Fecha actual	Lo mismo que Fecha	Lo mismo que Fecha
Hora actual	h:mm AM/PM	3:30 PM
	h:mm:ss AM/PM	3:30:00 PM
	H:mm	15:30
	H:mm:ss	15:30:00
Cálculo	Lo mismo que Número	Lo mismo que Número

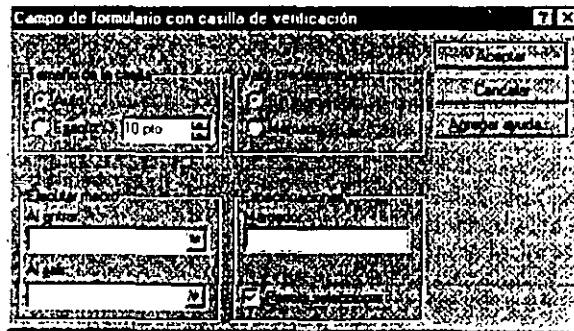
Cómo personalizar campos de formulario con Casillas de verificación

Podrá personalizar los campos con casilla de verificación, que exigen del usuario que una respuesta simple de "sí o no", determinando el tamaño y eligiendo si estarán marcadas o no por defecto. Al igual que con todos los otros tipos de campos, también puede personalizar los campos con casilla de verificación agregando macros (véase más adelante el apartado "Cómo agregar macros a un formulario"), agregando texto de ayuda (véase el apartado "Cómo agregar Ayuda a un formulario"), asignando un nuevo nombre al marcador (véase "Cómo asignar nombre y encontrar campos en un formulario"), o inutilizando el campo para entradas (véase "Cómo inutilizar campos de formulario").

Para personalizar un campo con casilla de verificación, siga estos pasos:

1. Abra el cuadro de diálogo Campo de formulario con casilla de verificación (véase la figura 39.18).
2. Determine el tamaño de la casilla de verificación seleccionando la opción apropiada:
 - Seleccione **Auto** para ajustar el tamaño de la casilla de verificación al tamaño del texto que la rodea.
 - Seleccione **Exacto** para darle a la casilla de verificación un tamaño específico. Pulse en la flecha hacia arriba o hacia abajo o pulse la tecla de flecha hacia arriba o hacia abajo para aumentar o disminuir

Fig. 39.18
Puede dar a una casilla de verificación el tamaño exacto que desee, y puede especificar si va marcada o no por defecto.



el tamaño de la casilla. O escriba el tamaño que desee; por ejemplo, escriba **12p** para una casilla de 12 puntos, **0,25"** para una casilla de cuarto de pulgada, **1 pi** para una casilla de 1 pica, o **1 cm** para una casilla de 1 centímetro. (Cuando abra el cuadro de diálogo la próxima vez, la medida se convertirá en un valor equivalente en puntos.)

3. Determine el Valor predeterminado seleccionando una de las siguientes opciones:

Si selecciona **Sin marca**, la casilla de verificación aparecerá vacía por defecto (una respuesta negativa). El usuario deberá pulsar la barra espaciadora para marcar la casilla.

Si selecciona **Marcado**, la casilla de verificación presentará una marca por defecto (una respuesta positiva). El usuario deberá pulsar la barra espaciadora para desactivar la selección de la casilla.

4. Elija [Aceptar].

Cómo personalizar campos de formulario con lista desplegable

Una lista desplegable ofrece a los usuarios 25 ítems entre los que elegir. Dado que la lista contiene únicamente respuestas válidas, ayuda a asegurar que la respuesta del usuario a un campo es válida. Además, ayuda a los usuarios a rellenar el formulario, porque no tienen que adivinar qué tipo de respuesta requiere el campo.

Con toda probabilidad, personalizará un campo de formulario con lista desplegable al insertarlo, ya que la lista no contiene nada hasta que le añade los ítems. Sin embargo, es posible que, posteriormente, desee agregar ítems a la lista, eliminar algunos o reordenarlos. Puede hacer esto editando el campo desplegable.

Para agregar ítems a la lista en un campo desplegable, siga estos pasos:

1. Abra el cuadro de diálogo Campo de formulario con lista desplegable (véase la figura 39.19).
2. En el cuadro **Elemento** de la lista, escriba el ítem que desea añadir a la lista.
3. Pulse en el botón [Agregar].

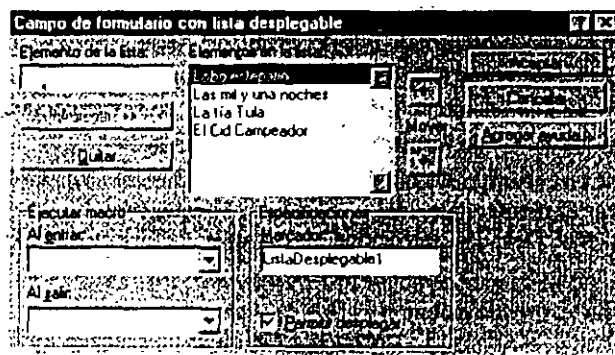


Fig. 39.19
Puede agregar ítems a una lista desplegable, eliminarlos o reordenarlos.

4. Repita los pasos 2 y 3 para agregar más ítems a la lista.

5. Elija [Aceptar].

Para quitar ítems de un campo con lista desplegable, siga estos pasos:

1. Seleccione el campo desplegable y abra el cuadro de diálogo Campo de formulario con lista desplegable.

2. Seleccione el ítem que desea quitar de la lista **Elementos en la Lista**.

3. Pulse en el botón **Quitar**.

4. Repita los pasos 2 y 3 para quitar más ítems.

5. Elija [Aceptar].

Para reordenar los ítems en un campo con lista desplegable, siga estos pasos:

1. Seleccione el campo desplegable y abra el cuadro de diálogo Campo de formulario con lista desplegable.

2. Seleccione el ítem que desea desplazar en la lista **Elemento de la Lista**.

3. Desplace el ítem hacia arriba realizando una pulsación en la flecha Mover hacia arriba, o desplácelo hacia abajo realizando una pulsación en la flecha Mover hacia abajo. (Con el teclado, pulse la flecha hacia arriba o hacia abajo para seleccionar el ítem que desea desplazar, pulse <Tab> para seleccionar la flecha Mover hacia arriba o Mover hacia abajo y después pulse la barra espaciadora para desplazar el ítem seleccionado hacia arriba o hacia abajo.)

4. Repita los pasos 2 y 3 para desplazar más ítems.

5. Elija [Aceptar].

Cómo formatear campos de formulario

Los usuarios no pueden formatear entradas en un formulario protegido cuando están rellenando el formulario. Sin embargo, cuando está creando un formulario, puede aplicar formato de fuente y párrafo al campo, así como muchas otras opciones de formato. Las respuestas aparecerán posteriormente en ese formato.

Debe insertar un campo de formulario antes de poder formatearlo. Recuerde que el documento debe estar desprotegido.

Para formatear un campo de formulario, primero seleccione el campo de formulario que desea formatear. Después emplee uno de los siguientes métodos para aplicar el formato:

- Elija el comando formato que desea utilizar y seleccione las opciones de formato que desea aplicar.
- Pulse en una opción de formato en una barra de herramientas.
- Pulse las teclas de formato de método abreviado.
- Pulse en el campo seleccionado con el botón derecho del ratón para desplegar el menú contextual y seleccione Fuente, Párrafo, o Numeración y viñetas. Después, seleccione las opciones de formato que desea.

◀ VÉASE "Cómo trabajar con celdas, cuadros de texto y campos", pág. 68

◀ VÉASE "Cómo formatear texto", pág. 179

Cómo inutilizar campos de formulario

En la mayoría de los formularios, se pretende que los usuarios respondan a todos los campos. Pero a veces deseará inutilizar un campo con el objeto de que los usuarios no puedan responder. Es posible que desee incluir una entrada por defecto en los campos inutilizados.

Para inutilizar un campo, siga estos pasos:

1. Desproteja el documento, si es necesario.
2. Seleccione el campo que desea inutilizar, y despliegue el cuadro de diálogo Campo de Formulario.
3. Elimine la opción apropiada: **Permitir llenado** (para campos con texto), **Permitir seleccionar** (para campos con casilla de verificación), o **Permitir desplegar** (para campos con lista desplegable).
4. Elija [Aceptar].

Cómo asignar nombre y encontrar campos en un formulario

Cada uno de los campos que introduce en un documento tiene un nombre: su *marcador*. Puede usar este marcador para encontrar un campo rápidamente. Por defecto, Word numera los campos que inserta, denominándolos **Texto1**, **Casilla7**, **ListaDesplegable13**, etc. Asignar cualquier nombre a un campo de formulario (ateniéndose, sin embargo, a las reglas para asignar nombres a los marcadores).

Para asignar nombre a un campo, siga estos pasos:

1. Desproteja el documento, si es necesario.
2. Seleccione el campo, y abra el cuadro de diálogo Campo de formulario.
3. En el grupo Especificaciones, seleccione el cuadro de texto **Marcador** y escriba el nombre.
4. Elija [Aceptar.]

Para buscar un campo con nombre, siga estos pasos:

1. Desproteja el documento.
2. Elija **Edición, Marcador**.
3. Escriba el nombre que desea buscar en el cuadro **Nombre de marcador** o selecciónelo en la lista.
4. Pulse en el botón **Ir a**. Word abrirá el campo sin cerrar el cuadro de diálogo. Vaya a otro campo o elija [Cancelar] para cerrar el cuadro de diálogo.

Cómo agregar Ayuda a un formulario

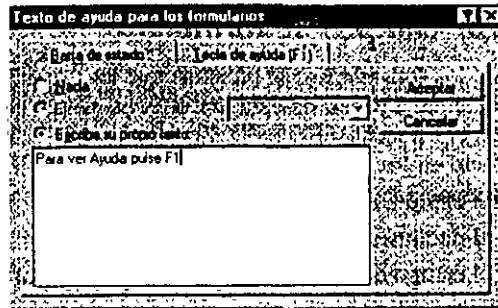
Puede facilitar a los usuarios la tarea de responder correctamente a un campo en su formulario agregando mensajes de ayuda. Cuando el campo está seleccionado en su formulario, los mensajes de ayuda pueden aparecer en la barra de estado en la parte inferior de la pantalla o como un cuadro de mensaje que se despliega cuando el usuario pulsa la tecla <F1>.

Consejo

El documento deberá estar protegido para que aparezca su mensaje de ayuda; de lo contrario, pulse <F1> para desplegar la Ayuda de Word.

Fig. 39.20

La ayuda puede aparecer en la barra de estado si usa la ficha Barra de estado.



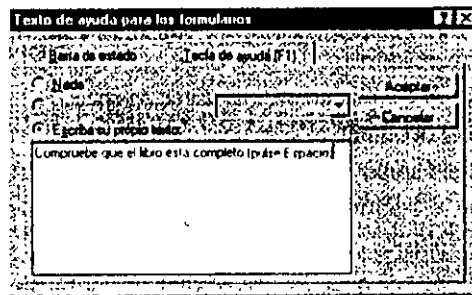
Puede escribir su propio texto para un mensaje de ayuda, o usar una entrada existente de Autotexto. Por ejemplo, puede tener una entrada de Autotexto que diga "Pulse <F1> para Ayuda" y que puede incluir como un mensaje de Ayuda de la barra de estado en cada uno de los campos para los que haya incluido ayuda de <F1> (véase la figura 39.20).

Para agregar ayuda a un campo de formulario, siga estos pasos:

1. Abra el cuadro de diálogo Campo de Formulario para el campo en el que quiera agregar la ayuda.

Fig. 39.21

La ayuda puede aparecer como un cuadro de mensaje cuando el usuario pulsa <F1> si usa la ficha Tecla de ayuda (F1).



2. Pulse en el botón **Agregar ayuda**.
3. Seleccione la ficha **Barra de estado** para agregar una línea de ayuda en la barra de estado o seleccione la ficha **Tecla de ayuda (F1)** para agregar la ayuda que haya de aparecer en un cuadro de mensaje cuando el usuario pulse <F1> (véase la figura 39.21).

4. Para agregar su propio mensaje de ayuda, seleccione la opción **Escriba su propio texto** y escriba su mensaje (hasta 255 caracteres).

O bien, para usar el texto de una entrada de **Autotexto** como ayuda, seleccione la opción **Elemento de Autotexto**, y seleccione en la lista la entrada de Autotexto que desea usar.

5. Elija [Aceptar].

6. Elija [Aceptar] de nuevo para cerrar el cuadro de diálogo Campo de formulario y volver a su documento.

Consejo

Si agrega ayuda <F1> a un campo, incluya también un mensaje de barra de estado que diga "Pulse F1 para Ayuda" de modo que los usuarios sepan dónde encontrar la ayuda.

Si está incluyendo mensajes de ayuda en la barra de estado, tenga presente que incluso si su formulario está protegido, los usuarios podrán desactivar esta línea de ayuda en la barra de estado. Asimismo, tenga presente que los usuarios no tienen modo de saber si la ayuda <F1> está adjunta a un campo (aunque un mensaje en la barra de estado puede ser útil, si se puede visualizar). Si su formulario está basado en una plantilla, intente incluir un mensaje (para alertar a los usuarios de la existencia de ayuda <F1>) en la macro AutoNuevo que se ejecuta cuando los usuarios crean un nuevo formulario.

Nota

Ayude a los usuarios a rellenar su formulario o déles instrucciones sobre qué hacer con el formulario cuando lo hayan terminado, incluyendo un mensaje útil como parte de una macro AutoNuevo que se activa cuando los usuarios abren el formulario. (Las macros AutoNuevo están adjuntas a plantillas y se ejecutan cuando crea un nuevo documento basado en una plantilla. Esta idea funciona de manera óptima con formularios basados en una plantilla.)

Cómo agregar macros a un formulario

Las macros pueden automatizar sus formularios de muchas maneras. Pueden activar o desactivar campos, dependiendo de la respuesta del usuario a un campo anterior. Pueden actualizar campos que contengan cálculos. Pueden hacer que Word ignore campos innecesarios.

Para usar una macro en un formulario, deberá crear la macro antes de aplicarla a un campo particular del formulario. Las macros funcionan en uno de dos momentos: cuando el usuario entra en el campo o cuando sale del mismo.

Al grabar o escribir macros para su formulario, tenga presente que las macros usan marcadores para localizar campos particulares y asegúrese de que su formulario no contiene nombres duplicados de marcador. Puede conocer el nombre automático del marcador de cada uno de los campos, o asignar al campo un nuevo nombre de marcador, seleccionando el campo, desplegando el cuadro de diálogo Campo de formulario, y mirando después el cuadro de texto Marcador (para más detalles, véase el apartado anterior "Cómo asignar nombre y encontrar campos en un formulario").

Para lograr que sus macros sean útiles, adjúntelas a la plantilla en la que se basa su formulario. Lo más fácil es hacerlo al crear la macro, pero también podrá adjuntar una macro a su plantilla eligiendo **Archivo, Plantillas**, realizando una pulsación en el botón **Organizador**, y eligiendo después la ficha Macros.

Para poder aplicar macros a un campo, en primer lugar el documento deberá estar desprotegido. Recuerde que para poder aplicar una a un campo de formulario, antes deberá crear la macro.

Para aplicar una macro a un campo de formulario, siga estos pasos:

1. Seleccione el campo al que desea aplicar la macro, y abra el cuadro de diálogo Campo de Formulario.
2. Si desea que la macro se ejecute cuando el usuario coloque el punto de inserción en el campo, seleccione la opción **Al entrar** en el grupo Ejecutar macro y seleccione la macro que desea en la lista.

O bien, si desea que la macro se ejecute cuando el usuario coloque el punto de inserción fuera del campo, seleccione la opción **Al salir** en el grupo Ejecutar macro y seleccione la macro que desea en la lista.

3. Elija [Aceptar].

Si no aparecen macros en la lista **Al entrar** o en la lista **Al salir**, eso quiere decir que no hay macros disponibles en la plantilla de su formulario. Deberá o bien crear una macro, o adjuntarla a su plantilla.

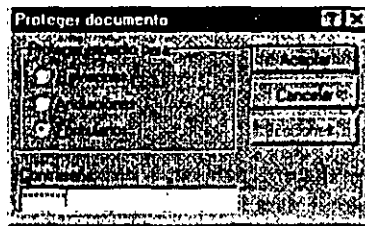
Cómo proteger y desproteger un formulario con una contraseña

Si no desea que los usuarios cambien su formulario, protéjalo con una contraseña. De esta manera, cualquiera que intente desproteger el formulario deberá introducir la contraseña (ud. incluido: no olvide su contraseña)

Para proteger un documento con contraseña, siga estos pasos:

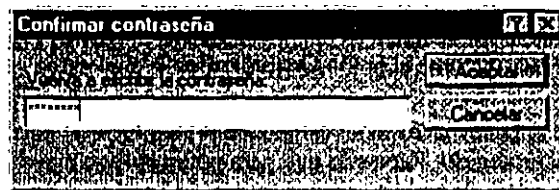
1. Elija **Herramientas, Proteger documento**. Aparecerá el cuadro de diálogo Proteger documento (véase la figura 39.22).
2. Seleccione la opción **Formularios**.

Fig. 39.22
Escriba su
contraseña en el
cuadro
Contraseña.



3. Seleccione el cuadro **Contraseña**, y escriba su contraseña. Elija [Aceptar]. Aparecerá el cuadro de diálogo Confirmar contraseña (véase la figura 39.23).

Fig. 39.23
Para confirmar
su contraseña,
deberá
reescribirla
exactamente
como lo hizo la
primera vez.



4. En el cuadro **Vuelva a escribir la contraseña**, reescriba su contraseña. El deletreo, el espaciado y la caja deben ser exactamente iguales. Elija [Aceptar]. (Si no reescribe la contraseña exactamente igual a como lo hizo originalmente, obtendrá un mensaje de error. Elija [Aceptar] e inténtelo de nuevo.)



5. Elija [Aceptar] para volver a su documento.

Para desproteger un documento con protección de contraseña, siga estos pasos:

1. Elija **Herramientas, Desproteger documento**, o pulse en el botón [Proteger formulario] de la barra de herramientas Formulario. Si el documento está protegido con contraseña, aparecerá el cuadro de diálogo Desproteger documento (véase la figura 39.24).
2. Escriba la contraseña exactamente igual a como la escribió originalmente en el cuadro **Contraseña**. El deletreo, el espaciado y la caja deben ser exactamente iguales.

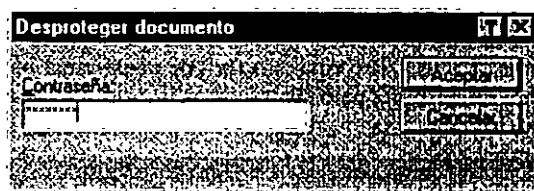


Fig. 39.24
Para desproteger un documento protegido con contraseña, deberá introducir la contraseña exactamente como la escribió originalmente.

3. Elija [Aceptar]. Si ha escrito la contraseña correctamente, su documento quedará desprotegido. Si escribió la contraseña de forma incorrecta, Word mostrará un mensaje diciendo que la contraseña es incorrecta y deberá elegir Aceptar para volver a su documento, que permanecerá protegido.

Cómo proteger parte de un formulario

Si su formulario está dividido en secciones, podrá proteger partes del mismo, al tiempo que deja otras sin protección. Podrá incluir protección de contraseña para las secciones protegidas.

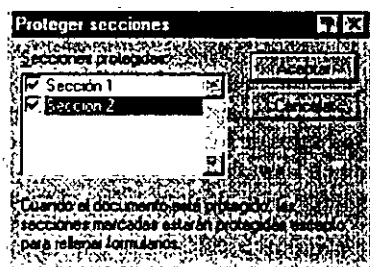


Fig. 39.25
Puede proteger parte de un formulario si éste está dividido en secciones.

Para proteger o desproteger parte de un formulario, siga estos pasos:

1. Elija **Herramientas, Proteger documento**. Aparecerá el cuadro de diálogo Proteger documento.
2. Seleccione la opción **Formulario**.
3. Pulse en el botón **Secciones** para desplegar el cuadro de diálogo Proteger secciones (véase la figura 39.25).

4. En la lista **Secciones protegidas**, seleccione cada una de las secciones que desea proteger, de modo que aparezcan con una marca en el cuadro. Quite la marca de todas aquellas secciones que no desea que estén protegidas.
5. Elija [Aceptar] para volver al cuadro de diálogo Proteger documento. (Si desea proteger las secciones con una contraseña, hágalo ahora. Para más detalles, véase el apartado anterior "Cómo proteger y desproteger un formulario con una contraseña.")
6. Elija [Aceptar] para volver al documento.

Cómo transformar formularios existentes

Los formularios que desea crear pueden existir ya en papel o en formato de otro programa. A veces, la forma más fácil de transformar un formulario es sencillamente reescribirlo, pero en otras ocasiones es posible que quiera utilizar datos existentes.

Puede emplear un escáner para facilitar la reescritura de un formulario existente. Escanee el formulario como dibujo, e insértelo en un cuadro de texto. Cuando el dibujo esté en el cuadro de texto, podrá desplazarlo a la capa detrás del texto en su documento. En la siguiente capa, podrá calcar el formulario escaneado para crear su nuevo formulario. Cuando termine, seleccione y borre el formulario escaneado.

Aunque no puede importar campo de formulario de un documento creado por otro programa a un documento Word, sí puede importar texto. Si un formulario existe en otro programa, importe el texto, formátelo convenientemente y agregue los campos de formulario. Puede convertir el texto en una tabla seleccionando el texto y eligiendo **Tabla, Convertir texto en tabla**. Para más detalles acerca de las tablas, véase el Capítulo 11, "Cómo trabajar con tablas y bordes".

Cómo imprimir un formulario

Puede imprimir un formulario de tres maneras distintas. Puede imprimir el formulario exactamente como aparece en pantalla, incluyendo los rótulos, posibles gráficos y los datos de los campos. Es posible que quiera imprimir los datos únicamente en formularios preimpresos, o bien que quiera imprimir solamente los rótulos y los gráficos, a fin de crear un formulario preimpreso.

◀ VÉASE "Cómo imprimir documentos", pág. 128

Cómo imprimir el formulario una vez completado

Para imprimir el formulario completo, incluyendo todo lo que aparece en su pantalla, utilice los comandos de impresión normales de Word. Emplee este método para imprimir formularios que ya haya rellenado.

Para imprimir el formulario completo, siga estos pasos:

1. Rellene el formulario, o abra un formulario completado.

2. Elija **Archivo, Imprimir**.

3. Seleccione las opciones de impresión que desea en el cuadro de diálogo Imprimir, y elija [Aceptar].

Para más detalles acerca de las opciones de impresión, consulte el Capítulo 8, "Cómo revisar e imprimir documentos."

Cómo imprimir solamente datos de formularios

Imprima solamente los datos cuando use formularios preimpresos. Dado que se usa la misma plantilla de formulario para imprimir el formulario en blanco que la que usa cuando se imprimen sólo los datos, los datos del formulario se alinearán correctamente con los campos.

Para imprimir solamente datos en un formulario preimpreso, siga estos pasos:

- 1. Inserte el formulario preimpreso en su impresora.



Fig. 39.26
Elija
Herramientas,
Opciones para
imprimir
solamente datos
de formulario.

2. Elija **Herramientas, Opciones**.

3. Seleccione la ficha Imprimir (véase la figura 39.26)

4. Seleccione **Imprimir datos** sólo para formularios del grupo Opciones para el documento actual, y elija [Aceptar].

5. Elija **Archivo, Imprimir**, seleccione las opciones de impresión, y elija [Aceptar].

Observe que este procedimiento asegura que cada vez que imprime este formulario, únicamente se imprimirán datos. Repita el procedimiento, desactivando la selección la opción **Imprimir datos** sólo para formularios del paso 4, si desea imprimir todo el formulario.

Cómo imprimir un formulario en blanco

Para crear su propio formulario preimpreso, imprima únicamente el formulario, sin los datos.

Para imprimir un formulario en blanco, siga estos pasos:

1. Elija **Archivo, Nuevo**: seleccione el formulario que quiere imprimir de la lista **Plantillas** y elija [Aceptar].
2. Sin rellenar el formulario, imprímalo, eligiendo **Archivo, Imprimir**; seleccione las opciones de impresión y elija [Aceptar].

Recuerde que los campos de un formulario aparecen sombreados si el botón **Sombreado de campo** está seleccionado en la barra de herramientas **Formulario**. Este sombreado no aparecerá cuando esté imprimiendo sus formularios. Si desea que el sombreado aparezca en un formulario impreso, use **Formato, Bordes y Sombreado** para sombrear áreas seleccionadas en su formulario.

SOLUCIÓN DE PROBLEMAS

Cuando imprimo los datos del formulario, no se alinean con los campos de formulario en el formulario preimpreso.

Asegúrese de que está utilizando el mismo formulario para imprimir sus datos que el utilizado para crear el formulario preimpreso.

Cómo guardar sólo los datos

Es posible que desee utilizar los datos recogidos en sus formularios con otro programa, como por ejemplo una base de datos. Para ello, guarde únicamente los datos e impórtelos al otro programa. Word guardará una copia de los datos en un documento **Sólo texto** (con la extensión **TXT**), creando un documento delimitado por comas que sólo contiene las respuestas en sus campos.

Muchas aplicaciones pueden leer los datos almacenados en archivos delimitados por comas. Microsoft Excel, por ejemplo, puede abrir y separar automáticamente cada porción de información en una celda de hoja de cálculo si el archivo usa la extensión **VSC** (valores separados por comas).

Para guardar sólo datos de su formulario, siga estos pasos:

1. Elija **Herramientas, Opciones**.
2. Seleccione la ficha **Guardar**.
3. En el grupo **Opciones**, seleccione la opción **Guardar datos** sólo para formularios.
4. Elija [Aceptar].
5. Elija **Archivo, Guardar Como** para guardar y asignar un nombre a su archivo de datos.

Cómo crear formularios con cuadros de diálogo Llenar

Con campos **{llenar}**, puede designar una carta de formulario, de manera que sólo sea necesario introducir un dato (como por ejemplo, un nombre)

una sola vez, independientemente del número de veces que aparezca en la carta. Esta opción es extremadamente útil para rellenar facturas, contratos, propuestas, o formularios de negocios en los que la mayor parte del cuerpo del texto permanece intacto. Los campos **(llenar)** son asimismo útiles cuando necesita insertar frases personales en grandes mailings. La figura 39.27 muestra un documento que puede crear para observar cómo funcionan los campos **(llenar)**.



En primer lugar, deberá crear una plantilla nueva para las cartas de formulario.

Para crear una plantilla nueva, siga estos pasos:

1. Elija **Archivo, Nuevo**. Aparecerá el cuadro de diálogo Nuevo (véase la figura 39.28).

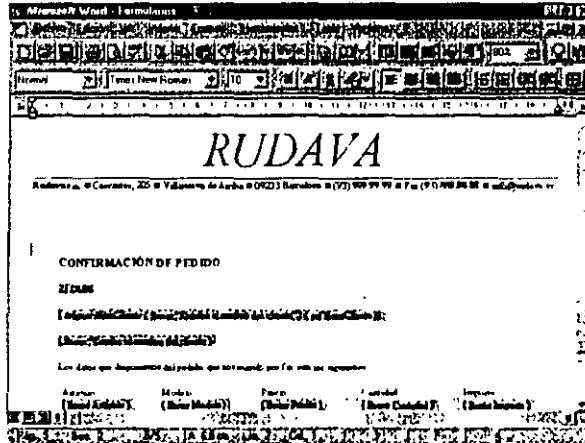


Fig. 39.27 Los códigos de campo pueden introducir datos o desplegar cuadros de diálogo que instan al usuario a introducir datos que pueden utilizarse repetidas veces en el documento.

2. Seleccione la opción **Plantilla** en el cuadro Nuevo.
Normal debe estar ya seleccionado en la lista **Plantilla**.
3. Seleccione una plantilla diferente en la que basar la carta, si lo prefiere. Puede haber creado una plantilla, por ejemplo, que incluya un membrete o el logotipo de la compañía.

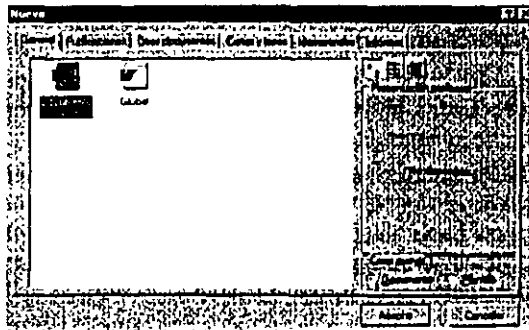


Fig. 39.28 Usando una plantilla para crear un formulario, impedirá que los usuarios cambien el original de forma accidental.

4. Elija [Aceptar].

5. Modifique la plantilla para que incluya cualquier cuerpo de texto, gráficos, tablas, etc., que desee incluir en la carta de formulario. Formatee el diseño de página de la plantilla para el membrete, si es necesario.

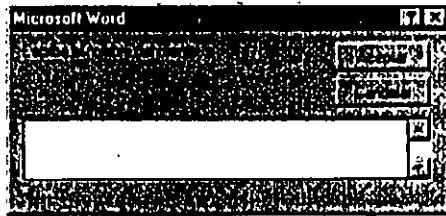
Mantenga la plantilla abierta y en pantalla de modo que pueda añadirle campos {llenar}, tal y como se describe en el siguiente apartado.

Cómo usar campos {llenar}

Para que su plantilla inste al usuario a introducir información clave, utilice campos {llenar}. La figura 39.29 muestra una pregunta de cuadro de diálogo generada por un campo {llenar}.

Para insertar el código de campo {llenar} en un documento, desplace el punto de inserción donde desee que aparezca la información del operador. Elija **Insertar, Campo**. Asegúrese de que [Todos] está seleccionado en la lista de categorías, y después elija **Llenar** en la lista **Nombre del campo**. En el cuadro de texto **Código**, coloque el punto de inserción después de **LLENAR**, escriba la pregunta para el usuario (sitúela entre comillas) y elija **Aceptar**.

Fig. 39.29
El campo {llenar} es una forma fácil de introducir datos en un cuadro de diálogo. No necesita crear una macro para desplegar el cuadro de diálogo.



Como alternativa, puede escribir el campo {llenar} en la plantilla. Para ello, coloque el punto de inserción donde desee que aparezcan los resultados, y después pulse <Control>+<F9>, la tecla de método abreviado **Insertar campo**. Coloque el punto de inserción entre los caracteres del campo, y escriba el tipo de campo y las instrucciones siguientes:

{llenar "Escriba la dirección del cliente."}

Para abrir el cuadro de diálogo que solicita el nombre del cliente y para actualizar el campo {llenar}, siga estos pasos:

1. Seleccione el carácter de campo en cualquiera de los extremos del código para seleccionar el campo {llenar}.
2. Pulse <F9>. Aparecerá el cuadro de diálogo **Llenar**.
3. Escriba la dirección de un cliente en el cuadro. Para empezar una nueva línea en el cuadro, pulse <Intro>. Elija [Aceptar] para completar el cuadro e inserte su entrada en el documento.

La entrada introducida por ud. aparecerá en el documento en el mismo lugar que el código de campo. El texto situado a continuación de la entrada insertada se desplaza hacia abajo o hacia la derecha, como si hubiera escrito el texto en esa ubicación manualmente. Para pasar de un campo a otro y ver sus resultados, pulse <Alt>+<F9>, o abra el cuadro de diálogo **Opciones**

Consejo

Incluya las preguntas que contengan más de una palabra entre comillas (" "). De lo contrario, Word utilizará sólo la primera palabra.

(eligiendo **Herramientas, Opciones**) y pulse en la casilla de verificación **Códigos de campo** de la ficha Ver.

Cómo reutilizar los resultados de los campos.

Si utiliza códigos de campo en cartas de formulario, puede pedir información al operador una vez, pero hacer que esa información aparezca en múltiples ubicaciones. Para reutilizar una entrada de un cuadro **{llenar}** en otras ubicaciones de una carta de formulario, deberá emplear los siguientes tres códigos de campo:

- **{fijar datos de marcador}** asigna datos a un marcador, que almacena la información para que se pueda usar más tarde. En el siguiente ejemplo, dado que el argumento de los datos para **{fijar}** son un campo **{llenar}**, la entrada del operador en respuesta al campo **{llenar}** se almacena en el nombre de marcador **Nomclien**. Si el dato es texto explícito que no cambia, como por ejemplo **Montana**, deberá incluirlo entre comillas. No incluya un espacio en el nombre de marcador.
- **{llenar [instrucción]}** despliega un cuadro de entrada de información en el que el operador puede introducir datos. Los corchetes ([]) indican que la instrucción es opcional.
- **{ref marcador}** muestra el contenido de un marcador en la ubicación del campo. Introduce este campo para repetir el contenido del marcador en otras ubicaciones dentro del documento.

La figura 30.30 muestra un código de campo que pide el nombre del cliente y lo almacena en el marcador **Nomclien**. El campo **{llenar}** pide el nombre. El campo **{fijar}** fija **Nomclien** igual que la entrada **{llenar}**. El campo **{ref}** muestra la entrada almacenada en **Nomclien**. Puede usar **{ref}** en toda la carta, después del campo **{fijar}**, incluso si el dato se ha introducido sólo una vez.

En la figura 30.30, los datos del campo **{llenar}** se introdujeron en un cuadro de diálogo. El código de campo **{fijar}** almacena la entrada en el marcador **Nomclien**. Los datos almacenados en **Nomclien** pueden ser mostrados otra vez en cualquier parte del documento con **{ref Nomclien}**. El código de campo **{ref}** referencia los datos almacenados en ese marcador. Puede reutilizar **{ref}** en el documento tantas veces como lo desee. Usando los conmutadores, podrá formatear la información que **{ref}** le devuelva.

Quizás recuerde que campos *anidados* son un código de campo dentro de otro código de campo. En el ejemplo, un campo **{llenar}** está anidado dentro de un campo **{fijar}**. El resultado de **{llenar}** se utiliza para proporcionar uno de los argumentos requeridos por el campo **{fijar}**. Para construir el campo anidado de la figura 30.30 desde dentro hacia afuera, siga estos pasos:

1. Coloque el punto de inserción donde necesite insertar la entrada.
2. Pulse <Control>+<F9>, la tecla de método abreviado para insertar campo.
3. Entre los caracteres del campo, escriba **llenar**, un espacio, y después la instrucción para el operador (incluida entre comillas), como "Escriba el nombre del cliente"

Consejo

Para actualizar los campos **{llenar}** en un documento entero seleccione por el documento y después pulse <F9> (Actualizar)

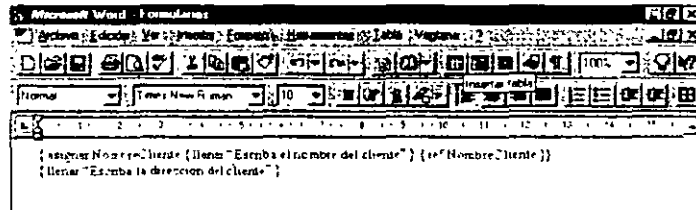
Consejo

Puede inventar sus propias palabras simples para usarlas como marcadores

Fig. 39.30

La combinación de códigos de campo {llenar}, {ref} y {ref} permiten al usuario rellenar un cuadro de diálogo y hacer que los datos se utilicen en todo el documento.

4. Seleccione el campo que acaba de escribir. Para seleccionar los caracteres y el contenido de un campo, seleccione un carácter de campo en uno de los extremos para seleccionar el campo entero.
5. Pulse <Control>+<F9> para encerrar la selección en caracteres de campo.



Este paso *anida* el primer campo completamente dentro de otro grupo de caracteres de campo. El punto de inserción se desplaza para situarse inmediatamente a continuación del primer carácter de campo.

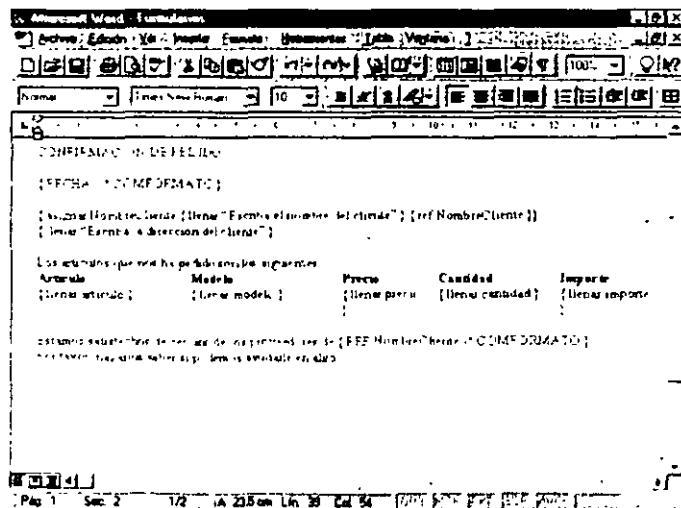
6. Escriba *fixar* inmediatamente después del primer carácter de campo, luego un espacio y el marcador apropiado, como por ejemplo *Nombre*. Deje un espacio después del marcador, pero no deje espacios en el nombre.

Este nuevo campo anidado solicita un nombre y lo almacena en el marcador, pero la entrada no aparece en pantalla. Para ver el resultado del campo, deberá actualizar el campo.

Para actualizar los dos nuevos campos y ver cómo se solicita y cómo se muestra el nombre del cliente, siga estos pasos:

Fig. 39.31

Reutilice los datos repitiendo la combinación {ref marcador} en cualquier lugar donde desee que aparezcan los datos {ref} solo muestra los nuevos datos cuando está seleccionado y actualizado



1. Seleccione la línea o líneas enteras que contengan ambos campos.
2. Pulse <F9> (Actualizar campos).
Aparecerá un cuadro de diálogo, pidiendo el nombre del cliente (véase la figura 39.29).
3. Escriba la entrada tal y como se requiere.
4. Elija [Aceptar].

El campo {fijar} almacena en el marcador el nombre que ha introducido en el campo {llenar}. El campo {ref marcador} muestra el contenido de un marcador en la carta. Puede introducir un campo {ref marcador} en múltiples ubicaciones del documento, en cualquier lugar donde necesite que se repita el nombre. En la figura 39.31, el marcador Nomclien se repite en la última línea de la pantalla. Sin embargo, el nuevo contenido de {ref marcador} no aparecerá hasta que cada campo {ref marcador} se actualice.

Después de introducir todos los códigos de campo, elija **Archivo, Guardar como**, y elija [Aceptar] para guardar la plantilla.

Estos campos muestran datos introducidos en el bloque dirección; también solicitan y muestran el tipo, modelo, y costo de los ítems pedidos por el cliente. El conmutador *mergeformat, usado en el campo {ref marcador *mergeformat}, asegura que el formato que ha aplicado al nombre de campo en el documento no cambia al actualizar el campo. El formato de los resultados del campo se corresponde con el formato que ha aplicado a la letra r en ref.

La inclusión del conmutador *mergeformat se controla por medio de la opción **Conservar formato** al actualizar en el cuadro de diálogo Campo.

Cómo guardar y asignar un nombre a la plantilla

Este documento debe ser guardado como plantilla porque lo ha abierto como tal. Para guardar su plantilla, elija **Archivo, Guardar como**. En el cuadro de texto **Nombre de archivo**, escriba un nombre de ocho letras para describir su formulario. Observe que no puede cambiar muchos de los cuadros de texto o de lista. Elija [Aceptar]. Word guardará la plantilla y añadirá la extensión DOT. Cuando guarde una nueva plantilla, asígnele un nombre que refleje el tipo de documento que ésta crea.

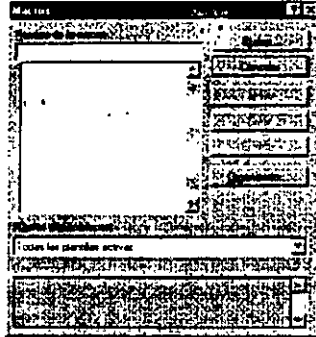
Cómo actualizar campos en un formulario

Para comprobar los campos que ha introducido, actualícelos. Esta acción desplegará o le permitirá introducir nuevos valores.

Para introducir datos en los campos {llenar} y actualizar los campos {ref}, siga estos pasos:

1. Elija **Archivo, Nuevo**, y seleccione una plantilla que contenga campos {llenar}. Elija [Aceptar].
2. Desplace el punto de inserción a la parte superior del documento y después pulse <F11> para seleccionar el siguiente campo. (Pulse <Mayús>+<F11> para seleccionar el campo anterior, o seleccione el documento entero eligiendo **Edición, Seleccionar todo** para actualizar todos los campos.)

Fig. 39.32
Puede grabar una macro que actualice automáticamente e códigos de campo cuando se abra un documento o plantilla.



3. Pulse <F9> para actualizar el campo seleccionado o, si el documento entero está seleccionado, para actualizar un campo a la vez, desde el principio hasta el final del documento.

4. Escriba los datos solicitados, y elija [Aceptar].

Si aparece un error en la ubicación del campo, despliegue los códigos de campo en el documento. Compruebe que la ortografía y el espaciado sean correctos. Utilice un espacio entre tipos de campo, instrucciones y conmutadores.

Consejo

Para conservar la entrada anterior, seleccione Cancelar, no ocurra nada si no existía una entrada anterior

Cómo crear una macro para actualizar campos automáticamente

Dado que este tipo de formulario a rellenar está diseñado para usarse repetidamente, puede hacer que Word le pida automáticamente al usuario la información al abrirse el documento. Puede hacer esto con una macro automática que actualice todos los campos del documento cuando éste se abre. Con la plantilla a rellenar como documento activo, siga estos pasos:

1. Elija **Herramientas, Macros**. Aparecerá el cuadro de diálogo Macros (véase la figura 39.32).
2. Escriba **AutoNuevo** en el cuadro **Nombre de la macro**.
3. En la lista **Macros disponibles** en, seleccione la opción **Documentos basados en [plantilla]**, donde [plantilla] es el nombre de la plantilla. La macro estará disponible sólo para esta plantilla o para documentos que se originen a partir de ella.
4. Pulse en el botón **Grabar**.
5. Introduzca una descripción, como **Actualiza todos los campos automáticamente**, en el cuadro **Descripción**.
6. Elija [Aceptar].

Aparecerá la barra de herramientas Grabar Macro, y el indicador **REC** aparecerá destacado en la barra de estado, indicando que la grabadora está activada. Siga estos pasos para grabar un proceso que actualiza todos los campos de un documento:

1. Pulse <Control>+<5> (utilice el teclado numérico) para seleccionar el documento entero.

2. Pulse <F9>. Aparecerá una instrucción generada por el primer campo {llenar}.
3. Pulse en el botón [Cancelar] para cada instrucción {llenar}.
4. Pulse <Control>+<Inicio> para volver a colocar el punto de inserción en la parte superior del documento.
5. Pulse en el botón [Detener] de la barra de herramientas Grabar macro.

Elija **Archivo, Guardar todo** para guardar la macro y cerrar la plantilla.

Para verificar la macro, siga estos pasos:

1. Elija **Archivo, Nuevo**.
2. Seleccione la plantilla en la lista **Plantilla**.
3. Elija [Aceptar].

Cuando abra el documento, la macro AutoNuevo ejecutará la macro de actualización. Introduzca una respuesta en cada cuadro de diálogo o elija Cancelar. Si la macro no funciona correctamente, grábela otra vez. Si vuelve a grabar empleando el mismo nombre (AutoNuevo), Word le preguntará si desea reemplazar la grabación anterior. Elija **Sí**.



PART I

Architecture

Chapter 1, "Microsoft Office Architecture," describes the overall architecture of Office. It emphasizes the components of Office that are shared among the Office applications, and provides an overview of the Office features that help the Office applications run smoothly together.

Chapter 2, "Microsoft Access Architecture," describes the architecture of Microsoft Access, including the Jet Database Engine, library databases, ISAMs, database objects, and how Microsoft Access resolves conflicts.

Chapter 3, "Microsoft Excel Architecture," describes the architecture of Microsoft Excel, including information on workbooks, cell values, formulas, cell formatting, charts, AutoFormats, macros, security, data mapping, and how Microsoft Excel resolves conflicts.

Chapter 4, "Microsoft PowerPoint Architecture," describes the architecture of PowerPoint, including an explanation of presentation templates, design templates, master elements, and how PowerPoint resolves conflicts.

Chapter 5, "Microsoft Schedule+ Architecture," describes the architecture of Schedule+, including information on the differences between running Schedule+ in group-enabled mode versus working alone, how Schedule+ resolves conflicts, and interoperability with Microsoft Exchange.

Chapter 6, "Microsoft Word Architecture," describes the architecture of Word, including detailed information about templates and how they work; how Word resolves conflicts; how to manage styles, AutoText entries, macros, and toolbars; whether components are stored with the template, the document, or the Word application; and how to create wizards and add-ins.

CHAPTER 1

Microsoft Office Architecture

Microsoft Office for Windows 95 teams up with the Windows 95 operating system to implement a method of work that is centered around information rather than applications. Office allows users to create information, then manipulate it with a suite of software tools. Each of these tools is a standalone application in its own right, yet Office integrates their functionality to form a unified work area—like the office metaphor for which it is named.

In This Chapter

Coordination of Applications	35
Implementation of OLE	36
Interoperability with Electronic Mail	42
Shared Application Tools	44
Shared Code	46
Shared Services	56

Coordination of Applications

Office for Windows 95 coordinates the work of Microsoft Excel, Word, PowerPoint, Schedule+, and Microsoft Access. From a user's point of view, this is apparent in the consistency of the user interface. Menu and command names are standardized, not only in their terminology, but in their order and placement on the screen. If users are already familiar with one Office application, it will be easy for them to use the tools of another application. From an architectural point of view, the Office applications form an integrated whole.

Implementation of OLE OLE is the basis for integration among the Office applications. Each of the Office applications is written to this standard.

Interoperability with Electronic Mail Each of the Office applications supports interoperability with a variety of electronic mail systems, including Microsoft Exchange, the mail client installed with Windows 95.

Shared application tools. Individual Office applications share tools such as spell checking, Microsoft Graph, Equation Editor, Data Map, and so forth. These tools are provided with the applications whether purchased individually or as part of Office.

Shared code As components of Office, the individual applications in Office share code in Office dynamic link libraries (DLLs). These DLLs provide functionality that is not available when the applications are purchased individually.

Shared services Office adds its own services to the Office applications. For example, the Microsoft Office Shortcut Bar enhances your ability to customize and standardize users' desktops, and Office Binders allows users to manage groups of documents from separate applications as if they were a single publication.

This chapter is to help you better understand these features, in order to better support your users.

Implementation of OLE

OLE is the basis for integration between the Office applications. OLE allows users to drag documents from a folder to an Office Binder or to their Microsoft Exchange Inbox so that the document retains its own properties while it gains the services of the *container* or destination application. OLE treats these compound documents as a single entity, yet maintains the uniqueness of each document.

OLE also allows users to insert data from a source document into a client document—cells from a Microsoft Excel workbook into a Word document, for example. Data can be linked or embedded. A *linked* object is a reference to the source and is updated dynamically with the source data. An *embedded* object is a copy of the source data that can be edited with the tools from the source application; it has no link to the source data. Documents with embedded objects take up more memory than documents with linked objects. This is because embedded objects are copies whereas linked objects are merely references.

OLE Properties

OLE standardizes the way that applications store properties for documents. Office uses a standard set of properties across all applications. In addition, users can define a wide variety of custom properties. Properties are stored with the document file. For details concerning the specifications for properties, see the *OLE Software Developer's Kit*, published by Microsoft Press and available at your local bookstore.

The OLE properties stored with a document are accessible to any application that supports OLE. For example, when a user drags and drops a Microsoft Excel worksheet into a Word document, the user can manipulate the worksheet's properties through Word.

Office applications include the following sets of OLE properties:

Summary information These properties are standard ones, familiar to users of Office 4.x as those that appear in the Summary Info dialog box. More properties have been added in Office for Windows 95 applications.

Document summary information These properties provide an enhanced set of properties, over and above those found in the Summary Info set.

Custom These properties are defined by the user with string names rather than property IDs. This allows users to create any property they want with any name they want.

For each of the Office for Windows 95 applications, the Property dialog box includes the following tabs:

- General
- Summary
- Statistics
- Contents
- Custom

Most of the properties that appear on the General, Summary, and Statistics tabs will be familiar to users of previous versions of Office applications. These are the Summary Information properties, with some Document Summary Information properties added.

Properties that appear on the Contents tab are new. These are Document Summary information properties.

Properties that appear on the Custom tab are also new. They are Custom properties that users define themselves.

The following tables show the source for each group of properties, by tab.

General Properties

The following table lists the source for the information displayed on the General tab.

Property	Source
Icon of application that created document	System
Filename	System
Type	System
Location	System
Size	System
MS-DOS name	System
Created	System
Modified	System
Accessed	System
Attributes	System

Summary Properties

The following table lists the source for the information displayed on the Summary tab.

Property	Source
Title	PID_TITLE
Subject	PID_SUBJECT
Author	PID_AUTHOR
Manager	PID_MANAGER
Company	PID_COMPANY
Category	PID_CATEGORY
Keywords	PID_KEYWORDS
Comments	PID_COMMENTS
Template	PID_TEMPLATE

Statistics Properties

The following table lists the source for the information displayed in the Statistics tab:

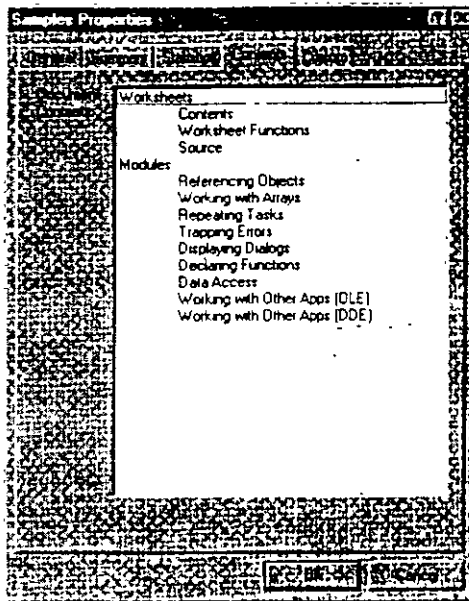
Property	Source
Created	PID_CREATE_DTM ²
Modified	System
Accessed	System
Printed	PID_LASTPRINTED
Last saved by	PID_LASTAUTHOR
Revision number	PID_REVNUMBER
Total editing time	PID_EDITTIME
Pages ¹	PID_PAGECOUNT
Paragraphs ¹	PID_PARCOUNT
Lines ¹	PID_LINECOUNT
Words ¹	PID_WORDCOUNT
Characters ¹	PID_CHARCOUNT
Bytes ¹	PID_BYTECOUNT
Slides ¹	PID_SLIDECOUNT
Notes pages ¹	PID_NOTECOUNT
Hidden slides ¹	PID_HIDDENCOUNT
MM clips ¹	PID_MMCLIPCOUNT
Presentation Target ¹	PID_PRESFORMAT
Scale Crop	PID_SCALE

¹ These statistics are made available according to the Office application. For example, Microsoft Excel uses none of these statistics, Word uses Pages, Paragraphs, Lines, Words, and Characters, while PowerPoint uses Slides, Paragraphs, Words, Notes, Hidden Slides, and Multimedia Clips.

² The value displayed here is derived from the property ID and not from the system, as on the General property tab. The value on the General tab tells when the document was created on the system and is not necessarily when the document was created absolutely.

Contents Properties

The Contents tab stores names of the different sections in the document (documents in a binder, sheet names in a workbook, slide titles in a presentation, and so forth). The following illustration shows the contents properties of a Microsoft Excel workbook.



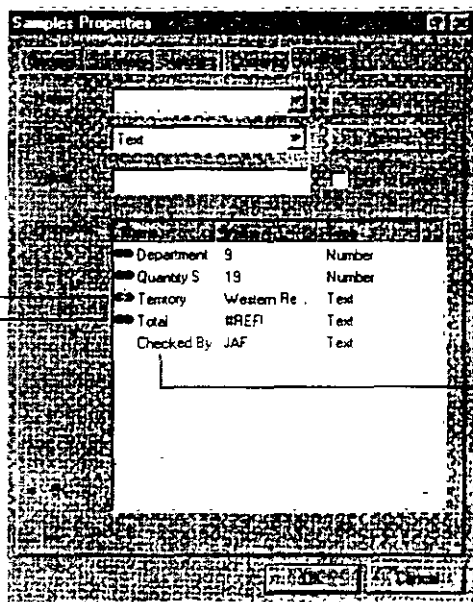
Custom Properties

The Custom tab provides an interface for users to create their own properties, which are either constants stored in the property stream or are linked to some document content, such as PowerPoint text, a Word bookmark or a Microsoft Excel named range. A *property stream* is a predefined part of a structured storage (file) that is used for storing the property data.

Valid property types are number (VT_I4 or VT_R8), date (VT_DATE), boolean (VT_BOOL), and text (VT_LPSTR).

When a property is linked to document content, the user chooses the source for the property value, and this value is displayed in the property list with a linked chain icon next to the entry. If the link source is not available, for example, if the name the link points to has been deleted, the icon changes to a broken chain link, shown in the following example.

Deleting a named reference in the Samples workbook broke this linked custom property.



This custom property is not linked to content of the Samples workbook.

Deleting data from the Samples workbook created a reference error for this custom property, but a link still exists to the named reference.

User-defined property names are stored separately from other properties so that users cannot create conflicts by naming a Custom property with the same name as a Summary Information or Document Summary Information property. For example, the property Author already exists as a Summary Information property, but a user could create a property named "Author." These names are stored separately in the property list and displayed on separate tabs in the Property dialog box.

In addition, users can hide properties of documents posted to a Microsoft Exchange public folder by using the underscore character as the first character of the property name. For example, a user may want to create a property strictly for selection purposes, such as "EmployeesEarningGreaterthan80k," which the user would never want to appear. The user can make the property invisible in the Microsoft Exchange viewer by including an underscore character as the first character of the string (for example, "_EmployeesEarningGreaterthan80k").

Note The underscore does not cause the Property dialog box to hide properties. Rather, it is an advisory to an application that reads properties, such as the Microsoft Exchange Server, that it should not display the property.

Interoperability with Electronic Mail

The individual Office applications detect whether a Messaging API (MAPI), such as the Microsoft Exchange client, Microsoft Mail, or cc:Mail™, is installed on the system. To determine existence of MAPI, the application checks the DLL32 value of the Registry key

Hkey_Local_Machine\Software\Microsoft\Microsoft Office\95\MapiDLL, and uses the DLL specified in this path. If this key does not exist, the application uses the Windows MAPI DLL, Mapi32.dll.

If the Office application detects a 16-bit vendor independent messaging (VIM) system, such as cc:Mail, the application installs a 32-bit DLL that converts MAPI calls to VIM, Mapivi32.dll, and sets the path in the Registry. Mapivi32.dll is a 32-bit DLL. To communicate with the 16-bit VIM mail provider, it requires the 16-bit DLL, Mapivitk.dll. For more information on how 16-bit applications communicate with 32-bit DLLs, see "Porting Your 16-bit Office-Based Solutions to 32-bit Office" in Chapter 12.

If the Office application detects a mail provider, it adds Send and Add Routing Slip commands to the File menu. This allows users to send documents directly from their application, without having to launch their mail software manually.

With the Microsoft Exchange client installed, users can use Word as their e-mail editor (WordMail), and can enable group scheduling by combining the use of Schedule+, the Microsoft Exchange client, and a postoffice. That postoffice can be the local type you install with Windows 95 or any other valid MAPI mail server that supports MAPI 1.0.

Office applications also check for a connection to the Microsoft Exchange Server on the user's system. To determine whether or not a connection to the Microsoft Exchange Server exists, the Office application checks for the Registry key Hkey_Local_Machine\Software\Microsoft\Exchange\Client\Extensions for installed mail providers. If the application finds a value for the Microsoft Exchange Server, it adds the Post To Microsoft Exchange Folder command to the File menu. This allows users to post documents to a public folder.

For more information about the interoperability between Office and electronic mail, see "Interoperability with Electronic Mail" in Chapter 15.

Documents Stored in Microsoft Exchange Public Folders

The Microsoft Exchange public folder can store three main types of items: documents, e-mail messages, and forms. By either dragging and dropping from the file system or by posting a document from an application, users can store documents in a public folder. For more information on public folders, see the documentation that comes with the Microsoft Exchange Server.

Public folders store documents as OLE objects, contained in invisible mail messages. To users, the documents inside a public folder may appear to be files stored in a file structure on disk. However, Microsoft Exchange public folders treat documents as mail messages, not files. This leads to two major differences between a Microsoft Exchange public folder and a folder in the disk file structure:

- Public folders can have multiple documents with the same name.
- Microsoft Exchange allows more than one person to open and edit a document at the same time, with *optimistic file locking*. With optimistic file locking, if more than one user is accessing the same server, the first user to save the document will have data saved without any messages. When other users go to save the document, Microsoft Exchange issues a warning that the message has been changed by another user, and provides an option for users to lose their changes or to save the document as a new document.

If each of these users is on a different server, the message is put into a conflict state. The most recently changed version is designated as the master message, and the other messages are attached to it. The master message is marked as being in conflict. When a user opens it, a form shows the different versions and allows a user with Edit All permission to merge the changes.

Note The Microsoft Excel file sharing feature does not work for documents in public folders. This is because Microsoft Excel cannot determine whether more than one person is accessing a document when the document is embedded in a message.

Properties Exchanged with Microsoft Exchange

When users store a document in a Microsoft Exchange public folder, Microsoft Exchange initiates a process to read out the properties from the document, and promote them to be MAPI properties.

There are three different sets of properties that Microsoft Exchange promotes:

- The 18 standard SummaryInformation properties
- The new standard DocumentSummaryInformation
- All of the UserDefinedProperties

After the properties are promoted, they become available in the Microsoft Exchange user interface. With the document highlighted, users can click Properties on the File menu to see the properties. Any user-defined properties that include an underscore as the first character are not displayed in the list of property names in Microsoft Exchange.

Users can search on properties in Microsoft Exchange, using the Advanced button in the Open dialog box, similar to how they search for properties from the Open dialog box in Office applications.

Shared Application Tools

This section discusses tools that are shared by Office applications, such as Microsoft Organization Chart, WordArt, AutoCorrect, Microsoft Graph, and so forth. In a way, these are standalone applications. Most include their own executable (.exe) file, and many include their own dynamic link libraries (DLLs) which must be present for the tool to run. Rather than being components of Office, these tools are included with one or more of the Office applications. For example, Word and PowerPoint share Microsoft Graph whether or not Office is installed on the user's system.

Applications that serve as tools for the Office applications are identified in the Registry, under the Hkey_Local_Machine\Software\Microsoft\Shared Tools keys. The default installation path for most of these tools is the \Program Files\Common Files\Microsoft Shared folder.

Many of these tools are applications that must be run from within an Office application. For example, if users double-click the icon for Graph5.exe in the Windows Explorer, they will get an error message stating that the application must be run from within another program. To run Microsoft Graph, the user must be running Microsoft Word, PowerPoint, Microsoft Excel, or Microsoft Access. Some of these tools are *OLE servers*, which means they are available when users click Object on the Insert menu of any application that supports OLE.

The following table lists the application tools shared by Office applications, the applications that share them, and their default location as Office Setup installs them. For a complete listing of these tools, see Appendix C, "List of Installed Components."

Tool	Applications that share	Default installation location
Art Gallery	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\Artgairy
Data Access Objects	Microsoft Access, Microsoft Excel	\Program Files\Common Files\Microsoft Shared\Dao
Data Map	Microsoft Access, Microsoft Excel	\Program Files\Common Files\Microsoft Shared\DataMap

Tool	Applications that share	Default installation location
Equation Editor	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\Equation
Graphics Filters	PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\Grphflt
Microsoft Graph	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\Msgraph5
Microsoft Info	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Windows\Msapps\Msinfo
Microsoft Query	Microsoft Access, Microsoft Excel, Word	\Program Files\Common Files\Microsoft Shared\Msquery
Microsoft Organization Chart	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\Orgchart
Spell checking dictionary ¹	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\Proof
Microsoft WordArt	Microsoft Access, Microsoft Excel, PowerPoint, Word	\Program Files\Common Files\Microsoft Shared\WordArt

¹ See the following section for more information about this tool.

Spell Checking Dictionary

The Office applications that implement spell checking all share the same dictionary file, Custom.dic. The default path for this file is the user's \Program Files\Common Files\Microsoft Shared\Proof folder. Customizations that users make to the spelling list by running spell check within an application are saved to this file.

It is possible to create a corporate dictionary, although you cannot share this file across a workgroup due to potential sharing violation problems.

► To create a corporate dictionary file

1. Open the file Custom.dic in a text editor, such as Notepad.
2. Edit the text file.
3. Save the file and install it to users' local workstations.
4. Edit the users' Registry key
Hkey_Local_Machine\Software\Microsoft\Shared Tools\Proofing Tools\Custom Dictionaries to point to the location of the corporate Custom.dic.

Shared Code

Office applications share code through the Office DLLs. These DLLs allow Office to integrate operations—such as creating a new document and sharing an AutoCorrect list—across all the Office applications. DLLs for Office, for the Office applications, and for shared applications often make calls to other DLLs, including Windows 95 system DLLs, so if another application on a user's system overwrites an existing DLL, Office may not run properly. This section discusses the Office DLLs and lists the DLLs called by the Office application programs.

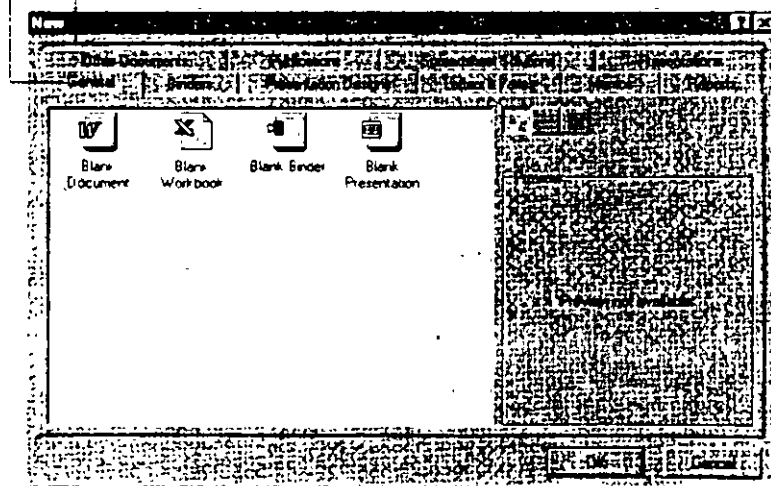
Office DLLs

The Office DLL primarily responsible for supporting all shared Office components is Mso95.dll. Mso95.dll allows users to create new documents in a way that integrates the Office applications with one another. Mso95.dll also creates a single list of entries that is accessed by AutoCorrect in each of the Office applications. The following section describes these two features that Mso95.dll supports.

Starting a New Document

When users click the Start A New Document button on the Microsoft Office Shortcut Bar, a dialog box appears, showing all available templates. When users double-click a template icon, the Office application associated with the template creates a new, blank document based on the template. By default, the templates shown in this dialog box are from the user's \Msoffice\Templates folder and its subfolders.

- Templates that appear on the General tab are stored in the user's Templates folder
- Templates that appear on other tabs are stored in subfolders of the user's Templates folder.



Msoffice\Templates is the default location for templates on users' local computers. Users can modify this setting by clicking Customize on the application control menu of the Microsoft Office Shortcut Bar and then clicking the Settings tab. This tab also allows users to set a location for templates on a network share, by clicking Workgroup Templates location and then clicking Modify.

You can maintain these settings for your users in the Registry key Hkey_Current_User\Software\Microsoft\Microsoft Office\95\FileNew. The path specified in the LocalTemplates key is the User Templates location, and the path specified in the SharedTemplates key is the Workgroup Templates location. For information on customizing these settings during Setup, see Chapter 9, "Customizing Client Installations."

If users have defined both the User Templates and Workgroup Templates locations, the New dialog box displays all available templates from both locations. Templates stored in the root folder of each location appear on the General tab of the New dialog box, and templates stored in subfolders appear on separate tabs. The names of the separate tabs match the names of the subfolders. For information on how to customize the tabs of the New dialog box, see "Supporting Custom Templates" in Chapter 12, "Support and Troubleshooting."

AutoCorrect List

The AutoCorrect feature, formerly available only in Word, allows users to automate corrections to frequently misspelled words and to create abbreviations that are expanded automatically for commonly used text. Because Office applications share an AutoCorrect list, all documents that a user creates in Office will have consistent spellings and stylistic conventions. For example, users can set AutoCorrect to replace the abbreviation "ppt" with "PowerPoint" as they type, and the convention of capitalizing the second "p" is maintained automatically.

Office installs a file containing a default AutoCorrect list, Msoffice.acl, to the user's Windows folder. When the Office application saves the user's personal list, it is stored as *Username*.acl, where *username* is the user's logon name and *nnn* is a number to further identify the user. You can modify the path for either the default or the user list. If you do this, you must modify the Windows Registry entry that points to this file, as explained later in this section.

When an Office application starts, the AutoCorrect code in Mso95.dll does the following:

1. Loads the .acl file pointed to in the Registry key
Hkey_Current_User\Software\Microsoft\Microsoft Office\95\AutoCorrect\List with the value named by Path.
2. Opens the file pointed to in the Registry key
Hkey_Local_Machine\Software\Microsoft\Shared Tools\AutoCorrect\List to find a default path, if the key/value in step 1 does not exist.

When the user quits the Office application, the AutoCorrect code in Mso95.dll does the following:

1. Saves the AutoCorrect list as *Usenamennn.acf*.
2. Writes the Hkey_Current_User key to point to the user's list file.

Users can modify AutoCorrect entries by clicking AutoCorrect on the Tools menu. These entries are stored in the Registry, in the key Hkey_Current_User\Software\Microsoft\Microsoft Office\95\AutoCorrect\Settings.

Caution The value of the Hkey_Current_User\Software\Microsoft\Microsoft Office\95\AutoCorrect\Settings key is binary and should not be edited. Modify AutoCorrect entries with the AutoCorrect command on the Tools menu in any Office application.

- ▶ **To create a customized AutoCorrect list and distribute it to your users**
1. Run an Office application, and modify the default list in the AutoCorrect dialog box.
 2. Quit the application, and find the resulting *Usenamennn.acf* file.
 3. Store this file where your users can access it.
 4. If you want to make this the default list for your users, you can modify users' Registry key Hkey_Local_Machine\Software\Microsoft\Shared Tools\AutoCorrect\List path entry to point to the file.

When users modify the master list and quit an Office application, the default list is stored on the local computer as *Usenamennn.acf*. When users customize their copy of the list, the changes do not affect the master list. To reinstall the original master list, users can delete their customized list and restart the Office application.

LLs Called by Office Applications

Office applications call DLLs from executable program (.exe) files and application DLLs. This section lists those DLLs that may be called *statically*, when the application starts up in its default state. Other DLLs could be called *dynamically*, controlled by users while running the applications or while running macros, converters, drivers, or add-ins.

The following lists of statically called DLLs are to help you troubleshoot for your users when they start up an application. For example, if a user's Office Binder does not run properly, you could check to see that none of the DLLs associated with Binder.exe have been deleted or overwritten by some other application on the user's system. The lists include DLLs that are called by other DLLs. For example, if an application calls Mso95.dll, the DLLs called by Mso95.dll are also listed in the table.

DLLs Called by Office

Executable files and DLLs associated with Office and its components call DLLs as shown in the following table:

Calling file	DLLs
Binder.exe	Kernel32.dll, Advapi32.dll, User32.dll, Gdi32.dll, Cmdlg32.dll, Shell32.dll, Comctl32.dll, Mso95.dll, Oledlg.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll
Fastboot.exe	Kernel32.dll, User32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll
Findfast.exe	Gdi32.dll, Kernel32.dll, User32.dll, Ole32.dll, Advapi32.dll, Mso95.dll, Shell32.dll, Comctl32.dll, Msvcr20.dll, Rpcrt4.dll
Msoffice.exe	Kernel32.dll, User32.dll, Gdi32.dll, Advapi32.dll, Shell32.dll, Comctl32.dll
Mso95.exe	Shell32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Advapi32.dll, User32.dll, Kernel32.dll, Mso95.dll, Gdi32.dll, Comctl32.dll
Bdrec.dll	Kernel32.dll, Advapi32.dll, User32.dll, Gdi32.dll, Comctl32.dll, Ole32.dll, Mso95.dll, Shell32.dll, Msvcr20.dll, Rpcrt4.dll
Commntb32.dll	Gdi32.dll, Kernel32.dll, User32.dll
Docobj.dll	User32.dll, Rpcrt4.dll, Kernel32.dll, Advapi32.dll
Explode.dll	Kernel32.dll, Advapi32.dll, User32.dll, Gdi32.dll, Shell32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Comctl32.dll
Mfcans32.dll	Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Kernel32.dll, User32.dll, Advapi32.dll
Mfcuia32.dll	Oledlg.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Mfcans32.dll, Kernel32.dll, Oleaut32.dll, User32.dll, Advapi32.dll
Mso95.dll	User32.dll, Gdi32.dll, Advapi32.dll, Shell32.dll, Kernel32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Comctl32.dll
Mso95fx.dll	Kernel32.dll
Mstool32.dll	Advapi32.dll, Kernel32.dll, User32.dll
Openenu.dll	Kernel32.dll
Osbreg.dll	User32.dll, Advapi32.dll, Kernel32.dll

DLLs Called by Microsoft Access

Executable files and DLLs associated with Microsoft Access and its components call DLLs as shown in the following table:

Calling file	DLLs
Msaccess.exe	Msjt3032.dll, Mswng300.dll, Vba232.dll, Mso95.dll, Ole32.dll, Oleaut32.dll, Msvcr20.dll, Gdi32.dll, User32.dll, Advapi32.dll, Kernel32.dll, Comctl32.dll, Shell32.dll, Rpcrt4.dll, Mpr.dll
Wrkgadm.exe	Comdlg32.dll, Advapi32.dll, User32.dll, Kernel32.dll, Gdi32.dll
Msaccreg.dll	Advapi32.dll, Mpr.dll, Mfcans32.dll, Ole32.dll, User32.dll, Kernel32.dll, Msvcr20.dll, Rpcrt4.dll, Oleaut32.dll
Msaxp30.dll	User32.dll, Gdi32.dll, Comdlg32.dll, Advapi32.dll, Shell32.dll, Kernel32.dll, Comctl32.dll
Msain300.dll	Msvcr20.dll, Kernel32.dll, User32.dll
Msau7032.dll	Version.dll, Gdi32.dll, User32.dll, Comdlg32.dll, Advapi32.dll, Msvcr20.dll, Msjt3032.dll, Oleaut32.dll, Kernel32.dll, Mpr.dll, Ole32.dll, Rpcrt4.dll
Msjrci30.dll	Kernel32.dll
Msjtrclr.dll	Msjt3032.dll, Mswng300.dll, Version.dll, Vbajet32.dll, Gdi32.dll, Msjter32.dll, Mpr.dll, Kernel32.dll, User32.dll, Advapi32.dll, Ole32.dll, Oleaut32.dll, Msvcr20.dll, Rpcrt4.dll, Msjint32.dll
Oc30.dll	Mfcans32.dll, Msvcr20.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Advapi32.dll, Ole32.dll, Shell32.dll, Kernel32.dll, Oleaut32.dll, Ole32.dll, Rpcrt4.dll, Comctl32.dll
Soa300.dll	Gdi32.dll, User32.dll, Ole32.dll, Kernel32.dll, Msvcr20.dll, Rpcrt4.dll

DLLs Called by Microsoft Excel

Executable files and DLLs associated with Microsoft Excel and its components call DLLs as shown in the following table:

Calling file	DLLs
Excel.exe	Mso95.dll, User32.dll, Gdi32.dll, Advapi32.dll, Shell32.dll, Kernel32.dll, Ole32.dll, Xlnt132.dll, Msvcr20.dll, Rpcrt4.dll, Comctl32.dll
Analys32.xll	Kernel32.dll, Xlcall32.dll
Solver32.dll	Xlcall32.dll, Kernel32.dll, Olecli32.dll, User32.dll
Xlcall32.dll	Kernel32.dll
Xlkey32.dll	Kernel32.dll, User32.dll
Xlkey32.dll	Kernel32.dll, User32.dll

Calling file	DLLs
Xlodbc32.dll	Kernel32.dll, Xlcall32.dll, User32.dll, Odbc32.dll, Msvcr20.dll, Advapi32.dll
Xlodbc32.dll	Xlcall32.dll, Kernel32.dll, User32.dll, Odbc32.dll, Msvcr20.dll, Advapi32.dll
Xlqpw.dll	Kernel32.dll

DLLs Called by Microsoft PowerPoint

Executable files and DLLs associated with PowerPoint and its components call DLLs as shown in the following table:

Calling file	DLLs
Powerpnt.exe	Kernel32.dll, Gdi32.dll, User32.dll, Shell32.dll, Advapi32.dll, Comctl32.dll, Ole32.dll, Oleaut32.dll, Msvcr20.dll, Rpcrt4.dll, Mso95.dll
Graflink.exe	Comctl32.dll, Kernel32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Advapi32.dll, Version.dll
Autoclip.dll	Kernel32.dll, User32.dll, Oleaut32.dll
Autoclip.dll	Kernel32.dll, User32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll
Ghost.dll	User32.dll, Kernel32.dll
Msroute.dll	Kernel32.dll, User32.dll, Gdi32.dll, Advapi32.dll
Pngcomp.dll	Kernel32.dll, User32.dll
Pp4x32.dll	Ole32.dll, Kernel32.dll, User32.dll, Gdi32.dll, Advapi32.dll, Msvcr20.dll, Rpcrt4.dll
Pp4x32.dll	Ole32.dll, Kernel32.dll, User32.dll, Gdi32.dll, Advapi32.dll, Msvcr20.dll, Rpcrt4.dll
Ppt7reg.dll	Kernel32.dll, Advapi32.dll
Ppt7reg.dll	Kernel32.dll, Advapi32.dll
Pptoolr.dll	Kernel32.dll, User32.dll
Pptoolr.dll	Kernel32.dll, User32.dll
Pptools.dll	Kernel32.dll, User32.dll, Advapi32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll
Presconf.dll	Wsock32.dll, Oleaut32.dll, User32.dll, Kernel32.dll, Advapi32.dll
Ttomb32.dll	User32.dll, Gdi32.dll, Lz32.dll, Kernel32.dll

DLLs Called by Microsoft Schedule+

Executable files and DLLs associated with Schedule+ and its components call DLLs as shown in the following table:

Calling file	DLLs
Schdpl32.exe	Gdi32.dll, Advapi32.dll, Kernel32.dll, User32.dll, Msvcrt20.dll
Datzap32.dll	Kernel32.dll
Mscal32.dll	Msspc32.dll, Shell32.dll, Winmm.dll, Gdi32.dll, Advapi32.dll, Mpr.dll, Kernel32.dll, User32.dll, Msvcrt20.dll, Comctl32.dll
Msscd32.dll	Msspc32.dll, Shell32.dll, Winmm.dll, Gdi32.dll, Advapi32.dll, Mpr.dll, Kernel32.dll, User32.dll, Msvcrt20.dll, Comctl32.dll
Msspc32.dll	Shell32.dll, Winmm.dll, Gdi32.dll, Advapi32.dll, Mpr.dll, Kernel32.dll, User32.dll, Msvcrt20.dll, Comctl32.dll
Mstre32.dll	Msspc32.dll, Shell32.dll, Winmm.dll, Gdi32.dll, Advapi32.dll, Mpr.dll, Kernel32.dll, User32.dll, Msvcrt20.dll, Comctl32.dll
Mstrs32.dll	Msspc32.dll, Mscal32.dll, Shell32.dll, Winmm.dll, Gdi32.dll, Advapi32.dll, Mpr.dll, Kernel32.dll, User32.dll, Msvcrt20.dll, Comctl32.dll

DLLs Called by Microsoft Word

Executable files and DLLs associated with Word and its components call DLLs as shown in the following table:

Calling file	System DLLs
Winword.exe	Wwintl32.dll, Advapi32.dll, Gdi32.dll, Kernel32.dll, Shell32.dll, User32.dll, Ole32.dll, Mso95.dll, Comctl32.dll, Msvcrt20.dll, Rpcrt4.dll
Schdmapr.dll	Oleaut32.dll, Ole32.dll, Msvcrt20.dll, Rpcrt4.dll, Kernel32.dll, User32.dll, Gdi32.dll
Station.dll	Kernel32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Advapi32.dll, Shell32.dll, Ole32.dll, Oleaut32.dll, Mapi32.dll, Comctl32.dll, Msvcrt20.dll, Rpcrt4.dll
Wordmail.dll	Kernel32.dll, Advapi32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Shell32.dll, Mapi32.dll, Mfcans32.dll, Ole32.dll, Mpr.dll, Oleaut32.dll, Comctl32.dll, Msvcrt20.dll, Rpcrt4.dll
Wwintl32.dll	Kernel32.dll, User32.dll

DLLs Called by Shared Applications

Executable files and DLLs associated with shared application tools call DLLs as shown in the following tables:

Shared application	Calling file	DLLs
Art Gallery	Artgalry.exe	Mpr.dll, Picstore.dll, Comctl32.dll, Pubdlg.dll, Kernel32.dll, User32.dll, Gdi32.dll, Condlg32.dll, Advapi32.dll, Shell32.dll, Mfcans32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Oleaut32.dll
	Picstore.dll	Kernel32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, User32.dll
	Pubdlg.dll	Kernel32.dll, User32.dll, Gdi32.dll
Data Access Objects	Odbcad32.exe	Odbc32.dll, Msvcr20.dll, Kernel32.dll, User32.dll, Advapi32.dll, Gdi32.dll, Comdlg32.dll, Lz32.dll, Version.dll
	Dao3032.dll	Msjt3032.dll, Msjter32.dll, Msjint32.dll, Kernel32.dll, User32.dll, Advapi32.dll, Oleaut32.dll, Mpr.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll
	Dbsnpntw.dll	Kernel32.dll, Msvcr20.dll, User32.dll
	Ds32gt.dll	Msvcr20.dll, Kernel32.dll, User32.dll
	Mscpxl32.dll	Msvcr20.dll, Kernel32.dll, User32.dll
	Msjint32.dll	Kernel32.dll, User32.dll
	Msjt3032.dll	Mpr.dll, Kernel32.dll, User32.dll, Advapi32.dll, Ole32.dll, Oleaut32.dll, Msvcr20.dll, Rpcrt4.dll
	Msjter32.dll	Msjint32.dll, Kernel32.dll
	Mslt3032.dll	Msjt3032.dll, Kernel32.dll, User32.dll, Advapi32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Mpr.dll
	Msp3032.dll	Kernel32.dll, User32.dll, Advapi32.dll
	Msr2x32.dll	Msjt3032.dll, Kernel32.dll, User32.dll, Advapi32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Mpr.dll
	Mstx3032.dll	Msjt3032.dll, Kernel32.dll, User32.dll, Advapi32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Mpr.dll
	Msvcr20.dll	User32.dll, Kernel32.dll
	Mswng300.dll	Msjt3032.dll, Msjter32.dll, Msjint32.dll, Mpr.dll, Kernel32.dll, User32.dll, Advapi32.dll, Ole32.dll, Oleaut32.dll, Msvcr20.dll, Rpcrt4.dll
	Mspb3032.dll	Kernel32.dll, User32.dll, Advapi32.dll

Shared application	Calling file	DLLs
Data Access Objects (continued)	Msx13032.dll	Msjt3032.dll, Kernel32.dll, User32.dll, Advapi32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Mpr.dll
	Odbc32.dll	Msvcr20.dll, Kernel32.dll, Advapi32.dll, User32.dll
	Odbc32gt.dll	Msvcr20.dll, Kernel32.dll, User32.dll
	Odbccp32.dll	Msvcr20.dll, Kernel32.dll, Advapi32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Lz32.dll, Version.dll
	Odbccr32.dll	Odbc32.dll, Kernel32.dll, User32.dll, Shell32.dll, Advapi32.dll, Gdi32.dll, Comdlg32.dll, Comctl32.dll, Msvcr20.dll
	Odbcjt32.dll	Odbcji32.dll, Msjt3032.dll, Msjter32.dll, Odbctl32.dll, Odbccp32.dll, Kernel32.dll, Advapi32.dll, User32.dll, Comdlg32.dll, Msvcr20.dll, Gdi32.dll, Mpr.dll, Ole32.dll, Oleaut32.dll, Rpcrt4.dll, Msjint32.dll, Lz32.dll, Version.dll
	Odbctl32.dll	Kernel32.dll, User32.dll, Msvcr20.dll
	Odbbse32.dll	Odbcjt32.dll, Odbcji32.dll, Msjt3032.dll, Msjter32.dll, Odbctl32.dll, Odbccp32.dll, Kernel32.dll, Advapi32.dll, User32.dll, Comdlg32.dll, Msvcr20.dll, Gdi32.dll, Mpr.dll, Ole32.dll, Oleaut32.dll, Rpcrt4.dll, Msjint32.dll, Lz32.dll, Version.dll
	Odex132.dll	Odbcjt32.dll, Odbcji32.dll, Msjt3032.dll, Msjter32.dll, Odbctl32.dll, Odbccp32.dll, Kernel32.dll, Advapi32.dll, User32.dll, Comdlg32.dll, Msvcr20.dll, Gdi32.dll, Mpr.dll, Ole32.dll, Oleaut32.dll, Rpcrt4.dll, Msjint32.dll, Lz32.dll, Version.dll
	Odfcx32.dll	Odbcjt32.dll, Odbcji32.dll, Msjt3032.dll, Msjter32.dll, Odbctl32.dll, Odbccp32.dll, Kernel32.dll, Advapi32.dll, User32.dll, Comdlg32.dll, Msvcr20.dll, Gdi32.dll, Mpr.dll, Ole32.dll, Oleaut32.dll, Rpcrt4.dll, Msjint32.dll, Lz32.dll, Version.dll
	Odpdx32.dll	Odbcjt32.dll, Odbcji32.dll, Msjt3032.dll, Msjter32.dll, Odbctl32.dll, Odbccp32.dll, Kernel32.dll, Advapi32.dll, User32.dll, Comdlg32.dll, Msvcr20.dll, Gdi32.dll, Mpr.dll, Ole32.dll, Oleaut32.dll, Rpcrt4.dll, Msjint32.dll, Lz32.dll, Version.dll

Shared application	Calling file	DLLs
Data Access Objects (continued)	Odtex32.dll	Odbcjt32.dll, Odbcji32.dll, Msjt3032.dll, Msjter32.dll, Odbct132.dll, Odbccp32.dll, Kernel32.dll, Advapi32.dll, User32.dll, Comdlg32.dll, Msvcr20.dll, Gdi32.dll, Mpr.dll, Ole32.dll, Oleaut32.dll, Rpcrt4.dll, Msjint32.dll, Lz32.dll, Version.dll
	Sqlsrv32.dll	Kernel32.dll, Msvcr20.dll, User32.dll, Advapi32.dll, Odbccp32.dll, Version.dll, Netapi32.dll, Gdi32.dll, Comdlg32.dll, Lz32.dll, Netbios.dll
	Vbajet32.dll	Msvcr20.dll, Kernel32.dll
Data Map	Datainst.exe	Mitmdl32.dll, Kernel32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Advapi32.dll, Shell32.dll, Comctl32.dll
	Datamap.exe	Mitmdl32.dll, Kernel32.dll, Gdi32.dll, User32.dll, Comdlg32.dll, Advapi32.dll, Shell32.dll, Comctl32.dll, Mfcans32.dll, Oleaut32.dll, Ole32.dll, Msvcr20.dll, Rpcrt4.dll, Mfcuia32.dll
	Midi10in.dll	Kernel32.dll
	Mitmdl32.dll	Kernel32.dll, Gdi32.dll, User32.dll
	Mo10in32.dll	Kernel32.dll
	Tm10in32.dll	Kernel32.dll
Equation Editor	Eqnedt32.exe	Kernel32.dll, User32.dll, Gdi32.dll, Advapi32.dll, Ole32.dll, Shell32.dll, Comctl32.dll, Msvcr20.dll, Rpcrt4.dll
Graphics Filters	Msbmp32.dll	Kernel32.dll
	Msgif32.dll	Kernel32.dll
	Msjpeg32.dll	Kernel32.dll
	Msjpeg32.dll	Kernel32.dll
	Mspcd32.dll	Kernel32.dll, Pcdlib32.dll, Gdi32.dll, User32.dll
	Mspcx32.dll	Kernel32.dll
	Mstga32.dll	Kernel32.dll
	Mstiff32.dll	Kernel32.dll
	Pcdlib32.dll	Gdi32.dll, Kernel32.dll, User32.dll
Text Converters	Wpequ532.dll	Kernel32.dll
Microsoft Info	Msinfo32.exe	Version.dll, Kernel32.dll, User32.dll, Gdi32.dll, Advapi32.dll, Shell32.dll, Comctl32.dll
	Imgwalk.dll	Kernel32.dll, User32.dll

Shared application	Calling file	DLLs
Microsoft Query	Msqry32.exe	Qryint32.dll, Kernel32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Odbc32.dll, Advapi32.dll, Msvcrt20.dll
	Qryint32.dll	Kernel32.dll
Microsoft Organization Chart	Orgchart.exe	Kernel32.dll, User32.dll, Gdi32.dll, Comdlg32.dll, Advapi32.dll, Ole32.dll, Comctl32.dll, Msvcrt20.dll, Rpcrt4.dll
Proofing Tools	Mssp232.dll	Kernel32.dll, User32.dll
	Msth32.dll	Kernel32.dll, User32.dll
	Gram32.dll	Gdi32.dll, Kernel32.dll, User32.dll
	Hyph32.dll	Kernel32.dll, User32.dll
Microsoft WordArt	Wrdart32.exe	User32.dll, Gdi32.dll, Kernel32.dll, Pubole32.dll, Mfcans32.dll, Ole32.dll, Shell32.dll, Advapi32.dll, Comctl32.dll, Oleaut32.dll, Msvcrt20.dll, Rpcrt4.dll
	Pubole32.dll	User32.dll, Gdi32.dll, Kernel32.dll, Advapi32.dll, Ole32.dll, Msvcrt20.dll, Rpcrt4.dll
Visual Basic for applications	Vba232.dll	Ole32.dll, Oleaut32.dll, Gdi32.dll, User32.dll, Msvcrt20.dll, Kernel32.dll, Advapi32.dll, Winmm.dll, Rpcrt4.dll
	Vba32.dll	Ole32.dll, Oleaut32.dll, Gdi32.dll, User32.dll, Crt.dll, Mpr.dll, Kernel32.dll, Advapi32.dll, Winmm.dll
	Vbar2132.dll	Ole32.dll, Oleaut32.dll, Gdi32.dll, User32.dll, Msvcrt20.dll, Kernel32.dll, Advapi32.dll, Rpcrt4.dll
Help and Cue Cards	Cuecrd32.dll	User32.dll, Kernel32.dll
	Visx.exe	User32.dll, Kernel32.dll
	Hlp95en.dll	Kernel32.dll, User32.dll, Gdi32.dll

Shared Services

Services that Office adds to the individual Office applications are designed to unify common activities such as creating and opening documents, making appointments, or adding tasks or contacts. In this way, users do not necessarily "run Schedule+" or "start Microsoft Excel." Instead, they go straight to a task, and Office opens the tools necessary to accomplish the task.

In particular, the services that Office provides are the Microsoft Office Shortcut Bar, Office Binders, and the Find Fast utility.

Microsoft Office Shortcut Bar

The Microsoft Office Shortcut Bar is an upgraded version of the Microsoft Office Manager from Office 4.x. It is a set of toolbars that users can use to start applications. By default, the Shortcut Bar includes toolbars for Office, Desktop, Programs, Accessories, and the Microsoft Network (MSN) if MSN has been installed on the system.

Users can create their own toolbars, and can include buttons of their choice for creating single-click shortcuts to their most commonly used documents and applications. They can create a shortcut by dropping any file system object (such as files, folders, printers, or the control panel) onto any Shortcut Bar toolbar.

As an administrator, you can also create and distribute custom shortcut toolbars. The toolbars that appear on the Shortcut Bar are *.tbb files stored in `\Msoffice\Office\ShortCut Bar` on the user's computer. The buttons that appear on the Office toolbar by default are shortcut files in `\Msoffice\Office\ShortCut Bar\Office`. You can use the Network Installation Wizard to modify the collections of files in these locations on users' computers. For information on using the Network Installation Wizard, see Chapter 9, "Customizing Client Installations."

The folder location for the toolbar you create must also be specified in the Windows Registry. Each toolbar is listed under its own key under the key `Hkey_Current_User\Software\Microsoft\Office Shortcut Bar\Toolbars`:

```
\Hkey_Current_User\Software\Microsoft\Office Shortcut Bar
  \Position
    \Height
    \Left
    \Top
    \Width
  \Toolbars
    Key: \<path>
    Value: <RGB: 4 bytes> <Buttons> <#Rows: 2 bytes>
```

Each toolbar has a binary value pair where the name of the value is the folder path and the value is set to a null delimited and double null terminated string of the files in that folder and their ordering. Separators can be specified by a hyphen (-) character. The four bytes preceding this list are for the value of the toolbar color. The final two bytes of the value are reserved for the number of rows of buttons last displayed on the toolbar.

For example, if a folder `C:\My Documents\Work` were listed under Toolbars, then its entry might look like this:

```
[key] c:\my documents\work [value]<rgb-4 bytes>my first file()my second
file()my third file()-()my last file()-()<rows-2 bytes>
```

In this case, the toolbar would contain three buttons, then a separator, then the last button; the buttons would correspond to each file and would appear in the order listed. Filenames in the list prefixed with an asterisk (*) are interpreted as hidden, so they will show up in the Customize dialog box, but not on the toolbar.

Office Binders

Office Binders are based on the metaphor of a three-ring binder, where all work that pertains to a single topic can be stored conveniently. Users can store spreadsheets, charts, documents, and presentations together in a binder, where they are saved and opened as a single unit, and can be printed in a single pass, with consecutive page numbering. Users can also disassemble a binder's contents and save each section as its own file.

Binders are OLE containers for embedded document objects. That is, opening a Word document section of a binder activates the Word tools necessary for editing the document, but any changes to the document are not reflected in the source document; likewise any changes to the source document are not reflected in the binder.

For more information about Office Binders, see "Supporting Office Binders" in Chapter 12, "Support and Troubleshooting."

Find Fast Indexer

The Find Fast indexer is a utility provided by Office that allows a user's system to create an index of document files on the local system or on a network share. Find Fast indexes are not seen by users, but are used by Office applications to make file searches faster. Indexes correspond to folders: the name, scope and location of each index matches that of its associated folder.

The indexer and indexes do not greatly tax system resources. The indexer can work in the background as an idle-time process, and indexes are usually less than five percent of the size of the document collection being indexed. For more information about using the Find Fast indexer, see "Finding and Managing Documents within a Workgroup" in Chapter 15.

By default, Setup installs the indexer as the file `Findfast.exe` in the user's `\Msoffice\Office` folder, copies a Find Fast shortcut to the `\Windows\Start Menu\StartUp` folder, and installs Find Fast in the Windows Control Panel. Setup also installs the Find Fast log file, `Ffastlog.txt`, in the user's `\Windows\System` folder. The location of the indexer is specified in the Registry key `Hkey_Local_Machine\Software\Microsoft\Shared Tools Location`, and the location of the log file is specified in the Registry key `Hkey_Local_Machine\Software\Microsoft\Shared Tools\Find Fast\95`.

Users can access the Find Fast indexer by going to the Control Panel and double-clicking on the Find Fast icon. The Find Fast dialog box allows users to create and delete indexes, to force an update to their index, to suspend indexing, and to see information about their index. By default, the Find Fast indexer creates an index of all Office documents, but users can choose to index whatever type of files they want. For example, they can index only PowerPoint presentations (*.ppt and *.pot files), or they can expand the index to include all files (*.*). Users can also choose the location of the index, such as the Msoffice folder or the root folder of their hard drive. When the indexer creates an index, it creates the files __ofidx.ffa, __ofidx.ffl, and __ofidx0.ffx in the folder being indexed. Indexes cover all searchable files of a folder, including subfolders. Indexes are updated automatically by default.

Instead of a root-level index, users may elect to create several smaller indexes for each volume that they want to search. Users can create as many indexes as they want, providing corresponding folders don't overlap. That is, the folder of one index cannot contain the folder of another index. Following are some reasons that multiple indexes might be practical:

- The user has multiple hard drives, such as a C and a D drive.
 - The user may want to index a Windows for Workgroups share. Because the folder for one index cannot be above or below the folder for another index, the user needs to create several indexes to cover all documents of a drive. For example, if the Windows for Workgroups folder is top-level, the user may want to create additional indexes for each other top-level folder.
 - The user creates an index for a share on a NetWare server. It would be unnecessary for the user to create an index for a Windows 95 or NT file server, as these servers should already be running an instance of the indexer, maintaining an index of server files.

How Indexes are Updated

When the Find Fast indexer creates an index, it produces a single master index that is optimized for speed and size. The Find Fast indexer updates and optimizes the master index as follows:

1. Perform a simple update to produce a sorted term list. The list contains one term for each document that is out of date. Documents are out of date if their current last-saved time stamp differs from the time stamp stored in the index.
2. Periodically merge collections of sorted term lists to produce a shadow index. A shadow index is structured the same as a master index, and is usually much smaller and faster than the collection of term lists it replaces.
3. Periodically merge together all shadow indexes and the previous master index to produce a new master index.

The interval between automatic updates is adjusted automatically by the indexer based on past updates, according to the following rules:

- Update interval initially is two hours.
- If more than 10 files are out of date on the current update, then the update interval is shortened by 15 minutes, down to a minimum of 30 minutes. Files added, deleted, or modified since the last update all count as being out of date.
- If 10 or fewer files are out of date on the current update, then the update interval is lengthened by 15 minutes up to a maximum of 24 hours.

Find Fast Indexer Log

As the Find Fast indexer updates indexes, it logs events and errors in a log file, Ffastlog.txt. Users are not alerted to errors and messages from the indexer. To see these messages, users must click Show Indexer Log on the Index menu in the Find Fast dialog box.

Events in the log file are listed so that the most recent events appear at the beginning of the file. The indexer log is truncated when its size exceeds 20K. When this happens, the log file displays a message that the log has been truncated.

CHAPTER 2

Microsoft Access Architecture

The architecture of Microsoft Access consists of the application, the database container or file, and the database objects, such as tables, queries, forms, reports, macros, and modules. The Microsoft Access application stores components such as menus and toolbars, as well as the Microsoft Jet database engine, and any add-ins written in Visual Basic for applications. The database container stores the database objects, and the database objects store the data itself, as well as OLE custom controls and procedures for extracting and reporting the data. This chapter explains the distribution of data, components, and database objects across the architecture of Microsoft Access.

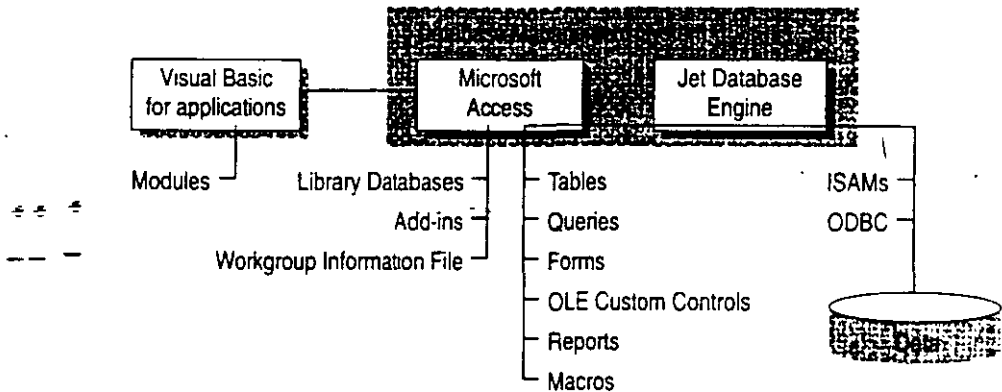
In This Chapter	Application Layer	61
	Database Layer	66
	Database Objects	68
	Where Microsoft Access Components Reside	70
	How Microsoft Access Resolves Conflicts	71

Application Layer

The application layer of Microsoft Access controls the user interface and provides an environment for the Jet database engine. In addition, the application layer makes calls to any *library databases*, which are collections of procedures, and *add-ins*, which are Visual Basic programs designed to accomplish specific tasks. When users are running Microsoft Access in a workgroup, their preference settings and security information are stored in a workgroup information file. This section explains the Jet database engine, library databases, add-ins, and the workgroup information file.

Microsoft Access and the Jet Database Engine

Together, Microsoft Access and the Jet database engine form a complete database management system (DBMS). Microsoft Access is the part of the DBMS that is responsible for the user interface and all the ways that users view, edit, and manipulate data through forms, queries, reports, and so forth. The Jet database engine retrieves data from, and stores data in, user and system databases. It is the data manager component of the DBMS.



Microsoft Access

Microsoft Access is the container for the user interface and for all application code. It includes all the default settings for menus, toolbars, and database objects. It also includes all the library database files that come with Microsoft Access.

Default Settings Users can modify the look and behavior of the Microsoft Access user interface in the Options dialog box. Some of these settings are stored in the Windows Registry, while others are stored in the user's workgroup information file, System.mdw. For information about Microsoft Access entries in the Windows Registry, see Appendix B, "Registry Keys and Values." For information about the settings that are stored in System.mdw, see "Workgroup Information File," later in this chapter.

Library Database Files The library databases that come with Microsoft Access include all the wizards and add-ins, as well as the default System.mdw. For more information about the library databases included with Microsoft Access, see "Library Databases," later in this chapter.

Jet Database Engine

The Jet database engine is a relational database engine that handles all database processing for Microsoft Access and Visual Basic. With Microsoft Access, the Jet database engine can also provide data to Open Database Connectivity (ODBC) client applications.

The Jet database engine is made up of a set of dynamic-link libraries (DLLs):

Jet database engine DLL The Jet database engine DLL (Msjt3032.dll) is the main program that evaluates and executes requests for data. If the request is for *native* data—data stored in the Microsoft Access Database (.mdb) format—then Msjt3032.dll also handles the reading and writing of the data. However, if the request involves non-native data, then Msjt3032.dll makes calls to either the ODBC Driver Manager DLL (Odbc32.dll) or one of the external ISAM DLLs, as explained later in this section.

Data Access Objects DLL The Data Access Objects DLL (Dao3032.dll) is the Jet component that provides a developer interface to the Jet database engine. Data Access Objects (DAO) includes a rich, high-level set of objects that insulates developers from the physical details of reading and writing records.

External ISAM DLLs The Jet database engine provides access to several external Indexed Sequential Access Method (ISAM) format files using a series of installable DLL files. The Jet database engine supports the external ISAM formats shown in the following table.

ISAM format	Supported by DLL
Xbase (dBase and FoxPro)	Msbx3032.dll
Paradox	Mspx3032.dll
Lotus	Mslt3032.dll
Microsoft Excel	Msx13032.dll
Text	Mstx3032.dll

These DLLs handle the reading and writing of data stored in dBase®, FoxPro®, Paradox®, Lotus, Microsoft Excel, and text file formats.

Library Databases

A library database is a collection of procedures that Microsoft Access makes calls to for specific tasks. Library databases contain code only, and they can be functions that Microsoft Access calls as needed. They do not contain database objects such as forms or tables.

To use library databases, Microsoft Access must have a reference to them. Users make this reference by clicking **References** in the **Tools** menu. Library database files have the file extension `.mda`.

Add-ins

Add-ins are Visual Basic programs that add functionality to Microsoft Access. Add-in files, like library database files, have the file extension `.mda`. Users can add or remove add-ins from Microsoft Access by clicking **Add-ins** on the **Tools** menu and then clicking **Add-in Manager**.

Add-ins can automate complex or repetitive tasks in Microsoft Access. You can create or customize add-ins for your users. For information about creating add-ins in Microsoft Access, see Chapter 16 of the Microsoft Access book, *Building Applications*.

Microsoft Access has three kinds of add-ins: wizards, builders, and menu add-ins. Wizards and builders are context-specific; they are called into the application when the user wants to do something, such as build a form, create a table, or make a query. Menu add-ins must be called into the application by the Add-in Manager.

Wizards

Wizards allow users to accomplish tasks quickly and easily, without knowing much about Microsoft Access. A wizard is usually a series of dialog boxes that lead a user, step by step, through a complex procedure. The dialog box format helps to limit the user's choices and provides them a defined sequence to follow.

You can create wizards for your own users. Microsoft Access provides direct support for some types of wizards. For example, if you create a wizard to design a specific type of form, your wizard can be installed to appear in the same list as the Microsoft Access Form Wizard. Microsoft Access supports the following kinds of wizards:

- Table and query wizards
- Form and report wizards
- Property wizards
- Control wizard

Builders

Builders are simpler tools than wizards. Builders usually consist of a single dialog box or form that guides the user through the process of constructing an expression or other single data element. The Microsoft Access Expression Builder is an example of this type of add-in.

Microsoft Access supports many kinds of builders automatically. If you create one of the following kinds of builders, it will appear on the list of Microsoft Access builders:

- Property builders
- Control builders
- Expression builders

Menu Add-ins

Menu add-ins are general-purpose mini-applications that accomplish tasks that are not context-sensitive in the way that wizards and builders are. A menu add-in usually operates on multiple objects or on the Microsoft Access application itself. The Microsoft Access Documentor is an example of a menu add-in.

Menu add-ins are supported by the Add-in command on the Tools menu.

Workgroup Information File

In Microsoft Access installations where user and group level security is being used, the user and group information is stored in the workgroup information file, System.mdw. For all Microsoft Access users, this file also stores user-specific preferences such as custom toolbars and other options. The System.mdw file is required for Microsoft Access to start up.

For users to share data, they join a workgroup by using the Workgroup Administrator, Wrkgadm.exe, to specify the System.mdw file on the network share where they want to run Microsoft Access. If a password has been defined, users are required to log on to this account. The shared System.mdw stores passwords for each user.

The System.mdw a user specifies is the file that Microsoft Access uses every time it starts up, until the user specifies some other System.mdw. If a user does not belong to a workgroup and therefore specifies no System.mdw, Microsoft Access uses the System.mdw on the local workstation, with default security settings. The System.mdw that Microsoft Access uses for any given desktop is specified in the Registry setting
Hkey_Local_Machine\Software\Microsoft\Access\7.0\Jet\3.0\Engines\Jet, as the string value for SystemDB.

Once users have joined a workgroup and established their settings, they should back up their System.mdw. If the file somehow gets corrupted, the user must re-create it. Users must have a System.mdw to run Microsoft Access.

Database Layer

Data is generally stored in tables in the database, but links can be established to tables in other databases, to data in other file formats (such as Microsoft Excel, dBase, and Paradox), and to ODBC data sources, such as SQL Server. These links are stored in the database container and act like native tables.

Installable ISAMs

Data that is stored separately from the main database is accessed through *lookup tables*, which are Index Sequential Access Method (ISAM) files. Databases created in formats other than Microsoft Access—such as FoxPro, dBase, and Paradox—use ISAMs that are non-native to Microsoft Access. To enable Microsoft Access to work with databases created in these other formats Microsoft Access ships with *installable ISAMs* (IISAMs). Using IISAMs, Microsoft Access treats non-native ISAM data very similarly to how it treats native tables.

≡ ≡ ≡

— —

IISAMs in Microsoft Access support the following ISAM file formats:

- Microsoft FoxPro, versions 2.0, 2.5, 2.6, and import/export (no linking) for version 3.0
- dBase III PLUS, dBase IV, and dBase V
- Paradox, versions 3.x, 4.x, and 5.0
- Microsoft Excel, versions 3.0 and 7.0
- Lotus, versions 2.x, 3.x, and 4.0
- Text files (fixed-width and delimited text files)

Microsoft Access supports two methods of access to external ISAMs: linked tables and direct table opening.

Linked Tables

Linking to tables is the preferred method for accessing external data. Linking to a table from a non-native database attaches the non-native database to the user's current Microsoft Access database. Any changes to the table are reflected in the linked database. Using linked tables, Microsoft Access stores connection information by adding an icon in the database window, so users can readily access the external table again.

The linked table method of accessing external data is preferred, because the data is live and up to date, and because the link works whether the data is coming from a table or query. For example, if the table is moved to a SQL server, the application can still access the data by linking to the table.

Direct Table Opening

The Microsoft Access user interface does not allow users to open an external table directly. Microsoft Access opens external tables directly only through Visual Basic. A developer might use this method to get a value quickly from a table that does not need to be accessed very often in the application.

The disadvantage to using this method for accessing external data is that it ties the application to the data source. For example, if the table is moved to a SQL server, the data is no longer ISAM data, and the application will fail when attempting to read tables directly. For more information, see "Separating the Application from the Data," later in this section.

ODBC Connectivity

Microsoft Access allows users to retrieve data from Open Database Connectivity (ODBC) data sources. The ODBC connectivity standard is typically used to connect to server-based database systems. Microsoft Access includes ODBC drivers for Microsoft SQL Server.

-- The ODBC standard can also be used to connect to non-server databases and spreadsheets.

As with ISAM data, Microsoft Access can connect to ODBC data sources by linking or opening tables. Microsoft Access also supports SQL Pass-through (SPT) queries for retrieving ODBC data. By using pass-through queries, users work directly with the tables on the server, rather than with linked tables. SPT queries also allow users to run procedures that are stored on the server. The disadvantage of applications that use SPT queries is that they are not portable. For example, an application written for a SQL server will fail if it attempts to access an Oracle server. Users create SPT queries by clicking SQL Specific on the Query menu and then clicking Pass-Through.

Separating the Application from the Data

Microsoft Access applications can be designed to be separate from the data they store. For an example of this type of architecture, see the Northwind Order Entry Application (Orders.mdb) that Setup installs with Microsoft Access. The forms, queries, reports, and other application features are self-contained within Orders.mdb. The database tables, however, are stored in the Northwind database (Nwind.mdb).

This is the preferred method of developing applications in Microsoft Access. Keeping application and database code separate makes both the application and the data easier to maintain. By keeping them separate, you can upgrade a database to future versions of Microsoft Access independently of the custom application. Keeping the application separate from the database also makes it easier for users with different versions of Microsoft Access to use the same data.

Database Objects

Database objects are stored in the database container, which is a file with the extension .mdb. Users can design new objects or open existing ones to work with their database. Database objects are tables, queries, forms, reports, macros, and modules. Each of these is described below.

Tables Tables are containers for storing data that is related in some particular way. Users can define separate tables for each subject on which they wish to keep data, such as employees, invoices, or inventory. Users can enter and modify data in tables directly or through data entry forms, which are described below.

Queries Users retrieve data from the database with queries. Queries are processed by the Jet database engine. The simplest form of queries are *select queries*, where users identify data by category, range, and boolean operators. Microsoft Access displays the results of select queries in *dynasets*, which are tables that are connected to the live data. They may appear to a user to be static tables, but they are actually a dynamic view of the data. When users open a query in Datasheet view, or open a form or print a report that is based on a query, Microsoft Access creates a dynaset using the information currently stored in the database. This allows users to enter and modify data. The changes they make effect the underlying tables in the database itself.

Forms Forms give users a way of entering data into databases, displaying data to the screen, and printing it. Forms can contain text, graphics, data, color, and OLE custom controls. The appearance of a form is controlled by a form template. For more information, see "Form and Report Templates," later in this chapter.

OLE custom controls OLE custom controls (OCXs) are similar to Visual Basic custom controls (VBXs). They are used to add custom functionality to forms and are stored in forms. Microsoft Access 7.0 includes three OLE custom controls: calendar control, spin box, and outline. Additional controls are available from third parties.

Reports Reports present information from a Microsoft Access database so that it is formatted and organized according to the user's specifications. This formatting and organization is controlled by a report template. For more information, see "Form and Report Templates," later in this chapter.

Macros Macros are actions or a set of actions that have been stored to automate tasks. They are associated with the event property of a form, report, section, or control.

Modules A module is a collection of Visual Basic declarations, statements, and procedures. Users can store related procedures in a single module. Databases can include multiple modules, including global, form, and report modules:

- Global modules can be used throughout the database. Microsoft Access loads them when a user opens a database.
- Form modules are associated with a specific form. Microsoft Access loads them when a user opens the form.
- Report modules are associated with a specific report. Microsoft Access loads them when a user opens the report.

Form and Report Templates

When users create a form or report without using a wizard, Microsoft Access uses a template to define the default characteristics of the form or report. The template determines which sections a form or report will have and defines each section's dimensions. The template also contains all the default property settings for the form.

The default templates for forms and reports are called Normal. However, users can use any existing form or report as a template. Users specify which template they want to use in the Options dialog box. This setting is stored in the workgroup information file, System.mdw, and applies to all databases the user opens or creates. Users can copy or export templates to use them in other databases. Additionally, you can create corporate templates to distribute to your users.

Form and report templates define:

- Whether to include a form or report header and footer.
- Whether to include a page header and footer.
- What the dimensions (height and width) of the sections will be.
- Which control default properties to use. However, templates do not create controls on new forms or reports.

Where Microsoft Access Components Reside

Microsoft Access components are stored with the application or an add-in, on disk, or in the Windows Registry. The following table lists components and where they reside.

Component	Stored
Tables	On disk in the database file (*.mdb)
Queries	On disk in the database file (*.mdb)
Forms	On disk in the database file (*.mdb)
Reports	On disk in the database file (*.mdb)
Macros	On disk in the database file (*.mdb)
Modules	On disk in the database file (*.mdb)
OLE custom controls	In the form or report in which it was created
≠ ≠ ≠ User passwords	On disk in System.mdw
- - - Permissions for security accounts	On disk in the database file (*.mdb)
Custom toolbar settings	On disk in System.mdw
Custom menu settings	On disk in System.mdw
User preferences, set with Options command on Tools menu ¹	In the Windows Registry
AutoCorrect settings	In the Windows Registry
Custom dictionary	On disk in Custom.dic
AutoFormat styles	In the AutoFormat Wizard
Expression Builder	On disk in Utility.mda
AutoDialer	On disk in Utility.mda
Normal template for forms and reports	On disk in Utility.mda
Zoom Box	On disk in Utility.mda
Save Output As	On disk in Utility.mda

¹ In multi-user environments, the workgroup file (System.mdw) also stores user preferences.

Where Security Settings Are Stored

Security settings are stored differently, depending on how a user is securing the database. Microsoft Access provides two methods of securing a database: setting a password for opening the database, or employing user-level security, which provides flexibility for allowing different users different levels of permission for the various objects in a database.

Password Security With password security, users specify a password from the Security command on the Tools menu. This password is stored with the database, in the .mdb file.

User-level Security With user-level security, System.mdw stores user accounts, the passwords used to verify user's identities, and the groups to which users belong. If users leave a group, they can be removed from the group and no longer have permissions.

The actual permissions for the database objects are stored in the database file. These permissions are associated with each object within the database as defined by the security administrator. Permissions can be assigned to individual users and groups.

For more information about security settings, see Security in on-line help, or see "Security Administration" in the Microsoft Access section of Chapter 15, "Using Workgroup Features and Applications with Office."

How Microsoft Access Resolves Conflicts

Conflicts can arise in Microsoft Access when a database is being replicated, or when more than one user attempts to edit the same record of a database, causing a locking conflict.

Replication Conflicts

During replication setup—for example, when a user drags a database into the Windows 95 Briefcase—the Jet database engine requests Globally Unique IDs (GUIDs) from the operating system that are associated with each row of data in the design master. These GUIDs are then copied into any replica databases. If a row changes in the design master or a replica, a counter is incremented for the row. This makes it easy for Microsoft Access to compare the values of rows, detect that a change has been made, and replicate the changed data to other databases. If the same row is changed in more than one database simultaneously, Microsoft Access selects between them based on the following rules:

- Microsoft Access chooses the database that has the highest value for its counter. That is, the row that has changed most often has a higher counter, because the counter is incremented every time a row is changed.
- If both rows have been changed the same number of times (counters are the same on both rows in both databases), Microsoft Access chooses a database randomly, as the replication logic cannot reasonably know which of the two rows has the correct data.

Regardless of how the data is selected, users who submit the data that is not chosen are told by a Conflict Wizard in Microsoft Access that their data was rejected. Users can then resubmit their change or accept the other data.

ng Conflicts

If record locking is set to No Locks, there may be locking conflicts when more than one user attempts to edit or save the same record. Users can set record locking by clicking Options in the Tools menu and selecting the Advanced tab.

The No Locks setting allows more than one user to edit a record simultaneously, but the record is locked during the instant when it is being saved. If two users attempt to save changes to the same record, Microsoft Access displays a message to the second user who tries to save the record. This user can then discard the record, copy the record to the Clipboard, or replace the changes made by the other user. Using this option, it is possible for users to write over one another's changes.

— For more information about record locking, see “Multiuser Settings” in the Microsoft Access section of Chapter 15, “Using Workgroup Features and Applications with Office.”

CHAPTER 3

Microsoft Excel Architecture

This chapter describes the architecture of Microsoft Excel in terms of its two primary elements: the application and the workbook. A section of the chapter lists Microsoft Excel components and where they are stored within Microsoft Excel, another section describes the architecture of the Microsoft Excel mapping feature, and the chapter concludes with an explanation of how Microsoft Excel resolves conflicting settings.

In This Chapter

Overall Structure	73
Where Components Are Stored	79
Microsoft Excel Data Map	84
How Microsoft Excel Resolves Conflicts	88

Overall Structure

The architecture of Microsoft Excel consists of the Microsoft Excel application and the workbook file. The Microsoft Excel application controls the standard menus, toolbars, and default settings for how workbook files are created, opened, and stored. The workbook file is a collection of sheets: worksheets, chart sheets, and macro sheets. The workbook file also stores workbook-level components, such as passwords, styles, and scenarios. To standardize workbooks and the Microsoft Excel application across your organization, you can save workbooks in formats other than a workbook file. You can save workbooks as templates and as add-ins.

Microsoft Excel Application

Microsoft Excel default settings and paths that are established at setup. These features can be customized after installation for a single user, a workgroup, or across an entire organization.

There are several ways to customize the Microsoft Excel application:

- **Through the user interface, in the Options dialog box** Users can customize Microsoft Excel on their desktop, by clicking Options on the Tools menu. The settings they make here are stored in their Windows Registry.
- **In the Windows Registry** For information on Registry settings for Microsoft Excel, see Appendix B, "Registry Keys and Values."
- **With an Add-In** Another way of customizing Microsoft Excel is to create add-ins. An add-in is a hidden, read-only workbook in which the Visual Basic code has been fully compiled from a source workbook. For a particular workgroup's specific tasks, you could create an add-in that runs as a standalone, customized version of Microsoft Excel. Or, you can create add-ins that provide supplemental functionality to your users' version of Microsoft Excel as installed on their system. For information about creating, maintaining, and distributing add-ins, see Chapter 12 of the *Microsoft Excel/Visual Basic for Windows 95 Programmer's Guide*.

— — — In addition to these methods of customizing Microsoft Excel, you can also customize specific features for your users. Some of these customizations are stored in a separate file on the user's system, as explained in the following list of features:

- **Toolbar Settings and Custom AutoFill Lists** When users modify their toolbar settings, Microsoft Excel stores these settings in the file *Username.xlb* in the user's Windows folder, where *Username* is the Windows 95 logon name of the current user. When users create their own AutoFill lists for filling in a range of cells with a series of data, these lists are also stored in *Username.xlb*.

You can distribute a common set of custom lists for AutoFill and sorting, or provide custom toolbars for particular tasks at your site. To do so, create the custom lists and custom toolbars in a workbook. Save the workbook, and place the resulting .xlb file on a network share, where users can merge the customizations into their own workbooks by opening the .xlb file with the Open command on the File menu.

Note When custom toolbars are attached to a module in a workbook, they are not stored in the user's \Windows*Username.xlb* file. Instead, they are stored with the workbook that contains the module.

- **Startup and Alternate Startup Folders** When Microsoft Excel starts, it opens all workbooks, charts, and workspace files stored in the user's Xlstart folder. When a user clicks New on the File menu, the workbook templates listed are those stored in Xlstart. If you want your users to open a workbook automatically when they start Microsoft Excel, move or copy the workbook to this folder. This strategy is especially useful if a workgroup is running Microsoft Excel from a network share, and all users are sharing the same Xlstart folder.

If users want to start Microsoft Excel with workbooks and templates stored somewhere other than Xlstart (such as a network share), they can specify an alternate startup folder. Users specify an alternate startup folder on the General tab in the Options dialog box. This setting is stored in the Windows Registry.

You might want to create an alternate startup folder for your users if they have Microsoft Excel installed on individual workstations, each with individual Xlstart folders. By specifying a single alternate startup folder on a network share, you would have a single folder to maintain for any workbooks, charts, or workspace files you want your users to open when they start Microsoft Excel.

- **Default Startup Workbook and Autotemplates** If users do not specify a startup workbook, Microsoft Excel displays a new, unsaved workbook when they start Microsoft Excel. This startup workbook is derived from the *autotemplate*, a template that opens automatically because it has the reserved filename Book.xlt, and it is stored in the user's Xlstart or alternate startup folder.

You can specify the default font, formatting, and other options for new workbooks for your users by creating a custom autotemplate. For example, you can create a workbook autotemplate that includes customized headers and footers and your company name or any text, formatting, formulas, and macros that your users want as defaults when creating new workbooks.

To create a workbook autotemplate, create or open the workbook you want to use as the autotemplate. Click Save As on the File menu, and save the workbook in the template file format, with the name Book.xlt. Distribute this file to your users' Xlstart or alternate startup folder.

When users click the New button on the Standard toolbar, Microsoft Excel creates a new workbook, based on the autotemplate. The autotemplate also shows up as the Workbook icon on the General tab in the New dialog box when users click New on the File menu or when they click the Start a New Document button on the Microsoft Office Shortcut Bar.

- **Workspace File** If you want your users to be able to open a group of workbooks in one step, you can create a workspace file for them. A *workspace file* contains information about which workbooks to open, their locations, and their size and position on the screen. The workspace file does not contain the workbooks themselves.

To create a workspace file for your users, open all the workbooks you want to include in the workspace. Size and position them as you want them to appear when the users open the file. Then, click **Save Workspace** on the File menu. The default name Microsoft Excel gives to the workspace file is `Resume.xlsx`. You may rename the file if you want.

Distribute this file to your users. If you want the workspace file to open automatically each time users start Microsoft Excel, copy the file to the users' `Xlstart` or alternate startup folder.

Note The workbook names and folder locations are saved with the workspace file. If the files are moved to another location, Microsoft Excel will not be able to find them.

Workbooks

- The workbook in Microsoft Excel is analogous to the document in Word, or the presentation in PowerPoint. The workbook is where the data is stored, in the Microsoft Excel file format. Microsoft Excel workbook files have the extension `.xls`.

Microsoft Excel stores workbook files in the binary interchange file format (BIFF). Workbooks are *compound files*: files that contain a hierarchical system of storages and streams. A *storage* is analogous to a folder, and a *stream* is analogous to a file in a folder. This file format is the OLE implementation of the Structured Storage Model standard.

For more information about the Structured Storage Model, see the *OLE 2 Programmer's Reference, Volume One*, and *Inside OLE*. For detailed information about the BIFF file format, see the *Microsoft Excel Developer's Kit*. All of these books are published by Microsoft Press and are available at your local bookstore.

Templates

A *template* is a special workbook used as a pattern to create other workbooks of the same type. Rather than an `.xls` file extension, templates have the file extension `.xlt`.

You can create templates for workgroups to maintain consistency. For example, you can create a sales report workbook, save it as a template, and distribute it to a workgroup. Users in the workgroup then create weekly sales reports based on the template or insert sheets from template into their workbook.

Settings saved in a template determine the following characteristics of new workbooks based on that template:

- The number and type of sheets in a workbook
- Page formats
- Cell formats
- Row and column styles
- Text, dates, numbers, formulas, and graphics, such as a company name and logo
- Custom menu commands, macros, and toolbars attached to a module sheet in the template

When users open a template, Microsoft Excel opens an untitled, unsaved copy of the template that contains all data, formatting, formulas, macros, styles, scenarios, and so forth that is contained in the template. The original template file remains unchanged.

- To create a template, first create the workbook that includes any text, formats, and formulas your users want, and then click Save As on the File menu. Enter a filename and select the folder, such as a network share, to store the template. In the Save As Type box, select Template. Storing workbook templates in a user's Xlstart or alternate startup folder automatically makes the template available when the user clicks New on the File menu.

Note Although templates in a user's startup folder are automatically available for creating new workbooks, only the template with the reserved name `Book.xlt` stored in the startup folder is an autotemplate. When users start Microsoft Excel, click the New button on the Standard toolbar, or click the Start a New Document button on the Microsoft Office Shortcut Bar, the new workbook is based on the autotemplate.

Add-Ins

In addition to saving workbooks as `.xls` and `.xlt` files, users can also save workbooks as add-ins. Add-Ins compiled from Microsoft Excel workbooks have the file extension `.xla`.

You can create custom add-ins to assemble and distribute custom features that, from the user's point of view, act as if they are built into Microsoft Excel itself. For information about creating, maintaining, and distributing add-ins, see Chapter 12 of the *Microsoft Excel/Visual Basic for Windows 95 Programmer's Guide*, published by Microsoft Press and available from your local bookstore.

Note Add-Ins can also be written in C. When compiled, these add-ins have the file extension .xll (similar to .dll files in Word).

Add-Ins Included with Microsoft Excel

Several add-ins are included with Microsoft Excel, although whether or not they are installed on a user's system depends on what kind of installation was chosen during setup, and on which add-ins were selected during a Custom installation. For information about the components installed for each type of installation, see Appendix C, "List of Installed Components."

Users can install additional add-ins by clicking Add-Ins on the Tools menu.

The following table describes the add-ins included with Microsoft Excel. Their default installation folder is \Excel\Library and its subfolders. Some add-ins require a dynamic link library or compiled C add-in in addition to the .xla file, as indicated in the table.

Add-In	Filename	Description
Analysis ToolPak	Funcres.xla, Procdb.xla, Analys32.xll	Adds financial and engineering functions, and provides tools for performing statistical and engineering analysis.
Analysis ToolPak - VBA	Atpvbaen.xla	Adds Visual Basic functions for Analysis ToolPak.
Microsoft Query ¹	Xlquery.xla	Retrieves data from external database files and tables using Microsoft Query.
ODBC	Xlodbc.xla, Xlodbc32.dll	Adds worksheet and macro functions for retrieving data from external sources with Microsoft Open Database Connectivity (ODBC).
Report Manager	Reports.xla	Prints reports that consist of views and scenarios.
Solver	Solver.xla	Calculates solutions to what-if scenarios based on adjustable cells, constraint cells, and optionally, cells that must be maximized or minimized.
Template Utilities	Tmpltnum.xla	Adds utilities used by Microsoft Excel templates.
Template Wizard with Data Tracking	Wztempl.xla	Creates a template to export worksheet data to a database.
Update Links	Updlink.xla	Updates links to Microsoft Excel 4.0 add-ins to directly access the new built-in functionality.

Add-In	Filename	Description
View Manager	Views.xla	Saves the current window display as a view and lets users apply their own saved views to see their worksheet in different formats.
AutoSave	Autosave.xla	Saves workbook files automatically.

To use Microsoft Query with Microsoft Excel, users must install the Microsoft Query application, the Microsoft Query add-in, the drivers for the types of data they want to retrieve, and the necessary ODBC files. Users can install these files by clicking Add/Remove Office Programs and selecting the Data Access option. For information about Data Access components, see Appendix C, "List of Installed Components."

Where Components Are Stored

Workbook components are stored on sheets of the workbook. These components include all the contents of all data cells (constant values and formulas), cell formatting, charts, macros, add-ins, and security settings. Security settings can be stored for an entire workbook, as well as for individual sheets.

- — — This section lists the various components of a workbook and explains how each is stored within the workbook.

Constant Values of Cells

Constant values stored in cells can be numeric values, including date, time, currency, percentage, or scientific notation, or it can be text. The way Microsoft Excel displays numbers in a cell depends upon the number format for the cell. This value may differ from the actual value Microsoft Excel stores, which is with 15 digits of accuracy. By default, Microsoft Excel makes calculations based on the stored value (full precision), but users can have Microsoft Excel calculate based on displayed values by selecting Precision As Displayed on the Calculations Tab in the Options dialog box.

If users are getting results from their formulas that appear to be wrong, it may be due to the difference in precision between displayed and stored values. For example, if two cells each contain the value 1.007, and a formula adds them in a third cell, the result is 2.014. If all three cells are formatted to display only two decimal places, the displayed calculation, $1.01 + 1.01 = 2.01$, appears to be wrong. Similarly, calculating on displayed values, $1.01 + 1.01 = 2.02$ appears to be correct, but this results in a value that is not as precise as it would be by calculating on stored values.

Caution Once users switch to calculating on displayed values, Microsoft Excel stores all constant values as their displayed values, and full precision values cannot be restored.

Calculating with precision as displayed:

- Affects all worksheets in the active workbook.
- Does not affect numbers in the General format, which are always calculated with full precision.
- Slows calculation because Microsoft Excel must round the numbers as it calculates.

Formulas

Formulas are a sequence of values, cell references, names, functions or operators that produce a new value from existing values. They are part of the data that is stored with the workbook file. Cell references in a function can be relative, absolute, or mixed references in the A1 style, or row-and-column (R1C1) style; or they can be name references.

Users who use the A1, or R1C1, reference style may experience difficulty with formulas if they reposition or delete cells, as these references refer to data by position. A way to avoid this problem is to reference cells by name.

Name References

Users can use names as references to a cell, a group of cells, a value, or a formula. Name references can be accessible to an entire workbook or restricted to a sheet. When restricted to a sheet, the same name, such as "Profit" can be used to define related cells on different sheets in the same workbook. Book-level names, on the other hand, can refer to cells on one worksheet and be used throughout the workbook, eliminating the need to re-create names for each new worksheet or to type worksheet references in formulas.

When users create book-level names, they simply type the name they are defining in the Name box on the formula bar. When they create sheet-level names, however, they must include the name of the sheet, such as "Sheet1!Profit" in the Name box.

Sheet-level names override book-level names when used on the sheet where they are defined. For example, in a workbook where Profit is defined both as a book-level name and a sheet-level name for Sheet1, if a user enters a formula that uses the name Profit on Sheet1, the formula uses the sheet-level name rather than the book level name.

Scenarios

Scenarios are the Microsoft Excel method of allowing users to pose what-if models in their data. To create a what-if model, users specify a set of *changing cells*, or cells that store hypothetical data under a name. This name is stored with the workbook file.

Scenarios can be sheet-level or book-level. Users can merge scenarios among different sheets in the same workbook and among multiple workbooks. To merge scenarios among workbooks, all the workbooks must be open.

When a user merges scenarios, there may be some duplicate names. "Best Case" and "Worst Case" are common examples. In such instances, Microsoft Excel appends additional information to the duplicate scenario names, such as creation date, creator name, or an ordinal number.

Users can protect scenarios by selecting the Prevent Changes and Hide options in the Add Scenario dialog box. For the protection option to take effect, however, the user must also activate protection for the sheet.

Cell Formatting Styles

Cell formatting is stored separately from the cell data, as a collection of format settings that are saved together as a style. Styles can be copied between cells, changed, or deleted, without affecting the data in the cell.

The following table shows the individual format settings determined by styles.

Setting	Determines
Number	Decimal places, separator, inclusion of dollar sign, style for displaying negative number and other options for formatting different kinds of numbers, such as currency, dates, fractions, and so on.
Alignment	Horizontal, vertical, text orientation, and whether or not text wraps in the cell.
Font	Typeface, style, size, special effects, and color of the text in the cell.
Border	Placement and style of the border of the cell.
Pattern	Shading and color of the cell.
Protection	Whether data in the cell is locked or the formula is hidden. This option does not take effect until the worksheet is protected, which the user can do by clicking Protection on the Tools menu.

Styles are saved with the workbook. Users can copy styles from one workbook to another by clicking Merge in the Style dialog box and then selecting the workbook they want to merge styles from. Both the source and destination workbooks must be open. All styles from the source workbook are merged into the destination workbook. If styles in the destination document have names that match styles being merged, the user is prompted to choose whether or not to overwrite existing styles in the destination workbook.

All cells in all of the sheets of a new workbook are initially formatted with the Normal style. Users can change and store the settings for the Normal style, but the change does not affect any other workbooks unless they merge the new Normal style into another workbook or save the workbook with the new Normal style to a template. If the Normal style has been modified in a user's autotemplate, the default formatting for cells in all new workbooks is the modified Normal style.

Note The default font that the Normal style uses is specified by the Standard Font setting on the General tab in the Options dialog box. This setting is stored in the Registry key Hkey_Current_User\Software\Microsoft\Excel\7.0\Microsoft Excel, as the string value for Font.

Charts

Users create charts based on a range of selected cells on a worksheet. Depending upon which option the user specifies, a chart is stored either on the same worksheet as the cells it is linked to, or it exists as an entire sheet within the workbook. Charts are linked dynamically to data on a worksheet. This means that changes to the data are updated in the chart, and changes to a data marker on the chart are reflected in the linked data cells.

Data markers are the chart symbols (dot, bar, area, slice, and so on) that represent a single data point or value originating from a worksheet cell. Users can modify data markers on the following types of charts only: bar, column, line, stacked, pie, doughnut, and xy (scatter) charts. Data markers on 3-D charts cannot be modified.

Text on the chart is also linked to text in worksheet cells. This text appears as data labels, legend entries, and labels for axis tick-marks. Editing text in the worksheet cells affects text in the charts that are linked to the cells. Users can also edit text in charts, but this breaks the link to the cells on the worksheet.

Microsoft Excel stores charts in the workbook file. Charts sheets can be copied into other workbooks, and charts can be copied and pasted into other workbooks and into other Office applications, such as Word documents and PowerPoint presentations.

AutoFormats for Charts

Instead of formatting chart items individually, users can quickly change the look of a chart using AutoFormat. Microsoft Excel stores AutoFormats as charts on separate chart sheets in a workbook with the reserved filename X15galry.xls.

AutoFormats, like cell styles, are a collection of characteristics. Each AutoFormat is based on one of the 14 predefined chart types, and can include a chart subtype, legend, gridline options, data labels, color settings, patterns, and layout. Microsoft Excel includes built-in AutoFormats and also allows users to create custom AutoFormats by selecting User-Defined and clicking Customize in the AutoFormat dialog box.

To create a unified look, you can build a library of custom AutoFormats for workgroups at your site. For example, you can create a series of AutoFormats with a consistent layout and color scheme, designed to be integrated into a PowerPoint presentation. To do this, you create custom AutoFormats in a workbook, and then save the workbook. When you save this file, your custom AutoFormats are stored in Xl5galry.xls.

To make the custom AutoFormats available to your users, do one of the following:

- Copy Xl5galry.xls to each user's Xlstart folder.
- Rename Xl5galry.xls and copy the file to the alternate startup folder defined for each user.

Macros

Macros store Visual Basic code in workbooks, allowing users to automate repetitive or complex tasks. You can create macros for your users, and distribute them in the following ways:

- **As macro sheets** When stored within workbooks, Visual Basic modules are stored on sheets called macro sheets. Users can copy these sheets into their own workbooks.
- **As workbooks** You can create a workbook of custom macros, where each Visual Basic module is its own macro sheet. To make the macro sheets available to users at startup, store the workbook in the users' Xlstart or alternate startup folder. To make the macro sheets automatically available to users at startup, name the file Xl5galry.xls and store it in the users' Xlstart folder.
- **As Add-ins** You can distribute the macros as a standalone, customized version of Microsoft Excel by saving a workbook as an add-in. To automatically open the add-in for users, copy the file to the users' Xlstart or alternate startup folder.

Security

You and your users can secure specific sheets or entire workbooks, according to how you set Protection options in the Protect Sheet or Protect Workbook dialog boxes. Security settings are stored with the workbook file. The only way to change them is to open the workbook file, modify the protection options, then save the file.

You can also assign passwords at the sheet level or workbook level. Passwords enable users to bypass sheet or workbook protections.

At the sheet level, the following items can be protected:

- **Sheet contents** Protects cells on worksheets or on Microsoft Excel 4.0 macro sheets, as well as items in charts.
- **Sheet objects** Protects graphic objects on worksheets and charts from being moved, edited, resized, or deleted.
- **Scenarios** Prevents changes to the definitions of scenarios on a worksheet.

At the workbook level, the following items can be protected:

- **Structure** Protects the structure of a workbook so sheets can't be deleted, moved, hidden, unhidden, or renamed, and new sheets can't be inserted.
- **Windows** Protects windows from being moved, resized, hidden, unhidden, or closed.

Microsoft Excel Data Map

The Microsoft Excel Data Map supplies six map formats for analyzing geographical data: Value Shading, Category Shading, Dot Density, Graduated Symbols, Pie Charts, and Column Charts. These formats and their uses are described in the Data Map Help (available only when the Data Map server is active). The map formats allow users to spot geographical patterns in their data, and to correlate one data item, such as sales in a city, to another, such as a city's population.

Data Map and OLE

Data Map consists of an OLE mini-server (meaning that it can only be invoked from within an OLE application) and a set of maps for use with the Data Map server. Users can insert a Data Map object in any OLE application. In OLE container applications, such as the Office applications, users can edit the object in place using OLE visual editing. Users can also move or copy Data Map objects between OLE applications using OLE drag and drop.

Regardless of the location of a Data Map object, data for the map must come from either a Microsoft Excel worksheet or a Microsoft Access database. A user can create a map in Microsoft Excel and then drag it to a Word document or PowerPoint presentation, but doing so breaks the link to the data. To update the map in Word or PowerPoint, the user must replace it with an updated copy of the map. Creating a map in Microsoft Excel and embedding the Microsoft Excel file in an Office Binder can also break the link to the data.

Data Map Terminology

Term	Refers to	Example
Map Template	A collection of features, such as political boundaries, cities, airports, and highways. Some map templates contain a feature that consists of points for single data items, such as postal codes. These single-point feature sets are also called <i>centroid maps</i> .	US States map template consists of the following features: <ul style="list-style-type: none"> • U. S. state political boundaries • U. S. major cities • U. S. minor cities • U. S. airports • U. S. interstate highways • Centroid map for five-digit ZIP Codes
Features	A set of political boundaries or physical entities, such as cities, which are in a template.	U. S. state political boundaries.
Data Set	Cell data that is mapped to a map template.	Sales by major U. S. cities in the US States map template.
Geographic data	Cell data that corresponds to names of boundaries and other features, such as cities, in a map template.	The cell data "AL" and "AK" corresponds to the boundary names "Alabama" and "Alaska," respectively, in the US States map template.

The GeoDictionary

All Data Map map files are installed in the \Program Files\Common Files\Microsoft Shared\Datamap\Data folder. Data Map keeps a data dictionary of all installed maps in the file Geodict.dct. The location of the geodictionary is set and maintained in the Registry in the key Hkey_local_machine\Software\Microsoft\DataMap\Directories, as the text value for MapData.

For Microsoft Excel to have access to a map, the map must be registered in the geodictionary. Therefore, if a user removes a map file from the disk, the geodictionary will still refer to the file, and the user will get error messages when Microsoft Excel attempts to access the file. Similarly, users cannot add new maps simply by adding map files to their disk. Copying map files to the disk does not register the files in the geodictionary.

To add or remove map files, users must use the Data Installer (Datainst.exe), a utility installed with Data Map, usually in the \Program Files\Common Files\Microsoft Shared\Datamap folder. The Data Installer provides an easy, graphical way to add and remove map files to and from the geodictionary. The geodictionary cannot be edited directly by a user, and must be edited using the Data Installer.

Note Removing a map from the geodictionary does not delete the files from the user's disk; you must delete the files manually.

About the Map Files

Data Map map files use the same format as MapInfo® 3.0 files. Each map consists of four to five files. For example, the World Map consists of the following files:

File	Description
World.dat	Binary file. Stores names for all boundaries or points in the map.
World.id	Binary file. Stores information about relationships with other maps.
World.ind	Binary file. Index used for speeding access to the .map file. Only used in large maps.
World.map	Binary file. Stores latitude and longitude for all points which make up all boundaries in the map.
World.tab	Map descriptor file, used for storing version information, a friendly name for the map, and description of the map feature structure.

How Data Map Works

There are three ways to create a Data Map object in Microsoft Excel:

- With a source data range selected
- Without a range selected, on a blank sheet
- By copying an existing Data Map object

When the Data Map object is created, Microsoft Excel checks for a source data range, which is the user's selection. If Microsoft Excel does not find a source data range, it creates a blank Data Map object. If Microsoft Excel finds a source data range, it sends the range to Data Map. Data Map takes a sample from each column of the source data range and attempts to match the sample against all installed map templates.

Each primary map in a template contains a set of labels in the .dat file. These labels correspond to a set of boundaries (such as "Washington," "Colorado," and so on). In order for the user's data to be displayed on a map, Data Map must be able to match geographic data in the user's selection to labels stored in the .dat file.

Data Map attempts to match over 80 percent of the geographic data against an installed map. If this level is attained, Data Map opens the matched map and begins to plot the data.

In some cases, Data Map can match more than one map. For example, "Germany" matches both the World by Country and the Europe by Country maps. In such cases, users are prompted to choose which of the matched maps they wish to use.

If no maps are matched, Data Map disregards the data and prompts the user to choose a map to display.

If the matching process takes more than five seconds, Data Map aborts auto-matching and prompts the user for a map to use.

Once a map is matched, Data Map displays the map and binds the user's data to it. It is possible that Data Map will not recognize some geographic data, either because the data is misspelled or because the user used an unfamiliar variation of a region's name, such as "Mainland China" instead of "People's Republic of China." Data Map displays all unknown labels one by one to help users find an acceptable alternative. The file Mapstats.xls, stored in the \Program Files\Common Files\Microsoft Shared\Datamap\Data folder, contains lists of proper labels for all installed maps, as well as demographic data to compare with your own data.

By default, Data Map plots values in the first column to the right of the geographic data as a Value Shaded format on the map. If Data Map finds multiple records that pertain to the same region, the values are aggregated. These values are usually added, but it is also possible to count and average them.

Data Map can map data on only one template and one map at a time. If a user's selection has labels for regions in more than one map, Data Map matches the greatest amount of geographic data to one map, and reads geographic data that does not match that map as a mismatch. For example, suppose a user's selection includes several U. S. states and the country Japan. Data Map matches the geographic data to the US by State map, but comes up with a mismatch for the label Japan.

Once users have created a Data Map based on a template, they can import data to other maps in the template. For example, suppose the active template is North America. This template contains three maps: US States, Canadian Provinces, and States in Mexico. The user can select geographical data for one of these maps (US States, for example) to create the North America Data Map object, but then the user must import data separately for the Canadian Provinces map and for the States in Mexico map. Data Map cannot map the user's original selection with data for multiple maps; it must map them one at a time.

How Microsoft Excel Resolves Conflicts

Conflicts can arise in Microsoft Excel when files of the same name reside in a user's startup and alternate startup folders, or when the user interface has been customized through macros or Visual Basic code.

Startup and Alternate Startup Folders

Microsoft Excel opens files in a user's Xlstart folder before opening files in the alternate startup folder. If there is a file in the alternate startup folder with a name that matches a file in the user's Xlstart folder, the file in the alternate startup folder is ignored. For example, if Xl5galry.xls exists in both the Xlstart and the alternate startup folder, only the custom AutoFormats defined in \Xlstart\Xl5galry.xls are available to the user.

Customized Toolbars

There are two ways to create customized toolbars in Microsoft Excel.

- **With the Customize command** Toolbars customized with the Customize button in the Toolbars dialog box are saved in the file *Username.xlsb* when the user quits Microsoft Excel. If the user is not logged on, the filename is Excel.xlsb
- **With a Visual Basic module** Toolbars attached to Visual Basic modules are stored in the workbook file.

When a user opens a workbook that has a toolbar attached, the attached Toolbar definitions are saved to the user's .xlb file. These definitions will exist on the user's computer after the user closes or even deletes the workbook. If custom macros are attached to the toolbar and the user tries to implement the toolbar after closing or deleting the original workbook, the user will get an error message stating that Microsoft Excel cannot find the macro. This is because the macro resides with the workbook where the toolbar originated. If users edit the toolbar, their changes do not affect the original toolbar stored with the workbook. When the user reopens the original workbook, Microsoft Excel uses the copy of the toolbar stored in the user's .xlb file rather than reloading the toolbar stored with the workbook. To generate the original version of the toolbar, users can delete the edited copy by selecting it and clicking Delete in the Toolbars dialog box.

For information about modifying Microsoft Excel toolbars with Visual Basic, see Chapter 9 of the *Microsoft Excel/Visual Basic for Windows 95 Programmer's Guide*.

Customized Add-Ins

If an add-in's source workbook contains a reference to another add-in—for instance, a reference to Xlodbx.xla—make sure that when you distribute the add-ins, the referenced add-in is stored somewhere on the user's computer where the calling add-in can find it. You should always store a referenced add-in in the folder that contains the calling add-in. When you compile a source workbook to an add-in, Microsoft Excel stores a hard-coded path to the referenced add-in in the calling add-in. If you move the calling add-in to another computer (for example, when you distribute the add-in to your users) and Microsoft Excel cannot find the file in the hard-coded location, it searches in the following locations, in this order:

- The folder that contains the calling add-in
- The System folder in the Windows folder
- The Windows folder
- The path defined by the Path environment setting (to see the path, type set in an MS-DOS® window)

For information about creating, maintaining, and distributing add-ins, see Chapter 12 of the *Microsoft Excel/Visual Basic for Windows 95 Programmer's Guide*.

CHAPTER 4

Microsoft PowerPoint Architecture



PowerPoint architecture consists of three layers: the application, a template, and the presentation. The application provides the default user interface for creating presentations: menus, commands, toolbars, and tools. Templates provide a model for creating new presentations, and contain *master* slides. Master slides govern the default look of every slide in a particular presentation. The presentation file contains the master information, as well as the presentation content, including text, graphics, formatting, multimedia content, and settings such as slide setup and view settings for that particular presentation.

In This Chapter	Application Layer	91
	Understanding Templates	92
	Presentation Layer	95
	How PowerPoint Resolves Conflicts	95

Application Layer

The application layer in PowerPoint provides the default settings for users to create presentations. Unlike other Office applications, such as Word and Microsoft Excel, users cannot modify the PowerPoint application with macros or Visual Basic modules.

As in Microsoft Excel, users can customize their toolbar settings. And, as in all Office applications, users can set preferences in the Options dialog box. These settings are stored in the Windows Registry. For information on PowerPoint entries in the Windows Registry, see Appendix B, "Registry Keys and Values."

Understanding Templates

The key to understanding PowerPoint architecture is understanding PowerPoint templates. PowerPoint has two kinds of templates:

- Presentation templates
- Design templates

Presentation Templates

Presentation templates are presentations that serve as a pattern for creating new presentations. By basing many presentations on a single presentation template, users can quickly create standard presentations with consistent formatting, graphic elements, and standard text.

For example, a standard company presentation template can save users time by including the company logo, providing a standard slide structure, providing layout for speakers notes and handouts, and providing the basic slide design using standard company colors, fonts, and boilerplate text. With part of the work already done for them, users simply fill in the specific information.

Presentation templates can provide the following:

- **Master elements** Text, graphics, and other elements that appear on each slide of the presentation. These elements include standard formatting, such as fonts and sizes, bullets, styles, slide backgrounds, and page orientation.
- **Boilerplate elements** Text, graphics, and other elements such as sound clips, graphs, or charts embedded onto particular slides (rather than appearing on each slide) of the presentation.

These templates are available to users when they specify Start A New Document on the Microsoft Office Shortcut Bar, or when they click New on the PowerPoint File menu. By default, these templates are .pot files stored in `\Msoffice\Templates\Presentations`. When users run the AutoContent Wizard, they are accessing these templates.

Design Templates

Design templates contain no textual content and no actual slides. Rather, they supply the look of the presentation. They are what is known as Templates in PowerPoint 4.0. Typically, design templates control the color scheme, font selection, bullet style, and overall look of the slides in a presentation.

Design templates consist of master elements only, as defined on the Slide Master and the Title Master. Users can edit design templates, by opening them directly and using Slide Master view. Slide and Title masters are discussed later in this chapter.

These templates are available to users when they click Apply Design Template on the Format menu. When applied to existing presentations, the master elements in the new design template replace the master elements in the presentation. Design templates are .pot files stored in \Msoffice\Templates\Presentation Designs.

Templates Applied to Presentations

In PowerPoint, all presentations are based on a template. When a user creates a new presentation, it initially has the characteristics of the selected presentation or design template. Users can create a new presentation in several ways.

If the user	Available templates are
Clicks New on the File menu in PowerPoint	All presentation and design templates in \Msoffice\Templates folder and subfolders.
Clicks Start A New Document on the Office Shortcut Bar	All Office templates as defined in the User and Workgroup templates locations for the Office Shortcut Bar. PowerPoint presentation templates are on the Presentations tab. Design templates are on the Presentation Designs tab.
Clicks the New button on the Standard toolbar in PowerPoint	Blank design template. The New dialog box does not appear.

The default blank design template is Blank Presentation.pot, which contains default settings for layout, font, and color scheme. Because it is a design template and not a presentation template, it contains no text, only placeholders for text and graphics.

Users can modify a presentation at any time by applying a different design template to an open presentation. The Apply Design Template dialog box allows users to apply a presentation template or a design template to an open presentation. When users apply a presentation template, it is only the master elements that are copied to the presentation. Any text or graphics in the presentation template are not copied.

Master Elements

The master elements defined for slides in a presentation are known collectively as the Slide Master. Users can also define a second type of master, the Title Master, whose default settings are a copy of the Slide Master. Design templates are specialized templates in that they contain *only* master elements.

Slide Masters A Slide Master is the main master for a presentation. Users can edit Slide Masters in master view to control the default placement and style of elements on every slide of the presentation. When users create their presentation, however, they can choose which master elements a slide adheres to or deviates from, and these slides will be exceptions to the master. Slides that are exceptions to the master are not updated if the user defines a new master or applies a new design template.

Title Masters A Title Master is another kind of master that users can define for their presentations. This allows for two consistent yet independent looks in a presentation: one for title slides (opening slide, section dividers, and ending slide, for example), and one for slides that make up the body of the presentation. By default, the characteristics of the Title Master are based on the characteristics of the Slide Master. Users can edit these characteristics by modifying a title slide in Title Master view.

Note The Blank Presentation.pot design template has only a Slide Master defined. Users can create a customized, unique Title Master based on the Slide Master by clicking New Title Master. All other design templates contain both Slide and Title masters.

Predefined Slide Layouts

When users create a new presentation that is based only on a design template, PowerPoint displays the New Slide dialog box, where users choose an AutoLayout for the first slide in their presentation. AutoLayouts are predefined slide layouts that users can apply to any slide in their presentation. The default AutoLayout for the first slide is the Title Slide layout, as shown in the following example.

Default layout for the first slide of a new presentation is the Title Slide AutoLayout.



Throughout a presentation, slides that have the Title Slide AutoLayout applied to them are identified as title slides, and their default characteristics are those of the Title Master, which users can modify separately from the Slide Master.

Handout and Notes Masters

Users can also define masters for handouts and for speaker's notes. Design templates do not affect the look of Handout and Notes masters. These masters are stored with the presentation file.

Note The Outline Master, which was available in PowerPoint 4, has been removed for PowerPoint 7.0. Printed outlines follow the format of the Handout Master.

Presentation Layer

- The presentation layer is comparable to the document layer of Word. When users create a presentation, all the content and settings associated with the presentation template and design template are copied to the presentation file. For example, changes to a Slide Master in a presentation based on a design template affect the active presentation only, not the design template file. Presentation files store Slide and Title masters, as well as all content that is an exception to the master slides. The presentation file also stores Handout Masters and Notes Masters.

PowerPoint stores presentation files with the file extension .ppt.

How PowerPoint Resolves Conflicts

A slide in a presentation may seem to have many influences converging upon it: presentation templates, design templates, Slide Masters, Title Masters, and layout that a user has specified for a particular slide. In the case of conflicting settings, the setting closest to the slide takes precedence. For example, if a slide's Slide Master is formatted with text in the Arial font on a blue background, but the user specified Times Roman text on a red background for this slide, the user's specifications take precedence over the Slide Master. PowerPoint resolves conflicts in the following order:

1. Settings defined for an individual slide (exceptions to the master)
2. A slide's AutoLayout (slides whose AutoLayout is Title Slide adhere to Title Master settings)
3. Slide Master and Title Master settings
4. Presentation template
5. Design template

There are two ways PowerPoint creates a new presentation:

From a design template In this case, PowerPoint copies all information from the Design Template including Slide and Title master information to the presentation file. From that point on, everything that is added to or modified in the presentation is stored in the presentation file. Changes to the master elements do not affect the Design Template.

From a presentation template In this case, PowerPoint copies all information from the presentation template, including text, graphics, and custom layout, as well as masters for slides, handouts, and notes, to the new presentation file. All modifications are stored in the presentation file. Changes to boilerplate text, master elements, or any other properties of the file do not affect the presentation template.

Slide Setup Default Settings



PowerPoint stores settings such as slide sizing and the page orientation of slides, notes, handouts, and outlines with the presentation file. A new presentation based on this presentation inherits these settings. These settings are the default settings for the layout of a presentation. Users specify them by clicking Slide Setup on the File menu.

CHAPTER 5**Microsoft Schedule+ Architecture**

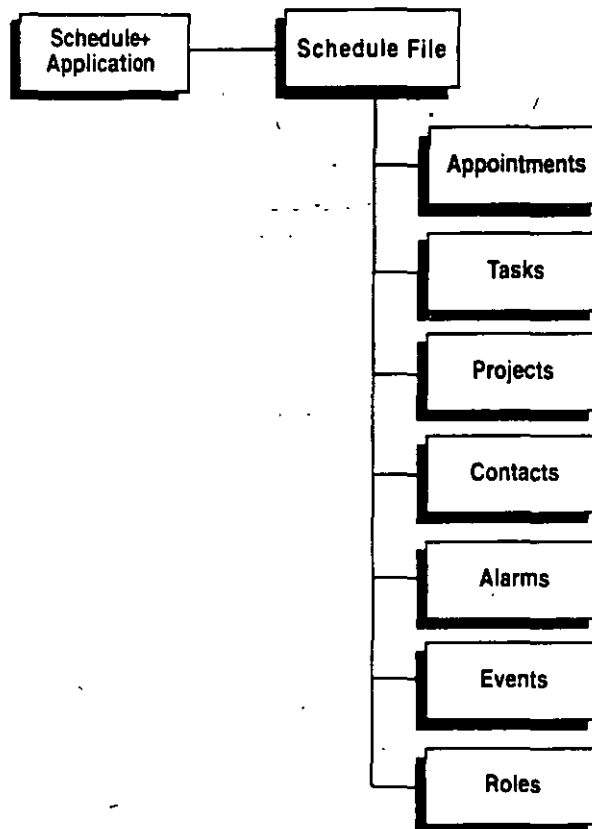
The architecture of Schedule+ is dependent on the existence of a version 1.0 Messaging Application Programming Interface (MAPI 1.0) on the local computer. MAPI 1.0 is available by installing the Microsoft Exchange client from Windows 95. If users have MAPI 1.0 installed, they can run Schedule+ in either stand-alone mode or group-enabled mode. If users do not have MAPI 1.0 installed on their computer, they can run Schedule+ in stand-alone mode only. This chapter provides information on how the components of Schedule+ fit together, explains the differences between running Schedule+ in stand-alone or group-enabled mode, and describes how Schedule+ works with the Microsoft Exchange Server.

In This Chapter

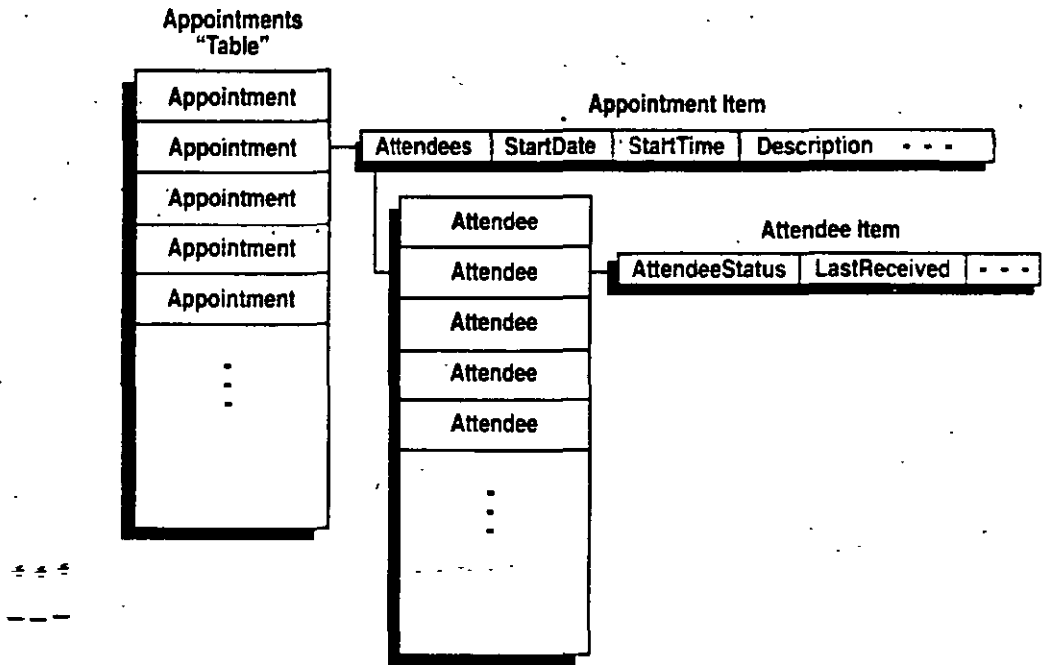
Schedule+ Components	98
Running Schedule+ in Stand-Alone Mode	105
Running Schedule+ in Group-Enabled Mode	105
How Schedule+ Resolves Conflicts	107
How Schedule+ Works with Microsoft Exchange Server	108

Schedule+ Components

On the user's system, Schedule+ consists of the Schedule+ application, the schedule file, and all the schedule items. At the top of the hierarchy is the Schedule+ application, which manages any number of schedule files. Each schedule file contains many items, such as appointments, tasks, contacts, and projects, as shown in the following illustration. In most cases, there is one schedule file for each user, although users can create multiple schedule files for themselves if they want.



Schedule items are like tables that can be further broken down by property. They all have a structure similar to that shown in the following illustration, where appointment items have properties such as attendees, starting date, starting time, and description, and attendee items have properties such as an attendee's free/busy status and last-received invitation.



Schedule+ Application

The Schedule+ application consists of the Schedule+ program file Schdpl32.exe and the Schedule+ DLLs. Additional files types installed by Setup for running Schedule+ are listed in the following table:

File extension	Description
.prt	Print layout files
.fmt	Paper format description files
.hlp	Online Help files
.ico	Icon files
.ctg	Mail forms configuration files

Users can customize Schedule+ default settings by clicking Options on the Tools menu. These settings are stored in the Windows Registry. For information about Registry settings for Schedule+, see Appendix B, "Registry Keys and Values." Other ways in which users can customize their Schedule+ application include the following:

- **Time and date settings** Schedule+ uses time and date settings that are controlled in Windows, under Regional Settings in the Control Panel. Users can customize these settings in Schedule+ and can specify a secondary time zone to use when scheduling meetings across time zones. These custom settings are stored in the Windows Registry.
- **View settings, such as the Tab Gallery or Columns settings** Users configure these settings by using the commands on the View menu, and the settings are stored in *Username.vue*, where *Username* matches the user's logon name. This file also contains the list of most recently used schedule files. You cannot edit this file directly, as it is binary. However, you can rename it, causing Schedule+ to create a new *Username.vue* file with default settings the next time the user starts Schedule+.
- **OLE automation** The Schedule+ application can be extended with OLE automation, using an OLE IDispatch interface. For information about developing software that accesses Schedule+, see the documentation that comes with the *Microsoft Solutions Development Kit* or the Microsoft Developer's Network (MSDN). For information about accessing MSDN, see Appendix F, "Resources."

Filters for Importing and Exporting Data

The Schedule+ application includes filters for exporting and importing task, calendar, and contact information to and from other personal information managers, such as Lotus Organizer time management software, Sharp 9600 personal digital assistant, and Timex Data Link watch.

The following table shows the personal information management systems supported by Schedule+ filters. Currently, only the Sharp digital assistants are supported. Filters for the rest of the personal information managers listed will be available free of charge through Microsoft support resources. For information about these resources, see Appendix F.

In addition to the filters listed here, IntelliLink Corporation supplies a translator package that will be available with future updates to Schedule+. In the meantime, you can call IntelliLink at (603) 888-0666 for information on obtaining the translator package and the converters it contains.

Personal Information Manager / File Format	Import/Export
Schedule+ 1.0	Import
ACT! for Windows	Import
ECCO Professional	Import
Lotus Organizer	Import
PackRat 4.1+	Import
Windows Accessories	Import
Sharp 9600	Import/Export
HP 95LX	Import/Export
Sharp 5000/7000/7200	Import/Export
Sharp 7600/7620	Import/Export
Sharp 8000/8200	Import/Export
Sharp 8600/YO-610	Import/Export
Timex Data Link Watch	Export
≠ ≠ ≠ -- -- CSV (comma-separated value file format)	Import/Export
ASCII format	Import/Export
Microsoft Schedule+ 1.x	Import/Export
Windows Cardfile and Calendar	Import/Export

The Text Import/Export Wizard allows information to be exchanged with other applications using a comma-delimited (CSV) format, a standard method for transferring text-based information. For example, this provides an easy way to import contact information into Schedule+ for users who currently store their contact lists in Microsoft Excel or Microsoft Access.

Users can also insert contact information into Microsoft Word documents by using the Address Book tool in Word. Word imports the contact information by reading the Schedule+ contact list using OLE automation. In addition to accessing address information for a specific Schedule+ contact through the Address Book, Word can also integrate the entire Schedule+ contact list into a mail merge.

Schedule File

The Schedule file (.scd) contains all appointments, projects, tasks, alarms, events, contacts, and roles for a single user. The first time a user runs Schedule+, Schedule+ creates a schedule file on the user's local drive. If the user is running Schedule+ in group-enabled mode, this file is copied to the user's server after the user has been running Schedule+ for 15 minutes. By default, Schedule+ accesses the local file rather than the user's server file, keeping network traffic to a minimum.

Schedule Items

Items are stored in the user's schedule file, as described in the preceding section. Schedule items are appointments, tasks, projects, contacts, alarms, events, and roles.

Appointments Appointments in Schedule+ are activities that occur within a specific time period. Users create appointments in the Schedule+ Appointment Book, which is visible on the Daily, Weekly, Monthly, or Yearly tab. Appointments can have reminders attached to them, and they can be made to be private, tentative, and recurring. A private appointment is hidden to all users except those to whom a user has granted Owner or Delegate Owner access permission. A tentative appointment does not show up as a busy time slot for a user when others view the user's schedule in their Planner. In other words, when a user schedules a tentative appointment for themselves, they can still be considered free to schedule other appointments. A recurring appointment occurs at regular intervals, such as 10 A.M. every Thursday.

If users are running Schedule+ in group-enabled mode, they can schedule meetings by inviting others to their appointments. The list of possible attendees that appears in the user's Meeting Attendees dialog box comes from the user's address list in the workgroup postoffice, or from the address list supplied by the mail server if your worksite is running Microsoft Mail 3.x.

Tasks Tasks are items on a user's To Do list. They can be part of a project, or they can stand alone. Users can assign priorities to their tasks, and they can schedule an end date and a duration for each task.

Projects Projects are groups of tasks. Users can create projects to organize their tasks, although tasks do not need to be associated with a project. Users can assign priorities to projects without affecting the priorities assigned to individual tasks within the projects. Users can designate any project as private, which hides all tasks within the project so that only those users granted Owner or Delegate Owner access permission can view the tasks.

Contacts Contacts are people who users keep track of in a Contact List in their schedule file. The Contact List stores information such as addresses, phone numbers, and user-defined notes. This list is not connected to a user's postoffice address list. Users can assign access permissions to allow other users to view or modify their contact information. Contacts designated as private are visible only to users granted Owner or Delegate Owner access permission.

Alarms Alarms are reminders that users can set for appointments and tasks. The reminder is a message box that appears at a specified interval before an appointment or at the start of an appointed day if the reminder is associated with a task. Reminders can be accompanied by sound if the user sets this option. The default sound is provided by the file Msremind.wav, but users can use the Control Panel to change this sound to any .wav file on the system. Users can set reminders automatically for each new appointment or task by clicking Options on the Tools menu and then clicking the Defaults tab. They can also set the default reminder time on the Defaults tab. Users can change reminder settings for individual appointments and tasks without affecting their default settings.

Events Events are activities or special occasions that occur on a specific day but do not fill a block of time on a user's schedule. Examples of events include birthdays, conferences, and trade shows. Users can designate events as annual if the event occurs on the same day every year. Events and annual events do not appear as busy times in the Planner.

Roles Roles are predefined combinations of access permissions. Users can assign roles to members of their workgroup by clicking Set Access Permissions on the Tools menu. Roles grant access to appointments, contacts, events, and tasks in a user's schedule, as shown in the following table.

Role	Access permissions
None	Cannot read or modify user's schedule. Question mark appears next to user's name in Invite pane of the Schedule+ Planner for free/busy times.
Read	Can read schedule, except for items designated as private.
Create	Can read schedule except for items designated as private; can create new schedule items.
Modify	Can read schedule and can modify existing items, except for items designated as private.
Delegate	Can read schedule and modify existing items, including items designated as private. Can send and receive meeting messages on user's behalf.
Owner	Can read schedule and modify existing items, including items designated as private. Can change access permissions for user's schedule. Can send and receive meeting messages on user's behalf.
Delegate Owner	Can read schedule and modify existing items, including items designated as private. Can change access permissions for user's schedule.
Custom	User sets read, create, and modify permissions separately for appointments, contacts, events, and tasks.

Server Files

When users run Schedule+ in group-enabled mode, Schedule+ creates files on their workgroup server to allow group scheduling. These files include the user's server schedule file, free/busy file, key file, and server files associated with the mail server your worksite uses to support Schedule+.

Server Schedule File Schedule+ creates the user's server schedule file the first time it synchronizes the user's local .scd file with the server. This initial synchronization creates a copy of the user's local schedule file in the Cal subfolder of the folder that contains the user's postoffice. The name of the file is *hex-ID.scd*, where *hex-ID* is a hexadecimal number that is incremented according to the last entry in the Schedul2.key file, which is also stored in the Cal folder.

Free/Busy Files Files with the extension .pof contain a user's free/busy information. Schedule+ stores these files in the Cal folder of the user's workgroup postoffice. A single .pof file stores free/busy information for the entire workgroup. If your worksite is running Microsoft Mail 3.x with Schedule+, there will be one .pof file for each postoffice participating in schedule distribution. These additional files contain Free/Busy information for users in other postoffices. All users in the workgroup postoffice can be invited to meetings, whether or not they are running Schedule+, and regardless of whether they are running Schedule+ in stand-alone or group-enabled mode. If users are not running Schedule+ or are running Schedule+ in stand-alone mode, a question mark appears next to their name in the Invite pane of the Schedule+ Planner view.

Key File Schedule+ creates .key files in the Cal folder of the workgroup postoffice. These files act as a maps to identify which server schedule file belongs to which user in the workgroup. These files contain each user's mailbox name and an eight-digit hexadecimal ID that corresponds to the user's schedule file. Users running Schedule+ 1.0 store their schedules in .cal files, which are mapped by Schedule.key. Users running Schedule+ 7.0 store their schedules in .scd files, which are mapped by Schedul2.key. You can use the utility Dumpkey.exe to display the contents of both .key files.

Mail Server Files Depending on the mail server you are using at your worksite to support Schedule+, such as Microsoft Mail 3.x, you may need to install and set up an administration program, a schedule distribution program, and other utilities. For information about how to set up this software, see the documentation that comes with your mail server.

Running Schedule+ in Stand-Alone Mode

Users can run Schedule+ in stand-alone mode whether or not their system is set up with a postoffice. If a user's system is mail enabled—which means it has a postoffice—Schedule+ asks the user at startup whether the user wants to work in group-enabled mode or alone. If a user's system is not mail enabled, Schedule+ always runs in stand-alone mode.

When users run Schedule+ in stand-alone mode, they can use all the time management features of Schedule+, create contact lists, and manage projects. However, they cannot share any of this information with others in their workgroup nor can they use Schedule+ to schedule meetings with other people, until they run Schedule+ in group-enabled mode.

The first time users run Schedule+ in stand-alone mode, Schedule+ prompts them for whether they want to create a new schedule file or to use an existing one. If users have been working group-enabled, they should use their existing .scd file (or .cal file if their schedule file is in Schedule+ 1.0 format), which is stored in the \Msoffice\Schedule folder by default.

If users choose to create a new schedule file, Schedule+ presents a dialog box for the user to supply a name and location for the file. The default location is the \Msoffice\Schedule folder, and the default name is the user's logon name. If the user's system is mail enabled, the default name of the schedule file is the name of the user's mail profile, as determined by Microsoft Exchange. For more information about creating new profiles in Microsoft Exchange, see "Profiles" in the Microsoft Exchange online help.

Running Schedule+ in Group-Enabled Mode

Before a user can run Schedule+, MAPI 1.0 must be available on the local workstation. Also, the user must have a valid mail account on one of the following platforms:

- **Windows 95 Workgroup Postoffice** This postoffice is included with Windows 95. It must be accessible by all users in the workgroup. For more information about setting up a Windows 95 workgroup postoffice and configuring the Microsoft Exchange client, see "Using the Microsoft Exchange Client with Microsoft Mail" in Chapter 26 of the *Microsoft Windows 95 Resource Kit*.
- **Microsoft Mail Server** Microsoft Mail 3.x supplies a Schedule+ transport that allows users to share schedules and invite one another to meetings across workgroups. For information about configuring the Microsoft Mail server to run with Schedule+, see the documentation that comes with your server.

- **Microsoft Exchange Server** The Microsoft Exchange Server includes MAPI drivers that extend the capabilities of the Microsoft Exchange client by providing access to public folders, creating custom forms and shared documents, scheduling meetings with others, managing time and tasks among workgroups, and creating custom electronic forms. For more information about configuring the Microsoft Exchange Server and setting up public folders, see the documentation that comes with the Microsoft Exchange Server.

When users run Schedule+ in group-enabled mode, they can share the following kinds of information between Schedule+ and other messaging or scheduling applications:

- **Meeting requests** When users make an appointment and invite others, Microsoft Exchange provides a form for sending the meeting request through electronic mail.
- **Free/busy information** On the Planner tab in Schedule+, a meeting organizer can add specific people to a list of invitees, and then check their availability before scheduling a meeting. Their busy times show up as colored bars in the Planner.
- **Details** If an invitee has given the meeting organizer permission to view the details of the invitee's schedule, the meeting organizer can use the right mouse button to click on a busy time for that person and view the details of an appointment, such as location, purpose of the meeting, and so forth. Details also include access permissions users can give each other to specific parts of their schedule, such as contacts or tasks.

The first time users run Schedule+ in group-enabled mode, Schedule+ creates a schedule file on the user's local drive, as described earlier for running Schedule+ in stand-alone mode. At a default interval of 15 minutes, Schedule+ copies the user's local schedule files to the server, in the Cal folder of the user's workgroup postoffice folder. Schedule+ synchronizes users' local and server schedule files at regular intervals. Users can configure the synchronization interval by clicking Options on the Tools menu and then clicking the Synchronize tab. Schedule+ also synchronizes schedule files when the user quits Schedule+, and when the user synchronizes the files manually from the Synchronize tab in the Options dialog box.

Note If users have been running Schedule+ in stand-alone mode, Schedule+ synchronizes their local and server schedule files the next time they run Schedule+ in group-enabled mode.

During users' initial Schedule+ session, their free and busy times are not available on the server until the local schedule file is copied to the server. These users can still be invited to meetings, because their names are in the workgroup's postoffice address book.

Running Schedule+ in group-enabled mode allows users to set access permissions on their schedule, allowing certain users to Read, Create, or Modify appointments and tasks. Users set access permissions by clicking Set Access Permission on the Tools menu. By granting create permission for appointments, users can open each other's schedules and create appointments as needed.

You can grant individuals various combinations of permissions based on the needs at your worksite. For more information about access permissions, see the Schedule+ online help.

When users invite attendees to their meetings, Schedule+ runs the users' mail software automatically. However, when users run Schedule+ in group-enabled mode, Schedule+ adds a button to the standard toolbar with the Microsoft Exchange Inbox icon. This button allows users to run Microsoft Exchange manually from Schedule+, making it easy to correspond with other invitees after reading a meeting request.

How Schedule+ Resolves Conflicts

Conflicts can arise in Schedule+ when the local and server schedule files are being synchronized, when backup server files are created, or during file recovery.

File Synchronization

Schedule+ synchronizes a user's local and server schedule files by merging changes from one file to the other. During the merge, schedule items are merged on a property by property basis. For example, if the text of an appointment is modified in one file and the time is modified in the other file, the appointment will have new text and a new time in the resulting merged file.

Recovered Files

During normal operation, Schedule+ monitors file read and write operations to ensure the user's data integrity remains intact. Data integrity is measured using two methods.

Checksums Each block of data is validated with a checksum. A *checksum* is a calculated value used to test data integrity. If the checksum does not match the recorded value, the operation fails.

Data typing Each data block has a specific type. If the data type of the written file does not match that of the file being read, the operation fails.

If the operation fails on either method, Schedule+ calls an internal recovery function that seeks to prevent further data loss by rebuilding the data file without the damaged data. In recover mode, Schedule+ removes the damaged data and copies the valid data blocks to a new file.

If the damaged file is the local file and the server file is undamaged, the user can synchronize with the server file to merge undamaged data back into the local file. Likewise, if the damaged file is the server file, the user can merge the undamaged local file into the server file. Users can force synchronization by clicking Options on the Tools menu, clicking the Synchronization tab, and then clicking Synchronize Now.

When Schedule+ recovers a damaged file, it stores the recovered file in the user's Temp folder, or if this folder does not exist, in the Windows folder. The filename is Recoverx.scd, where x is a number from 0 to 9. If files Recover0.scd through Recover9.scd exist on the user's computer, Schedule+ continues to overwrite Recover9.scd.

How Schedule+ Works with Microsoft Exchange Server

Schedule+ works with the Microsoft Exchange client for much of its group scheduling capabilities:

- Meeting requests and responses appear as custom forms in the Inbox.
- Meeting attendees are selected from the Microsoft Exchange address book.
- Microsoft Exchange and Schedule+ use the same user list, so there is no need to maintain two different lists.

When combined with a Microsoft Exchange Server, Schedule+ becomes an even more powerful group scheduling platform. In addition to all of the benefits of using just the Microsoft Exchange client:

- Schedule+ adopts the client/server architecture of the Microsoft Exchange Server. You do not need to add anything to Schedule+; the enhancement to Schedule+ is built into the Microsoft Exchange Server.
- Time zone information is stored on each Microsoft Exchange Server so that meeting requests from a user in another time zone are booked at the correct times for their time zone, and free/busy information in the planner view is adjusted for time zone differences.
- Microsoft Exchange Server public folders are used to store free/busy information. Because public folders can be replicated among Microsoft Exchange Servers, this information can easily be shared with others.

- Schedule+ can be incorporated into any public folder application, such as customer tracking, using OLE automation.
- No separate administration is required for Schedule+ on a Microsoft Exchange Server. All of the administrative features of the Microsoft Exchange Server automatically incorporate Schedule+.

Accessing Personal Schedules

Microsoft Exchange stores information in public and private folders. These folders are kept in an information store on a Microsoft Exchange server. The information store is made up of two elements: the public information store, where public folders are located; and the private information store, where private folders are located.

Microsoft Exchange stores users' personal schedules in private folders on their home servers. Each time users change their personal schedules, the Microsoft Exchange private folder is updated to reflect that change. Individual users can delegate access permissions to their personal schedule using the Set Access Permissions dialog box in Schedule+.

When one user tries to access another user's personal schedule in the same site, the following events occur:

- Schedule+ checks to see that the user has the appropriate access permissions to view the personal schedule information.
- If the appropriate permissions are in place, Schedule+ connects to the private information store and logs onto the private folder where personal schedule information is stored.

Accessing Free and Busy Times

Schedule+ free and busy times are stored in a public folder on a Microsoft Exchange server. The Schedule+ Free Busy Information public folder is a hidden folder and can be viewed by choosing the Hidden Recipients command from the View menu in the Microsoft Exchange Administrator program.

The Schedule+ Free Busy Information public folder is created automatically when you install Microsoft Exchange and configure your first site and server. Each time you create a new Microsoft Exchange mailbox, an entry is created in the Schedule+ Free Busy Information public folder for the associated account. By default, each time the users logs in and change the information in their schedule file, their entry in the Schedule+ Free Busy Information public folder is updated to reflect the change. Individual users can prohibit the publication of their free/busy information using the Set Access Permissions dialog box in Schedule+.

The information in the Schedule+ Free Busy Information public folder is available to all other users within a Microsoft Exchange site. When a user invites attendees to a meeting the user scheduling, Schedule+ connects to the public information store on the Microsoft Exchange server that contains the Schedule+ Free Busy Information public folder, logs on to the public folder, and accesses all free/busy information for the attendees.

Sharing Free/Busy Information Between Sites

Schedule+ Free Busy Information can be shared between sites, extending group scheduling functionality to your entire organization. Every Microsoft Exchange public folder is represented by two entries: a directory entry, which records configuration information about the public folder, and an entry in the public information store, which is where the actual content of the public folder is stored.

Each time you create a new site and initiate site replication between the two sites, the directory entry for the Schedule+ Free Busy Information public folder is automatically replicated to the new site. When users in the new site invite users from the original site to a meeting, Schedule+ detects the directory entry for the Schedule+ Free Busy Information public folder, and attempts to connect to the public information store on the Microsoft Exchange server that contains the Schedule+ Free Busy Information public folder.

CHAPTER 6

Microsoft Word Architecture

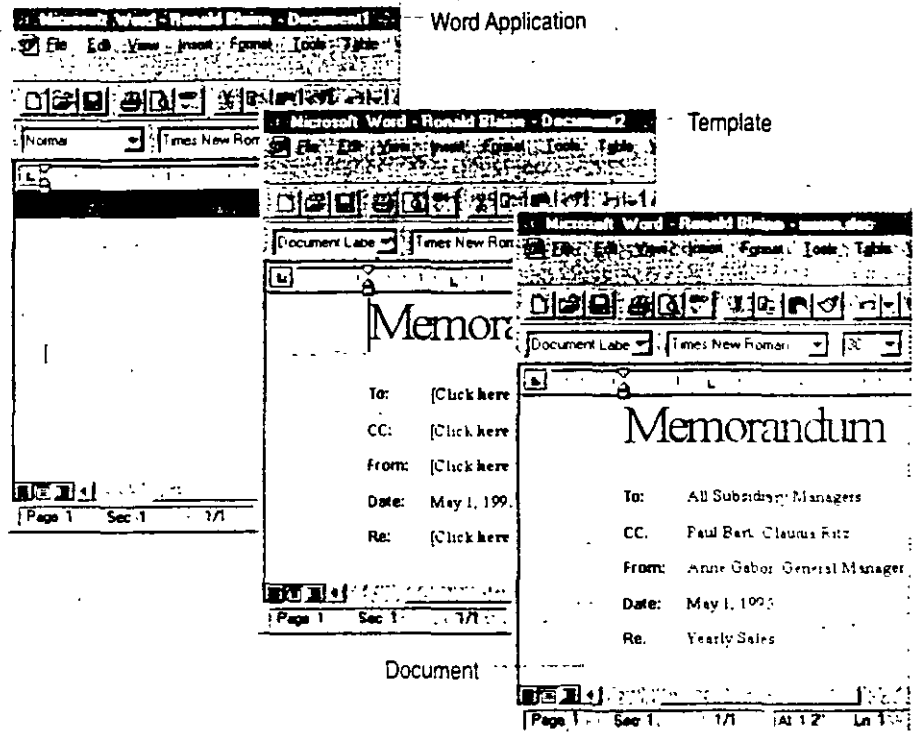
Microsoft Word architecture consists of three layers: the Word application, the document, and the template. Each of these layers affects Word documents in a different way. The Word application provides the standard Word menus, commands, and toolbars. Templates serve a dual purpose: they provide a model for creating new documents and also act as a storage container for styles, macros, AutoText entries, and customized Word commands and toolbar settings. The document file contains the text, graphics, formatting, and settings such as margins and page layout for that particular document.

In This Chapter

Template Layer 112
How Word Resolves Conflicting Settings 118
Where Components Are Stored 119
Managing Styles, AutoText Entries, Macros, and Toolbars 126
Wizards 126
Add-ins 127

Template Layer

The key to understanding Word architecture is understanding how the document, template, and application layers interact. Of the three layers, the template is the layer that has the greatest impact on both the document and the application layers.



Templates can provide the following:

- Customized menus, toolbars, and keyboard assignments that place frequently used commands on the menus and toolbars, remove unused commands from menus and toolbars, and provide key combinations to quickly carry out commands and macros.
- Macros to automate complicated and repetitive tasks with a single command.
- Boilerplate text and graphics, such as your company's name and logo, in every document based on that template.
- Standard formatting, such as fonts and sizes, styles, margin settings, and page orientation.
- The ability to automatically insert frequently used text and graphics by including specialized AutoText entries.

Impact on Application Layer

Users can customize the Word application with templates. With templates, you can adapt Word menus, toolbars, keyboard assignments, and macros to meet the needs of different types of users or for different types of documents.

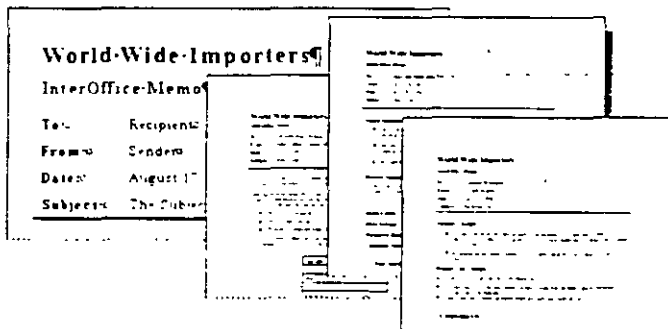
For example, you can create a template for new users that has a toolbar with buttons and menus designed to step them through common tasks. More advanced users, who are familiar with the tasks, don't need the extra guidance and can use a template with toolbar buttons that meet their particular needs.

Impact on Document Layer

A template provides a guide or pattern for creating documents. By basing a group of documents on a single template, users can quickly create standard documents, such as letters and memos, with consistent formatting and standard text.

For example, a template for memos can save users time by setting the page margins, inserting the company logo, and providing the text for standard headings, such as "Memo," "To:," and "From:." With part of the work already done for them, users simply fill in the additional text.

- Template for a memo



Templates give you a head start in creating the types of documents you work with most often

Templates Attached to Documents

In Word, all documents are based on a template. When a user creates a new document, it initially has the characteristics of the selected template. A user can create a new document in several ways.

If the user	Available templates are
Clicks New on the File menu in Word	All Word templates and wizards as defined in the user's User Templates location. Word templates have the file extension .dot. Word wizards are specialized templates with the file extension .wiz.
Clicks Start A New Document on the Office Shortcut Bar	All Office templates as defined in the User and Workgroup templates locations for the Office Shortcut Bar. Word templates are on the General, Letters and Faxes, Memos, Reports, Other Documents, and Publications tabs.
Clicks the New button on the Standard toolbar in Word	Normal template. The New dialog box does not appear.

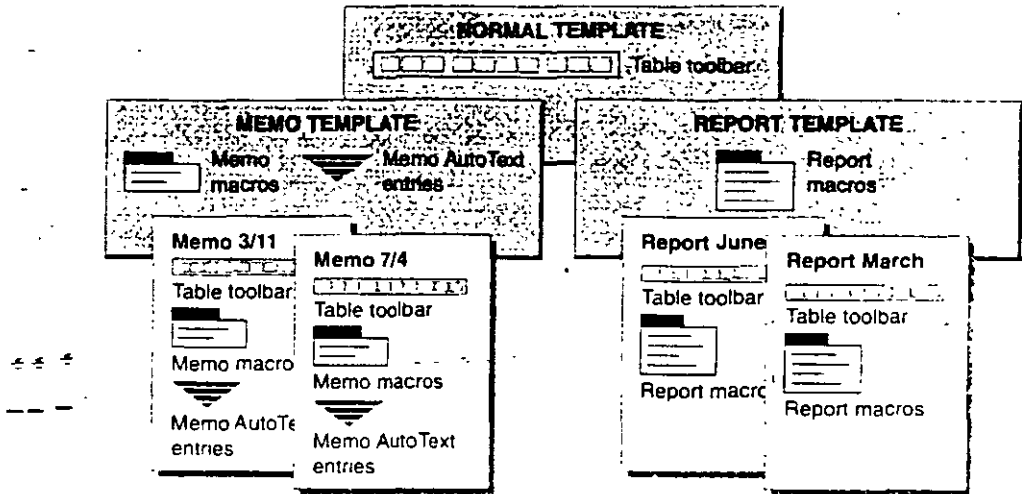
Unless users select another template when creating a new document, Word defaults to the Normal.dot template. In addition to Normal, Word comes with templates for the most common types of documents, such as letters, memos, reports, and mailing labels. Users can use the templates as they are or customize them to their own specifications. Users can also create their own templates.

In addition to basing a new document on a template, users can attach a different template to an open document at any time by clicking Templates on the File menu.

Global Templates

Items stored in the Normal template are global—that is, available to all documents. These items include AutoText entries, macros, menu assignments, toolbar button assignments, and keyboard assignments. If a document is attached to a template other than the Normal template, the items stored in that template are available only to documents that are also based on that template.

Document settings and commands specific to only one type of document, such as a fax sheet, are available only to documents based on that template. Other commands and components, such as AutoText entries, that you want available to all documents should be placed in a global template. By using custom and global templates judiciously, your users have a flexible method for creating documents that can be highly customized while adhering to company-wide standards of design.

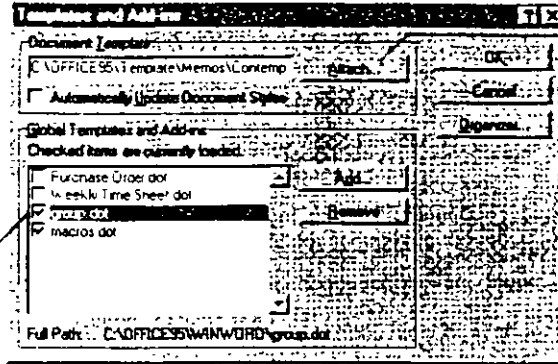


Items stored in the Normal template are available to all of your documents. Items stored in any other template are available only to documents based on that template.

Making Template Items Global

Sometimes users may want to use a macro, toolbar, or AutoText entry from another template. To make these items available to all open documents, you can make a template global. Users can then use the macros, AutoText entries, and customized command settings found in both the global and Normal templates in all their documents. However, styles in the global template are not added to your users' documents; Word continues to use the styles from the template on which the document is based. Customized items in the global template are available for the remainder of the current Word session.

Use the Templates command on the File menu to open a template as a global template.



Attaching a template replaces the template currently attached to the document.

Customized items from selected templates are available to all documents. The template attached to each document does not change.

Making Template Items Global at Startup

You may have special toolbar settings or other items stored in a template that you want to make available to users at any time they are using Word. For example, you may have a special corporate or workgroup template that contains special macros and AutoText entries that all users may want to use. Rather than duplicating this information in every copy of the Normal template, you can make this template available globally when a user starts Word.

To load a template globally at startup, make a copy of the template. Store the duplicate in the Startup folder, which is inside the Winword folder (or the folder indicated in the Startup-path entry of the Registry key Hkey_Current_User\Software\Microsoft\Word\7.0\Options). Do not simply move the template from its original folder. Doing so makes the template unavailable when a user creates a new document by clicking New on the File menu.

Storing Templates

The Office Setup program sets default locations for different kinds of documents and Word components: user documents, user templates, spelling dictionaries, clip art, and so forth. Users can change these default locations by choosing Options on the Tools menu. For information about changing these defaults systematically, see "Modifying Client Setup with the Network Admin Wizard Application" in Chapter 9.

Note Changing settings for User Templates and Workgroup Templates in Word changes them for all Office applications. For more information about specifying user and workgroup templates, see "Starting a New Document" in Chapter 1.

User and Workgroup Templates

In a workgroup, there are usually two places where Word templates are stored:

- Templates used by only a few users or that a user may want to customize (such as Normal, because it's always loaded globally) are usually stored in the user's template folder on the user's local hard disk. The default location of these templates is determined by the User Templates setting.
- Templates needed by an entire workgroup or that users do not need to customize are typically stored on a network volume. The default location of these templates is determined by the Workgroup Templates setting.

When creating a new document, templates from both the User Templates folder and the Workgroup Templates folder are listed together in the New dialog box.

By default, Word stores user templates in the Template folder. For workgroup templates, there is no initial default setting. To change the default location, click Options on the Tools menu, and then choose the File Locations tab. Select User Templates or Workgroup Templates from the File Types list, and click Modify. Enter or select the new folder for user templates and for workgroup templates.

Normal Template

At startup, Word looks for the Normal template in the following locations: the user's Template folder, the Winword folder, and the current folder—in that order. If Word cannot find the Normal template, Word uses the standard document and command settings that have been preset by Microsoft. AutoText entries, macros, and command settings the user stored for global use in the Normal template will not be available until the user returns the Normal template to one of the default search locations and then restarts Word.

Note When Word is first installed, there is no Normal.dot file. It is constructed and saved the first time a user exits Word.

How Word Resolves Conflicting Settings

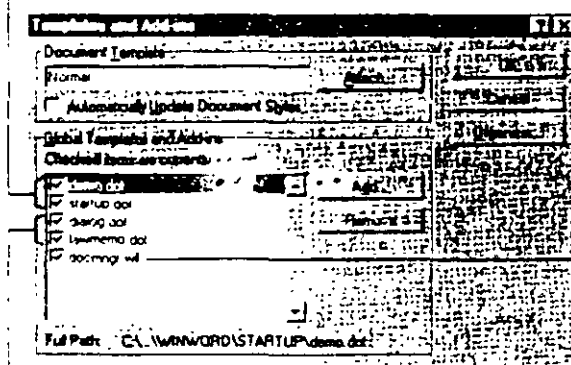
Throughout the discussion of templates, you may have noticed that more than one template can affect the working environment of a document. Each document has access to the macros, AutoText entries, and custom commands and toolbar settings from the attached template, the Normal template, and any global templates. These templates, as well as add-ins or Word itself, may each define a macro or setting in some way that differs from the other templates or add-ins associated with a document. The definition or setting that takes precedence is the one that resides closest to the document. Therefore, Word resolves such conflicts in the following order of priority:

- Template attached to the active document
- Normal template
- Additional global templates
- Add-ins
- Application layer

For example, the style Heading 1 is defined in the Normal template as Arial bold, 18 points, but the document is attached to a template where the style Heading 1 is defined as Times Roman bold, 24 points. Word uses the setting in the attached template, so Text formatted to the Heading 1 style is in Times Roman bold, 24 points.

If several global templates have conflicting settings, Word resolves the conflicts in the order in which they are listed in the Templates And Add-ins dialog box. Templates in the Startup folder appear at the top of the list and have a higher priority. Subsequent ranking on the list is determined by alphabetic order.

Templates in Startup folder show up first in alphabetical order.



Add-ins are listed last.

Global templates follow

Where Components Are Stored

The various components of Word, such as styles, macros, default page settings, AutoText entries, AutoCorrect entries, and custom command settings are stored at two levels of Word's three-level architecture. Some components reside strictly at the document level, others reside at the template level, and still others are stored with both the document and the template. Of those components that reside at the template level, some are stored specifically with the Normal or the attached template, while others can also be stored in a global template. The table below shows how components from each kind of template affect the document and what is contained in the document file.

Component	Document file	Attached template	Normal template	Global templates
Document text and graphics	✓			
Boilerplate text and graphics	✓	✓ ¹	✓ ^{1,3}	
Styles	✓	✓ ²	✓ ^{2,3}	
Default page settings	✓	✓ ¹	✓ ^{1,3}	
AutoText entries		✓	✓	✓
Macros		✓	✓	✓
Custom command and toolbar settings		✓	✓	✓
AutoCorrect formatted text entries			✓ ⁴	

¹ Copied to the document when first created.

² Copied to the document when first created or each time document is opened if the option to automatically update styles is selected.

³ Applies only to documents attached to the Normal template.

⁴ All plain text AutoCorrect entries are stored in the Windows Registry.

Text and Graphics

When a document is first created, Word copies all information from the attached template including boilerplate text and graphics to the document file. From that point on, the text and graphics are saved with the document file. Changes in the boilerplate text and graphics in the template will not affect documents previously created with that template.

Styles

When users create a new document, Word copies styles from the template on which the document is based. No link between template styles and documents styles is made. So, when users change the styles in the document, those changes are saved with the document file. Likewise if users change styles in the template after the document is created, those changes are not automatically reflected in the document.

To automatically update the document styles to match the template, click Templates on the File menu and select the Automatically Update Document Styles check box. Each time a user opens the document, Word copies the styles in the attached template to the document.

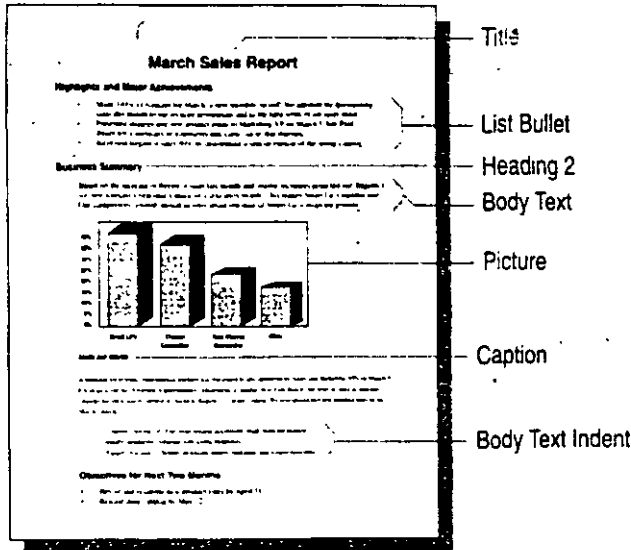
Styles and AutoFormat

≡ ≡ ≡
— — —

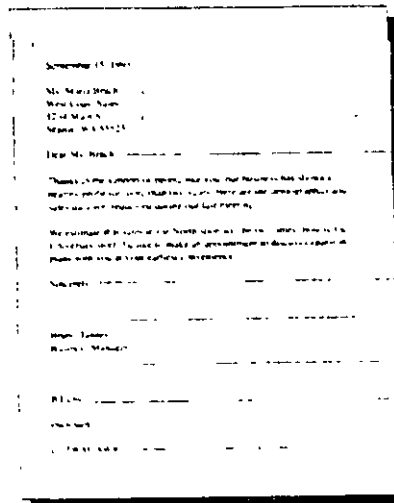
The AutoFormat command analyzes text in a document and applies styles as appropriate from the styles that have been copied to the document. These document styles originally come from the attached template.

If AutoFormat cannot find an appropriate style among those in the document, it uses the built-in styles provided by Word. These built-in styles allow AutoFormat to format body text, headings, bulleted lists, and other text elements. Users can customize built-in styles in their documents by clicking Style on the Format menu and then clicking Modify. Customizing a built-in style in a user's document has no effect on the original style, because the built-in style is copied to the document by AutoFormat.

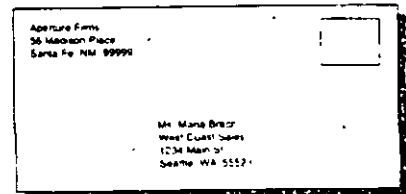
The following illustrations show some of the built-in styles you can use to format common text elements.



- Date - Inside Address



Envelope Return



Salutation

Body Text

Closing

Signature Name

Signature Job Title

Reference Initials

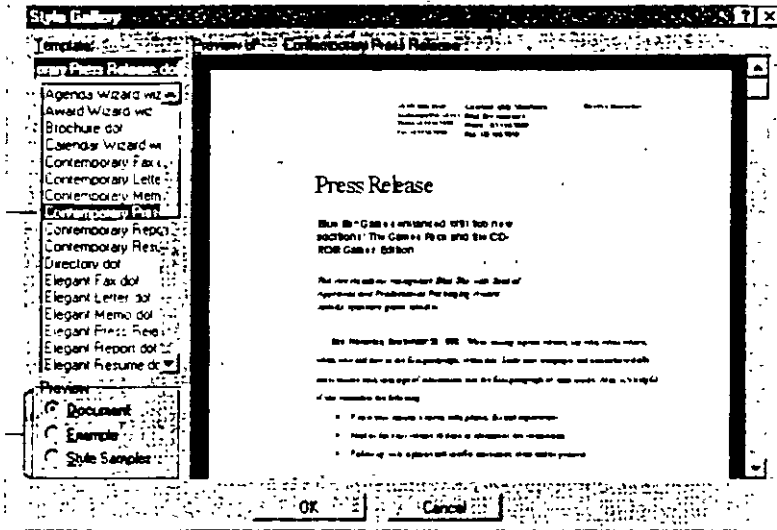
CC

Envelope Address

Style Gallery

When users apply styles to text in a document—either by using the AutoFormat command or by applying styles themselves—they can use the Style Gallery command on the Format menu to see how the appearance of the document changes according to what template is applied.

- Click a template to preview its formatting. Double-click to copy the selected template styles to your document and close the dialog box.



- Preview your document, an example document, or samples of styles in a template.

By using the Style Gallery, a user may decide to apply a different template to the document. When a user chooses to use the styles from another template, Word copies all styles from the selected template into the document. This has the following effects:

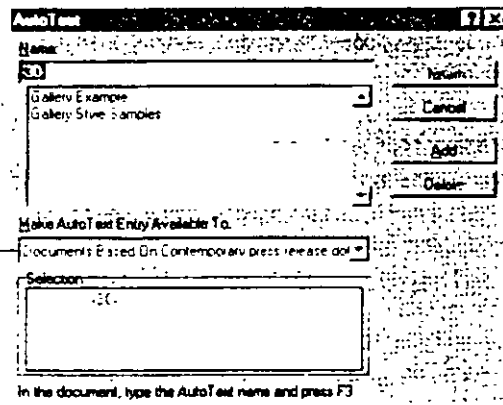
- A style from the template that has the same name as a style in the document replaces the style in the document.
For example, suppose the Heading 1 style in the template is defined as Arial 14-point bold, and the Heading 1 style in the document is defined as Times 12-point italic. Word redefines the Heading 1 style in the document as Arial 14-point bold. All paragraphs formatted with the Heading 1 style are changed to reflect the new style definition.
- Template styles that are not in the document are added to the document.
- Styles unique to the document are not affected.

The Style Gallery command does not attach the selected template to the active document. Rather, it copies the styles from the selected template to the active document. The template on which the active document is based remains attached to the document. To attach a different template to a document, use the Templates command on the File menu.

Users can also use the Style Gallery to preview their own templates. To create an example document, they select the text in the template and assign it the predefined AutoText entry Gallery Example. This predefined AutoText entry is displayed in the preview box of the Style Gallery when Example is selected.

AutoText

AutoText entries are stored in templates. In order for users to use the AutoText entry in any Word document, store it in the Normal or other global template. If you want to make the AutoText entries available only in documents of the same type as the active document, store them in the template attached to the active document.



- AutoText entries are stored in either the Normal template or the template attached to the active document.

To add or edit AutoText entries in a global template, you must open the template in a non-global mode. This is because global templates cannot be edited. To open a global template non-globally, you can open it directly (in the File Open dialog box, select Document Templates in the File Of Type box), or open a document to which the template is attached—not a document to which the template has been loaded.

To determine whether a template is attached or loaded into a document, click Templates on the File menu. The template whose name appears in the Document Template box is the attached template, and is editable. The template whose name appears in the Global Templates And Add-ins box is a loaded template and is read-only.

Macros

Word stores macros in templates. When a user creates a new macro, it is stored by default in the Normal template and is available globally. A user can also specify which template a macro should be attached to.

To store a new macro in a template other than Normal, either the template itself or a document attached to the template must be active when the user creates the macro. If users want to create a new macro in a global template, they must open the template itself rather than an attached document. When attached to a document, global templates are read-only.

- To create a new macro using the Macro dialog box, the user selects the existing template name in the Macros Available In box and then clicks Create to open a new macro-editing window.
- To record a new macro, the user selects the existing template name in the Make Macro Available To box in the Macro Record dialog box.

Together, macros and templates can be used to create a highly customized version of Word designed to accomplish a particular task or set of tasks. For example, a group of macros and templates could be designed to automate the creation of forms and other documents a company uses. Because the macros are stored in one or more custom templates, the custom version of Word is relatively simple to distribute; it's just a matter of copying the templates. If the macros are stored in the Normal template instead, it is much more complicated to install the custom version on another machine, which already has its own Normal template.

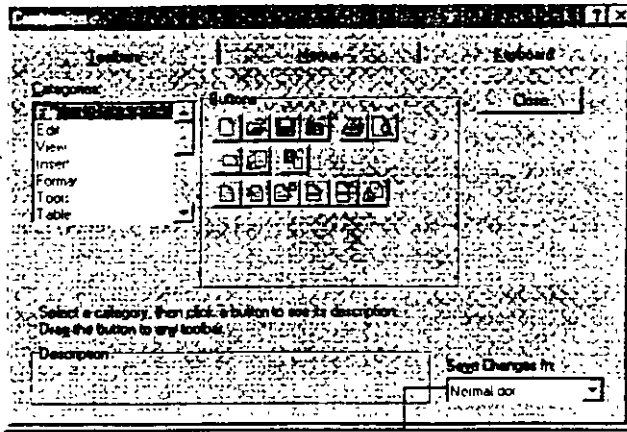
AutoCorrect

Any AutoCorrect entries that use formatted text, with Formatted Text selected in the AutoCorrect dialog box, are specific to Word, and are stored in the user's own Normal template.

Custom Toolbars, Menus, and Shortcut Keys

By customizing toolbars, menus, and shortcut keys, you can change Word to better suit the needs of your users. For example, you can add frequently used commands and dialog box options to toolbars and menus. And you can remove items users rarely use. You can also customize shortcut key assignments by creating the shortcut keys that work best for your users.

To customize toolbars, menus, and shortcut keys, click Customize on the Tools menu. You can specify whether to store these customizations in the Normal template, an open template, or the template attached to the active document. To make customized settings available in any document, save the settings in the Normal template or to a template that will be made global.

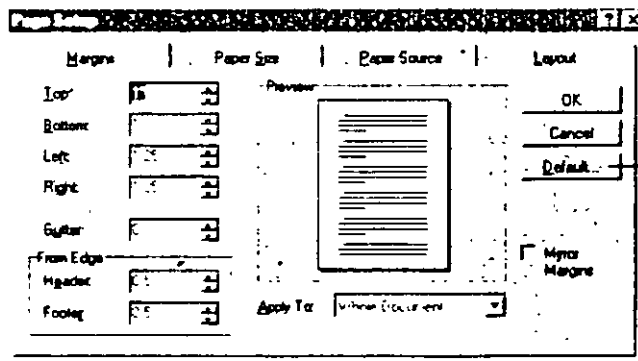


Select template here.

Page Setup Default Settings

Default page settings for the document layout, such as margins, layout, and paper size, are similar to styles. The default settings are saved with a template. Each new document based on that template inherits those default settings from the template. Subsequent changes to the settings in the document reside with the document and are not automatically saved to the template. However, these settings can be stored in the template.

To save settings from the document to the attached template, click Default in the Page Setup dialog box. The changes will be reflected in subsequent documents created from this template.

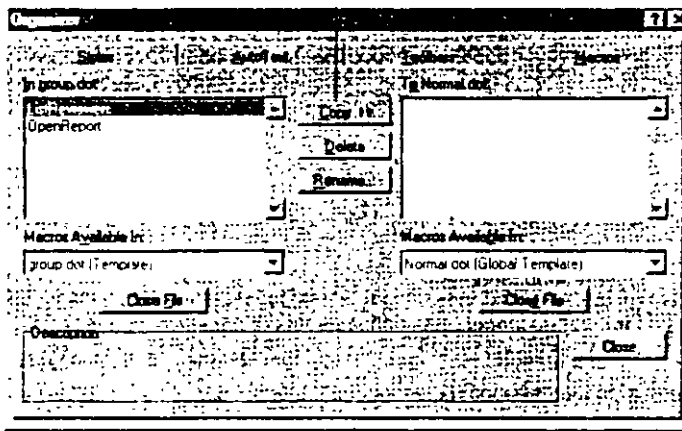


Save settings to the template attached to the active document.

Managing Styles, AutoText Entries, Macros, and Toolbars

Users may want to take advantage of styles, toolbars, macros, or AutoText entries that have already been created in another template. They can use the Organizer dialog box to copy these items between templates. Styles can also be moved between templates and documents.

You can copy macros between any two templates.



Users can also use the Organizer to delete or rename styles, AutoText entries, macros, or toolbars. To open the Organizer, click Templates on the File menu and then click Organizer.

Wizards

From a user perspective, a *wizard* is a fast, easy way to create a document—because the wizard does the work. Architecturally, wizards are specialized templates. They are designated with a .wiz extension instead of a .dot extension, and their Type is described as MS Word Wizard rather than MS Word Template. The wizards that come with Word all share a common user interface. The key component of the wizard is the macro or set of macros stored with the template. These macros automate the creation of the document.

Word provides wizards to help users create the following types of documents: agenda, award, calendar, fax cover sheet, legal pleading, letter, memo, newsletter, and resume. Word also includes a wizard to help users create tables. You can create your own wizards to step users through complex tasks specific to your company. For more information about creating your own wizards, see "Creating a Wizard," in Chapter 9 of the *Microsoft Word Developer's Kit*.

Add-ins

Add-ins are programs written in the C programming language. They act like custom commands or custom features you can use in Word. For more information about add-ins, see Appendix C in the *Microsoft Word Developer's Kit*.

You can place add-in commands on toolbars and menus or assign them to shortcut keys, just as you can with Word macros. Architecturally, add-ins fit in at the template layer. They modify toolbars, menu commands, and shortcut keys just like templates.

- You can write your own add-ins or obtain them from software vendors. To use an add-in, you must load it into Word. Add-ins are installed and loaded the same way global templates are. Like a template, the add-in program remains available until you quit Word. If you want to load an add-in program automatically each time you start Word, store it in the Startup folder.

F 10

CHAPTER 8

Installing Microsoft Office

This chapter describes the most basic installation process for Microsoft Office for Windows 95. Because there is a lot of flexibility in the installation process, the chapter begins with the installation planning process and installation options, and is followed by specific procedures for you to follow as you install Office.

*** If your Office setup needs to be customized, the next chapter, "Customizing Client Installations," covers how to alter Office Setup to customize the installation process for your specific needs.

In This Chapter	Installation Planning 157
	Client Installation from Physical Media 166
	Client Installation Over the Network 170
	Installation Issues 179
	Client Installation Example 187

Installation Planning

Before beginning the process of installing Office on your users' computers, you need to make some decisions about the installation and plan the process. There are a number of options to choose from in the following main areas.

Decision	Description
I. Location of Office files	Where Office files are installed
Local hard disk	All files are installed on user's local hard disk.
Server	Files are shared on the network server.
Both	Some components are installed locally, and some are shared on the server.
Shared Windows	Used with any of the above options, system files are installed in a shared Windows environment.

Decision	Description
2. Installation options	Which Office files are installed
Typical	Installs common Office files.
Compact	Installs minimum Office files.
Custom	User selects which Office files to install.
Run from CD	Leaves some files on CD to save disk space
3. Installation media	Where Office files are coming from
Physical media	Floppy disks or CD.
Network	Central network share.
4. Installation method	How the installation is done
Interactive	User runs Setup interactively.
Batch	Setup runs unattended.
Push	Setup is initiated without user intervention.
5. Application-specific installation issues	Issues that affect what must already be installed, such as a mail system, and what you install for the individual Office applications.

Some of these decisions affect others. For instance, if client installations are from physical media, then all files must be installed locally and the only installation method available is interactive.

Location of Office Files

When deciding where to place Office files, consider how the client computer will be used and evaluate the benefits of each placement option, including how each option will affect your ability to support these configurations over the long run.

Both Windows and Office have the option of being *local* (files are installed on the client computer's local hard disk) or being *shared* (files are located on a network server and users share those files over the network).

Consider the following as you make this decision:

- **Performance** What is the performance on the client computer when executing Windows or Office software? Executing Windows software and Office applications locally is generally faster than executing them over the network.
- **Disk space** How much local disk space is available on the client computer for Office software? Sharing software from a server reduces the disk space needed on a client computer's local hard disk, while it increases the need for disk space on the server.

- **Maintenance and support** How do you install new software in your organization and how are updates to Windows or Office software distributed to users? Software on a server can be maintained and updated more easily than software on client computers.
- **Availability** Is Windows and Office software available when the user needs it? Software installed locally is always available to the user, while software shared from a server is available only when the network is up and the server is running.

Client computers can be configured to run under any combination of these environments:

	Local Windows, local Office	Shared Windows, shared Office	Shared Windows, partially shared Office ¹
Performance	Best overall	Lowest overall	Compromise for Office, lowest for Windows
Disk space	Highest local usage, lowest server usage	Lowest local usage, highest server usage	Split between client and server computers
Maintenance/ support	No central control complicates task	Simplifies control for both Office and Windows	Simplifies control for Windows
Availability	Always available	Depends on both Office and Windows servers	Depends on both Office and Windows servers

¹ Executables for Office are local, other support files are on the server

Office software can be split between the server and client computers, or between the local hard disk and the CD, providing even more options for distributing disk space and managing the software.

Windows Installed Locally vs. Shared

This decision is made when you deploy Windows. If Windows is installed locally, all Windows software is on the client computer's local hard disk. When Windows is shared, you put the common Windows files on a network server and only user-specific data files are stored on the local hard disk.

Office can be installed under either environment. Some files included with Office are system-level files, and they are installed in the Windows folders. If Office is installed in a shared Windows environment, those files are shared from the Windows server and the space required by Office on the local hard disk is reduced, even if Office itself is installed locally.

Whether Windows is local or shared has a bearing on the amount of available disk space for Office as well as overall performance of the client computer. It also has a bearing on maintenance and support, as Windows software installed once on a central server is easier to control than software installed on each user's computer.

For more information about making this decision for Windows 95, see Chapter 4, "Server-Based Setup for Windows 95," in the *Microsoft Windows 95 Resource Kit*.

Using Windows 95 system policies is an alternate way to control the Windows environment even if the Windows software is installed locally on users' computers. For information about using Windows 95 system policies to configure client computers from a central location, see "Customizing the User Environment" in Chapter 11.

Office Installed Locally vs. Shared

If Office is installed from floppy disk, the only choice is to install all files locally and place them on the client computer's local hard disk. If Office is installed from CD, the user has the choice of installing all files on the local hard disk or installing some files locally and leaving the rest on the CD to be accessed as needed. This reduces the local disk space needed for Office.

If you install Office from a network server, you can leave some or all of the Office files on the server and put only some of the files on the local hard disk. This can be done regardless of whether Windows is local or shared.

Installation Types

There are several basic installation types available in the Office Setup program when Setup is run interactively by the user. You can also create your own customized type if the Setup program is to be run in batch mode using a customized script.

Typical and Compact For these two types, a predefined set of Office features is installed. The Typical type installs the most commonly used Office components, while the Compact type installs only the minimum files needed to run Office, and uses the least amount of disk space.

Custom The third type gives the user a list of all available Office components and the user can select which options are installed.

Run from CD When installing from CD, a fourth type is available, which leaves some files on the CD to be accessed directly from there rather than copying them to the local hard disk.

Using the tools available on the CD supplied with this book, you can also create a customized Setup script in which you define the features to be installed on the client computer. Setup is run in batch mode using this script, with no interaction by the user, and the components are installed as you defined them. For more information, see Chapter 9, "Customizing Client Installations."

Installation Media

Office can be installed on a user's computer from the floppy disks or CD, or Office can be placed on a network server and users can install Office from there.

Floppy disks Floppy disks do not need special hardware (such as a CD-ROM drive or networking hardware), but they require the installer to be physically present at the computer during installation.

CD The CD contains additional files not available on the floppy disks and a special CD installation option allows the user to leave some files on the CD to reduce local disk space usage. But a CD requires a CD-ROM drive and it requires the installer to be physically present at the computer during installation.

Network Installing over the network simplifies the distribution of Office to users, and it provides additional options for sharing files among users. But installing over the network requires network access during installation, and, if files are shared, continuous network access while Office is being used.

An alternative to installing Office directly on client computers from physical media is to perform an administrative installation that copies the Office files on a network server. Users then install Office directly from the server. In this case, you can customize the installation process by modifying the Setup script that users will be executing on the server.

Installation Method

After decisions have been made regarding the client configuration, there are three methods that can be used to install Office on the client computers.

Interactive The interactive method means users run Setup and choose the installation options they want. This is the only method that can be used if Office will be installed from physical media.

Batch The batch method involves creating a customized Setup script that pre-defines all the installation options for the user. When Setup is run using the script, installation occurs with no interaction with the user.

If you have a custom installer program that you use in your organization for installing software and tools on client computers, you may be able to add the installation of Office to the installer's list of software. Using the Setup batch capability, you can define a command line for Setup that installs Office with the client options you've chosen. Using Microsoft Mail, you can create a message with a Windows package object that contains a link to the setup batch script. The script will run when the user double-clicks the object in the mail message.

To completely automate the installation process, use batch scripts with push installations.

Push A *push* installation of Office is an installation that is run on a user's computer without user intervention. You can do this by editing a user's logon script to launch the installation process when the user logs on to the network, or by using a tool such as Microsoft Systems Management Server to perform the installation on remote-user computers.

Application-Specific Installation Issues

Some Office applications have special installation requirements that you need to consider before completing your overall Office installation plan. If you intend to install these applications, this section will help you understand these requirements before you continue with your Office installation.

You need to also read the `Ofreadme.txt` and `Network.txt` files included with Office to be sure you understand the latest updates and issues associated with installing and running Office.

Electronic Mail-Enabled Applications

Because Office Setup automates the installation of electronic mail support, you should install and set up any email client software on user systems before installing Office. For example, if a user wants to use Word as their email editor, the Windows Exchange client must be installed before Office or Word Setup is run. If Exchange is installed after Office, Office or Word must be reinstalled.

If users have more than one email client, Setup asks during client installation which client the Office applications should use.

For more information about how to install and configure mail-enabled applications, including using Word as your email editor, see "Interoperability with Electronic Mail" in Chapter 15.

Microsoft Access

If you are using Microsoft Access to link Paradox data on a network at the same time as users of Paradox, you must be sure to define a unique Paradox user name in the registry.

The `ParadoxUserName` entry is in the Windows registry folder `HKkey_Local_Machine\Software\Microsoft\Jet3.0\Engines\Paradox`. When Setup asks the user for a user name and organization name, Setup puts the user name in this registry key when Office is installed. If the user does not specify a user name when asked by Setup, or if Setup is run with the `/q` command line option to run without user interaction but without the `/n` option to specify a user name, then Setup puts the organization name into this registry key.

There must not be more than one user registered with the same Paradox user name on the same network, so it is important to specify a unique user name during Office Setup.

Microsoft Schedule+

From the administrator's point of view, setting up Schedule+ is somewhat different from setting up other Microsoft Office applications. Most setup and configuration issues for using Schedule+ in workgroup mode are taken care of as part of the process of setting up a postoffice that can be accessed by the Windows 95 Exchange client, or happen automatically when a user logs on to Schedule+ in workgroup mode for the first time.

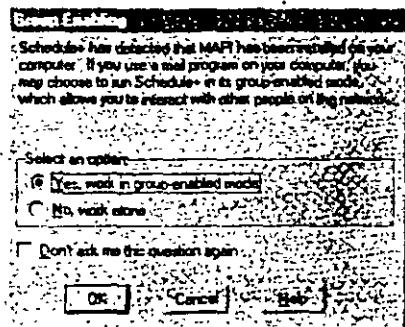
Installation Requirements

Using Schedule+ in workgroup mode requires Windows 95, and the Windows 95 Exchange client connected to a MAPI-compliant postoffice. The postoffice can be one of the following:

- Windows Messaging System (WMS) workgroup postoffice provided with Windows 95 or Windows for Workgroups 3.x.
- Workgroup postoffice provided with Windows NT 3.x.
- Microsoft Mail Server 3.x postoffice.
- Microsoft Exchange Server postoffice (available after the release of Windows 95).
- Any mail server or messaging service provider for which the vendor has provided a valid MAPI 1.0 driver and transport specification.

Note Schedule+ for Windows 95 runs only in stand-alone mode on Windows NT 3.51. When Microsoft Exchange Server is available, it will include a Windows NT Exchange client that allows Schedule+ to be used in workgroup mode.

To make using Schedule+ as straightforward as possible, set up the postoffice and configure the Exchange client before Schedule+ is started for the first time. If the Exchange client is installed as part of the Windows 95 process but is not yet connected to a valid postoffice, users see the following dialog box when they start Schedule+.



Note This dialog box also appears on workstations where the Microsoft Mail 3.x client is installed. Because Schedule+ 7.0 requires the Exchange client for workgroup mode, Schedule+ 7.0 can only be used in stand-alone mode on a Microsoft Mail client workstation. Users who are working with Schedule+ 7.0 in stand-alone mode should click No, Work Alone and Don't Ask Me This Question Again before clicking OK.

For more information about the Exchange client setup process, and on how to specify a postoffice after the Exchange client has already been installed, see Chapter 26, "Electronic Mail and Microsoft Exchange," in the *Microsoft Windows 95 Resource Kit*.

Using Schedule+ in Stand-Alone Mode

No special intervention by the administrator is required for Schedule+ to be used in stand-alone mode. The first time Schedule+ is started, a new schedule (.scd) file is created for the current user in the Windows folder of the local computer.

Using Schedule+ with a Windows 95 Workgroup Postoffice

The recommended steps for setting up Schedule+ for use with the Windows Messaging System (WMS) postoffice provided with Windows 95 are as follows:

1. Set up the workgroup postoffice on a computer where all users in the workgroup will have access.
2. Install Windows 95 on the user machine and specify the Windows 95 workgroup postoffice during the Exchange-client phase of setup.

For more information about how to set up a Windows 95 workgroup postoffice and configure the Exchange client, see "Using the Microsoft Exchange Client with Microsoft Mail" in Chapter 26 of the *Microsoft Windows 95 Resource Kit*.

Note Installing Microsoft Exchange on a computer with Schedule+ 1.0 disables the workgroup functions of Schedule+ 1.0. If Windows 95 is installed using interactive setup, and if the Exchange client is on the list of installed options and Schedule+ 1.0 is present, the user sees a message that Schedule+ 1.0 can no longer be used in workgroup mode, and asking if they want to continue anyway. The default answer is Yes.

The first time a user starts Schedule+, a schedule file is created on the user's local computer just as in stand-alone mode. At the next synchronization interval, the local .scd file is copied to the Cal folder on the workgroup postoffice server. Synchronization takes place by default every 15 minutes, and also when quitting Schedule+. The user can also force synchronization to take place by clicking Synchronize Now on the Synchronize tab in the Options dialog box.

The fact that a workgroup postoffice user's schedule file is not copied to the server until the user logs on for the first time means that that user's free and busy times and schedule details cannot be read until they start Schedule+ at least once.

- A user whose local schedule file has not yet been copied to the server can still be invited to meetings, because their name is present in the postoffice address book. However, a question mark displays next to the user's name in the Invite pane of the Schedule+ Planner view.

Using Schedule+ with a Microsoft Mail Server

From the client perspective, the steps in setting up Schedule+ 7.0 to work with a Microsoft Mail version 3.x server are the same as those described in the previous section.

On the server side, the Schedule+ 1.0 administration program (Adminsch.exe) that is distributed with Microsoft Mail Server can still be used to administer most postoffice-related Schedule+ functions. The only commands that do not work are the Adminsch.exe commands for deleting archive information and deleting inactive Schedule+ schedule files. Schedule+ 7.0 schedule files have a .scd extension which the Schedule+ 1.0 admin program does not recognize.

Running the server clean-up commands of the Schedule+ 1.0 administration program on Schedule+ 7.0 schedule files can damage the files.

An update to the Adminsch.exe program that fixes this limitation may be available at a later date through the Microsoft Product Support Quick-Fix Engineering (QFE) program.

Using Schedule+ with Other MAPI Servers

To use Schedule+ for Windows 95 with other MAPI 1.0 mail servers, set up the postoffice first and then install Windows 95 and the Exchange client, as recommended above.

For information about how to best configure the MAPI server for use with Schedule+, see the documentation that comes with your server.

Upgrading to Microsoft Exchange Server

A next-generation client-server messaging system known as Microsoft Exchange Server will be available from Microsoft after the release of Windows 95. Microsoft Exchange Server provides advanced electronic mail, scheduling, groupware applications, and custom application development. Microsoft Exchange Server includes MAPI drivers that extend the capabilities of the Windows 95 Microsoft Exchange client. For example, when Microsoft Exchange is connected to an Exchange Server, you can access a replicated public folder, create custom forms and shared documents, schedule meetings with others, manage time and tasks, and create custom electronic forms for use in business.

Microsoft Exchange Server requires a computer running Windows NT Server version 3.51 or higher. It also includes a built-in X.400 gateway, plus support for SMTP and for Microsoft Mail for Intel® and Apple® Macintosh® computers. For more information about Microsoft Exchange Server, contact your Microsoft sales representative.

Microsoft Word

Not all converters are installed with Word in a Typical or Compact Office installation. To ensure that you get the converters you need, install Office using the Custom installation type and select the converters you want. You can reinstall Office later to add or remove converters as your needs change.

For a list of converters installed with each installation type, see Appendix C, "List of Installed Components."

Client Installation from Physical Media

If you have decided to use physical media, either floppy disks or CD, to install Office on each client computer:

- The user or whoever performs the installation must be physically present at the user's computer.
- Office files are installed on the user's local hard disk or, if installing from CD, the user can install some components on the local hard disk and access the rest directly from the CD.

- The user chooses what components will be installed and the name of the top-level folder.
- Setup is run interactively.

Client Installation from Floppy Disks

To install from floppy disks, you need to either purchase a copy of the product for each user, or, if you have multiple licenses and a large number of users, you can copy the floppy disks to a network file server. Users can then run Setup from the server. Setup functions the same as if it had been run from the floppy disks except that it will not prompt the user to insert each floppy disk.

The advantage of copying the floppy disks to a server over creating a normal administrative installation point is that this method takes less server disk space because the files are compressed. However, it is much less flexible and provides little opportunity for customization. For information about creating an administrative installation point, see "Client Installation Over the Network" later in this chapter.

--- The procedures in this section cover both installing from floppy disk and from the server.

▶ To install from floppy disks

1. Insert Disk 1 into the disk drive.
2. In Windows 95 Explorer, switch to drive A, and then double-click Setup, or click Run on the Start menu and type `a:\setup`
In Windows NT, double-click Setup.exe in File Manager or click Run on the Program Manager File menu and type `a:\setup`
3. Follow the instructions on the screen. If needed, give instructions to your users about what to do on each screen for your organization. Users can get help while running Setup by pressing F1 or choosing the Help button.
4. Select the installation type: Typical, Compact, or Custom.

For Typical and Compact, the components installed are predefined. For a list of files installed for these two installation types, see Appendix C, "List of Installed Components." For the Custom installation type, the user chooses which of the Office components to install, and, for some components, can also change the default destination folder.

Use the next procedure if you have multiple licenses and want to copy the disks to a server so users can install from there. The following example uses an MS-DOS batch file for copying all the disks to a folder on a file server. You could also enter the equivalent command line for Extract.exe in the Run command in the Windows 95 Start menu or in the Run command in the File menu of the Windows NT Program Manager.

To copy floppy disks to a file server

For this example, assume the floppy disk drive is drive A: and the file server folder in which you want to copy the Office files is X:\disks.

1. Create a folder on your file server:

```
mkdir x:\disks
```

2. Copy all the files from disk 1 to a subfolder called disk1.

```
mkdir x:\disks\disk1
copy a:\*.* x:\disks\disk1
```

3. Use Extract.exe (which you copied to X:\disks in step 1) to copy each subsequent disk to additional subfolders. To do this, create two batch files:

- Create X:\disks\copydisk.bat:

```
echo Insert disk #%1 in drive A
pause
mkdir x:\disks\disk%1
a:
for %%I in (*.*) do x:\disks\disk1\extract /c a:\%%I
x:
x:\disks\disk%1\%%I
x:
```

Given a disk number in the command line (%1), this batch file prompts you to insert the disk into drive A, creates the corresponding subfolder under X:\disks, and then copies the cabinet file from the disk to the subfolder.

- Create X:\disks\copyall.bat:

```
@echo off
call x:\disks\copydisk.bat 2
call x:\disks\copydisk.bat 3
call x:\disks\copydisk.bat 4
```

...and so on, for each disk in the floppy disk set. This batch file calls the previous batch file once for each disk.

- Run X:\disks\copyall.bat and insert each floppy disk as prompted. This creates a complete set of subfolders under X:\disks for all the disks in the set.

4. Make X:\disks available to users of the file server.

Users can now connect to the file server in folder X:\disks, run Setup.exe from the Disk1 subfolder, and proceed as if they were running Setup directly from the floppy disk set. Instead of prompting for each floppy disk, Setup goes to each subfolder of X:\disks to find the next set of files.

For more information about the floppy disk format and the Extract.exe program, see "Copying Disks and Extracting Files Manually" later in this chapter.

Client Installation from CD

To install from CD, you need to either purchase a copy of the product for each user, or, if you have multiple licenses, you can copy the CD to a server. The procedures in this section cover both cases:

If you have a large number of users, and you have a single Office CD with multiple licenses, you can share the CD on a network server or you can copy the files from the CD to a server. Users can then run Setup from the server. Setup functions the same as if it has been run from the CD except that if you copy the files to the server, users will not have the Run From CD installation option.

The advantage over creating a normal administrative installation point is that this method takes less server disk space, because the shared component files are not duplicated. However, it is less flexible and provides less opportunity for customization. For information about creating an administrative installation point, see "Client Installation Over the Network" later in this chapter.

--- To install from CD

1. Insert the Office CD into the drive.
2. In Windows 95 Explorer, switch to the CD drive, and then double-click Setup, or click Run on the Start menu and type *drive:\setup* where drive is the the CD-ROM drive letter.

In Windows NT, double-click Setup.exe in File Manager or click Run on the Program Manager File menu and type *drive:\setup*

3. Follow the instructions on the screen. If needed, give instructions to your users about what to do on each screen for your organization. Users can get help while running Setup by pressing F1 or choosing the Help button.
4. Select the installation type: Typical, Compact, Custom, or Run From CD. For Typical and Compact installation types, the components installed are predefined. For a list of files installed for these two installation types, see Appendix C, "List of Installed Components."

For the Custom installation type, the user chooses which of the Office components to install, and, for some components, can also change the default destination folder.

For the Run From CD installation type, a predefined small set of components is installed on the user's hard disk while most of the files are left on the CD and run from there. For a list of which files are installed on the disk and which are left on the CD, see Appendix C, "List of Installed Components."

Use the next procedure if you have multiple licenses and want to copy the CD to a server so users can install from there. For this example, assume the CD disk is in drive D and the file server folder to which you want to copy the Office files is X:\disks

► **To copy CD contents to a file server**

1. Create a folder on your file server:

```
mkdir x:\disks
```

The server must have enough space to hold the entire contents of the CD. For Microsoft Office Standard for Windows 95, this is about 95 MB; for Office Professional, it is 126 MB.

2. Copy all the files from the CD to the x:\disks folder. You can use the MS-DOS XCOPY command to copy the entire folder hierarchy:

```
xcopy d:\ c:\disks\ /s /e /v
```

3. Make x:\disks available to users of the file server.

Users can now connect to the file server in folder X:\disks, run Setup.exe, and proceed as if they were running Setup directly from the CD. Setup detects that Setup is not running from a CD and presents only the Typical, Compact, and Custom installation types.

≡ ≡ ≡

Client Installation Over the Network

If you have decided to have your users install Office over the network, you and your users will be involved in a two step process:

1. You must create the *administrative installation point*—the set of folders that will hold all the Office software, residing on a file server to which users can connect.
2. Users will perform client installations—running Setup from the administrative installation point to install Office software on their computers.

Creating an administrative installation point is different from just copying files from the floppy disks or CD to a server. Some or all of the Office component files can be shared from the administrative installation point reducing the disk space used on user computers. The shared Office components, such as Spelling Checker and ClipArt Gallery, can be installed to a different folder or even a different server to allow for more flexibility in sharing Office components. Also, you can modify the administrative installation point to customize how users install Office. For details on how to customize an administrative installation point, see Chapter 9, "Customizing Client Installations."

While you use Setup.exe to create the administrative installation point, and users use the same Setup.exe to install Office, it is run in a different way for each task. To avoid confusion in the discussion below, these two modes will be referred to as follows:

- To create the administrative installation point, you run Setup.exe from the physical media with the */a* command line option. This is called *administrative Setup* and is run once to create the administrative installation point.
- To perform a client installation, a user runs Setup.exe from the administrative installation point without the */a* command line option. This is called *client Setup* and is done every time a user wants to install Office on his or her own computer.

Creating the Administrative Installation Point

There are two steps to creating an administrative installation point. First, you need to select the server and prepare it, making sure there is enough disk space and that it's free from viruses. Second, you install the Office software on the server.

Preparation

The administrative installation point contains two primary folders:

- **Main Office Folder** The first folder is for the main Office application files, such as Winword.exe and Excel.exe. The default folder name is \Msoffice. If your users will be sharing these main Office files over the network rather than installing them locally, this is the folder they will use to access those files.
- **Shared Applications** The second folder is for the shared application files, such as Mspell.dll (Spelling Checker) or Artgalry.exe (ClipArt Gallery). The default folder name is \Msapps. The shared applications are those tools that are used by more than one Office application. The shared components include: Spelling Checker, WordArt, Organizational Chart, Microsoft Graph 5.0, MS Info, Clip Art Gallery, graphics filters, text converters, and Microsoft Excel sheet converters.

If your users will be accessing these shared component files over the network rather than installing them locally, this is the folder they will use to access those files. For more information about this option, see "Client Installation" later in this chapter.

When you perform an administrative installation, Setup asks for the location of these two folders. They don't have to be on the same disk or even on the same file server. But you do need to share these folders on the network so that users have read access to both of them. You can either create a single server share that contains both folders, or you can create two separate shares, on one server or even on separate servers, with one share for each folder.

Later, during client installation, users will access these folders as you have defined them here: they will connect to the main Office folder to run client Setup, and they will be asked to identify the shared applications folder during Setup.

To prepare the administrative installation point, use the following check list:

- Pick the location of the administrative installation point.
 - Make sure that there is sufficient disk space. Office Standard needs about 125 MB for the administrative installation, Office Professional needs about 170 MB.
 - You need read, write, delete, and create permissions to use these folders while doing the administrative installation.
- At the administrative installation point, before beginning the installation procedure:
 - Make sure that all folders are empty. If a previous version of Office exists, delete all of it; move any custom templates you want to save.
 - All users sharing applications or running shared Windows from this server must be logged off.
 - Lock all folders to network user access during the administrative installation.
 - Disable virus detection software to prevent erroneous virus detection triggers as Setup writes into various executable files.
- The computer on which you will be running Setup.exe must be running Windows 95, Windows NT Workstation 3.51 or later, or Windows NT Server 3.51 or later.
- Either before or after running administrative Setup, share the folder that contains the main Office applications and the folder that contains the shared applications. Users need read access only to these folders.
- If users will be running in a server-based Windows environment, after running administrative Setup, run client Setup once on the same server-based Windows environment that your Office users will be using. During the first client installation, Setup will place some files, such as shared applications, in the server-based Windows folders for all users to share. You will need write and create permissions to the server-based Windows folders. Subsequent client installations will not need write access to the shared Windows folders because the shared applications will already be there.

Installing the Software

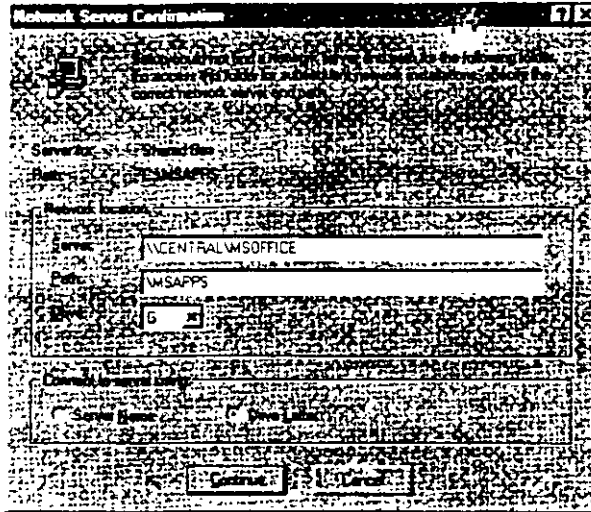
After you have the administrative installation point prepared, do the following to install Office software in those folders.

► **To create the administrative installation point**

1. If you are installing from floppy disks, insert Disk 1. If you are installing from CD, insert the CD.
2. In Windows 95, click **Run** on the Start menu and then type **setup /a**
In Windows NT, click **Run** in the Program Manager or File Manager File menu and type **setup /a**

Note Because you must use the /a command line option to run Setup in administrative mode, you have to type the command to start Setup rather than double-clicking Setup.exe.

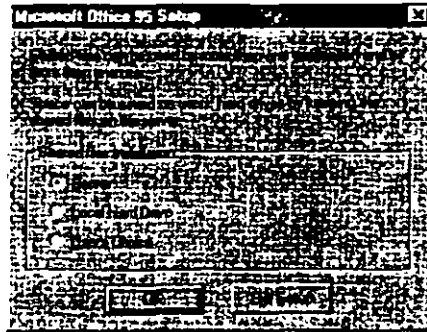
3. When asked, type your organization's name.
When users install from this installation point, Setup will use the organization name you type here as the organization name for each user.
4. Setup will next verify the product number.
If you are installing from CD, enter the product ID printed on the CD package. If you are installing from floppy disks, Setup generates a unique product ID and displays it for you.
5. Setup will ask for the destination folder for Office. If necessary, change the folder name to the name of the folder where you want the main Office files to reside.
6. Setup will next ask for the destination folder for the shared applications. If necessary, change the folder name to the name of the folder where you want the shared application files to reside.
7. Setup will ask for the server and path for the shared applications folder. Type the exact path that users will have to type when they connect to it to install Office.
You can specify whether the shared applications folder is referenced with a drive letter or with a UNC path (UNC stands for Universal Naming Convention and refers to the \\servername\sharename syntax used for defining servers and shared areas on the server for LAN Manager-compatible networks). If you select the Drive Letter option, the drive letter in the Drive box is used for connecting to the location of the shared files.



Setup will try to verify the server by connecting to it as specified. If it cannot connect, you'll see an error message. This error can occur for several different reasons. The following table explains what to do in each case to correct the error.

Cause	Correction
Misspelled server or share name.	Click Edit and correct the name.
Share not yet created.	Click Continue and remember to create the share before users will need it.
You are running directly on the server and you cannot connect to a local share on the server (many network operating systems will not allow this).	Click Continue; you can verify the share later from another computer on the network.
The UNC path does not work for some other reason.	Click Continue; be sure to verify that the share is accessible before telling users to install from this server.

8. Setup will ask how shared applications are to be installed by the end user. The option you choose will determine whether the user can choose how to install the shared applications, or whether you make that choice for them. The options are:
 - **Server** Shared applications remain on the server and users access them over the network. Users running client Setup do not get a choice, even if they are running Setup interactively.
 - **Local Hard Drive** Shared applications are installed on the user's local hard disk. Users running client Setup do not get a choice.
 - **User's Choice** Users running client Setup choose between Server and Local Hard Drive installation options when they run client Setup.



9. Setup will check for available disk space on the server and then copy all the Office files from the floppy disks or CD to the administrative installation point.

Users can now run client Setup from this administrative installation point and install Office on their computers. Based on the responses you provided while setting up the administrative installation point, you have customized the installation process: When a user runs client Setup from the administrative installation point, Setup will either install Office on the user's local hard disk, or it will configure the user's system to run Office remotely from the server, or it will give the user the choice of running locally or on the server.

This level of customization is generally sufficient for many installations, but there are more elements of the client installation process that you can customize. For a description of all the ways you can modify the administrative installation point to customize client installation, see Chapter 9, "Customizing Client Installation."

Client Installation

Now that you have created the administrative installation point, users can connect to that location, run client Setup, and install Office. This section describes the client installation process.

In the following installation procedure, client Setup is interactive, which means the end user answers questions. Client Setup can also be run in a batch mode, which means the answers to setup questions come from client Setup script files rather than the user. You customize these scripts to provide the appropriate responses. For information about customizing the client Setup process, see Chapter 9, "Customizing Client Installation."

When a user installs from an administrative installation point, client Setup needs to know where to put the main Office application files and the shared application files.

This section describes the client installation process, highlighting the important decision points, but doesn't give all the details of the client installation process.

► **To install on the client computers**

1. Connect to the administrative installation point and access the folder containing the main Office application files.
2. From this folder, run Setup.exe:
3. Setup asks for the user's name and uses the organization name you entered when you created the administrative installation point.
4. Setup verifies the product ID.
5. Setup then asks for the destination folder for Office.

The folder specified here can be on the local hard disk or it can be a network drive to which you have read, write, create, and delete permissions. Even if the software is not going to be installed locally but will be accessed from the server, Setup still creates a folder with some user-specific files in it.

6. If you chose to allow users to select where shared applications are installed, the user selects one of the following:

- **Server** This option leaves the shared application files on the server and you run the files over the network.
- **Local Hard Drive** This option installs shared application files locally on your hard disk.

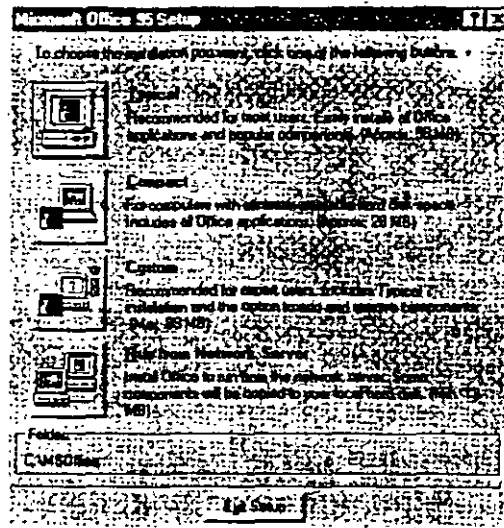
Whether this option was pre-selected during the creation of the administrative installation point or the user selects it, Setup does the following:

- If the option is Server, Setup configures the client system to point to the server share defined during administrative installation (the server location of the shared applications folder).
- If the option is Local Hard Drive, Setup asks for a folder name and copies the shared application files to that folder.

7. Setup asks for the installation type. There are four types:

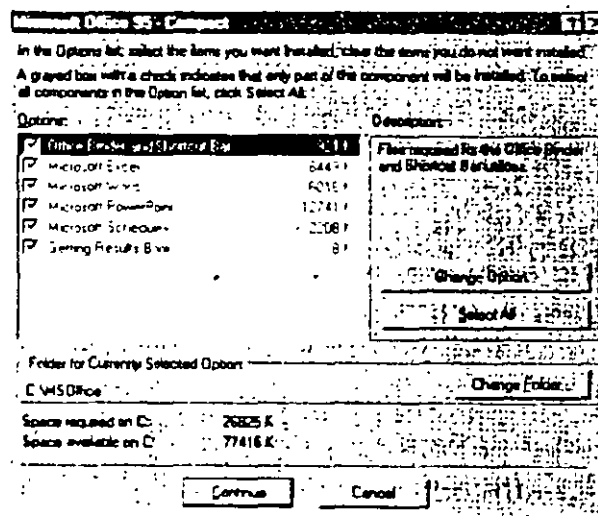
- **Typical** A predefined subset of components are installed representing the most commonly used features of Office.
- **Compact** A predefined subset of components are installed representing the minimal set of files that deliver basic Office features.
- **Custom** The user selects some or all Office components to be installed.
- **Run From Network Server** The main Office application files are left on the server and run from the network.

For a complete list of the options installed for each installation type, see Appendix C, "List of Installed Components."



If Compact is selected, Setup displays the list of Office applications with check boxes. The user can choose which applications are to be installed by setting or clearing the check box for a particular application. The list consists of:

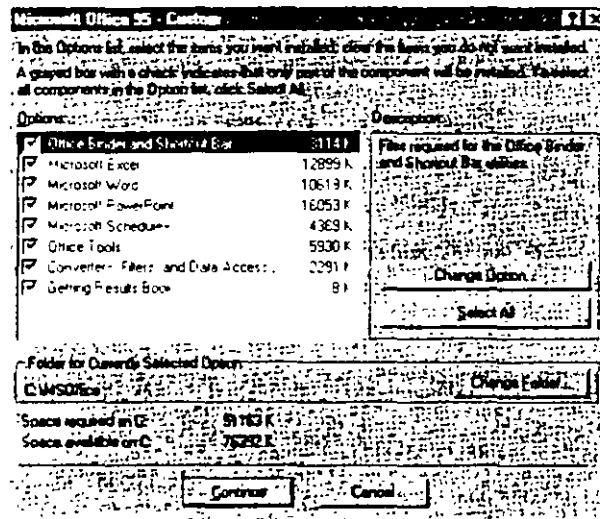
- Office Binder and Shortcut Bar
- Microsoft Excel
- Microsoft Word
- Microsoft PowerPoint
- Microsoft Schedule+
- Microsoft Access (Office Pro only)



The subcomponents installed with each application are predetermined for the Compact installation type and are defined to provide the basic functions of Office using a minimum amount of disk space.

If Custom is selected, Setup displays a series of lists so you can choose which components to install. The first list contains the main Office components:

- Office Binder & Shortcut Bar
- Microsoft Excel
- Microsoft Word
- Microsoft PowerPoint
- Microsoft Access (Office Professional only, not in Office Standard)
- Microsoft Schedule+
- Office Tools
- Converters, Filters, and Data Access
- Getting Results Book



For each of these components, the user can click Change Option to view the list of subcomponents within the component. For example, selecting Microsoft Word and clicking Change Option displays the list:

- Microsoft Word Program Files
- Online Help
- Wizards, Templates, and Letters
- Proofing Tools
- Address Book

- Word Mail
- Dialog Editor
- Text Converters

Each of these subcomponents may also have more subcomponents—if so, Change Option is enabled when the component is selected. The user can continue to walk through all the components for Office choosing which ones to install.

For each component displayed in the list, the check box will be in one of three states:

- **Checked** All subcomponents of this component will be installed
- **Clear** No subcomponents of this component will be installed
- **Gray** Some, but not all, of the component's subcomponents have been selected to be installed; click Change Option to see and change the selected subcomponents.

— — —
The Select All button allows the user to select all the components displayed in the dialog box along with their subcomponents.

For some components, you can change the folder where that component is installed. If the folder can be changed, the Change Folder button is available.

8. If you are running Windows NT, Setup asks for the Program Group for the application icons.

Setup checks for available disk space, and starts copying files. If Setup determines that there is not enough disk space for all the components chosen, you have two choices.

- If you used the Custom installation option, you can specify a different destination folder for some components to install them on another hard disk. Even if you change the folder for every component, Setup will still install some files in the original Office folder you specified.
- Exit Setup and restart, specifying an Office folder on a different hard disk.

Installation Issues

This section describes the format used on the floppy disks and how to extract files from the disks if you need to do so manually. It also describes the product identification method used by Microsoft; explains compliance checking, which verifies that the user has an appropriate older version of Office or competitive product installed before installing Office as an upgrade; and covers the licensing options available from Microsoft.

Copying Disks and Extracting Files Manually

With the exception of the Setup disk (Disk 1), your Microsoft Office disks use a format called DMF (Distribution Media Format). DMF increases the capacity of a 3.5-inch floppy disk, reducing the number of disks needed to install your application and therefore speeding up the installation process.

Warning Because DMF has been only recently developed, many existing disk examination or copying utilities do not recognize DMF and can corrupt the DMF disk. For more information, see "DMF Disk Issues" later in this section.

What Is DMF?

Microsoft Office 3.5" floppy disks are created using a recently developed compression and disk creation technology that reduces the number of floppy disks needed to install Office. The three parts to the new technology are Diamond, Quantum, and DMF.

Diamond A disc layout tool used by Setup that combines multiple files into a single *cabinet* (CAB file) or across multiple cabinets on separate disks. Extract.exe is the tool used for extracting and decompressing a single file from a Diamond cabinet, or for copying Diamond disk images from floppies to the network; it cannot put new or changed files back into a cabinet. Extract.exe is described later in this section.

Quantum New compression algorithm used by Diamond.

DMF Distribution Media Format is essentially a method of storing 1.68 MB of data on a high density 3.5" floppy disk. Because DMF reduces the gap between sectors, there is no way to write to a DMF floppy disk using standard disk copying tools without damaging the floppy disk.

Disk 1 of a DMF disk set is always a normal, 1.44 MB read/write disk so it can be handled using normal disk utilities such as the MS-DOS COPY command.

To create or copy a DMF disk, the DiskWrite utility is used. For Select customers, DiskWrite is available on the Select CD. For MOLP customers and corporate accounts with distribution rights, DiskWrite is available on a fulfillment CD from Microsoft.

DMF has no effect on Setup functionality and is used only for storing software when put on floppy media.

DMF Disk Issues

Because of the way DMF modifies the disk at the sub-sector level, a DMF disk cannot be handled in the same way as a standard MS-DOS disk. Here are some of the issues to keep in mind:

- Many existing utilities such as Norton Disk Doctor, Microsoft ScanDisk, MS-DOS DiskCopy, and Microsoft Windows Copy Disk do not recognize DMF. You should not use disk utilities to examine a DMF formatted disk, as these utilities can corrupt the DMF disk.
- You cannot copy DMF formatted disks using MS-DOS DiskCopy or Microsoft Windows Copy Disk. You must use the tool Extract.exe.
- Because the DMF format reduces the gap between sectors on a disk, low-quality or older disk drives may have trouble reading the disk.
- Any utility, or computer virus, that writes an MS-DOS type of master boot record on the disk will render any DMF data on the disk unusable.

Using Extract.exe

- The preferred method for placing Office files on a network server from which users can run client Setup is to create an administrative installation point. For more information, see "Client Installation Over the Network" earlier in this chapter. But if you need to copy the Microsoft Office disks to a network server or other permanent storage drive, you can use the copy switch (/c) with the Extract.exe utility on Disk 1 to copy the Microsoft Office installation files to the target location.

To see a list of and help about the Extract command options, type `extract /?`

For example, after creating a folder called C:\Disks on your hard disk for the Microsoft Office files, copy all the files on Disk 1 to that folder. Because Disk 1 does not use DMF, you can use the standard MS-DOS Copy command, `copy a:*.* c:\disks`. Switch to drive A and type the following command to copy the rest of the disks to the folder C:\disks:

```
FOR %I IN (*.* ) DO C:\DISKS\EXTRACT /C A:\%I C:\DISKS\%I
```

A cabinet (.cab) file includes many files stored as a single file. If you need only a single file that is contained in one of the cabinet files, you may search for it using the /d switch with Extract.exe. After you find the file, you can use Extract.exe again to make a copy of the file in the new location.

The file setup.inf is a disk information file for Setup that describes the contents of each disk. For a description of the format of this file, see Appendix A, "Setup Command Line Options and File Format."

Here are some examples of how to use the Extract command to find and extract files from a cabinet file.

To	Type this command
List all files in the cabinet file Disk1.cab.	<code>extract /d a:\disk1.cab</code>
List all EXE files in Disk1.cab.	<code>extract /d a:\disk1.cab *.exe</code>
Extract a file named Any.exe from Disk1.cab and copy it to the current folder. If the file spans more than one disk, Extract.exe will prompt you to enter the second disk.	<code>extract a:\Files.cab any.exe</code>
Extract Any.exe from Disk1.cab and copy it to c:\Office.	<code>extract a:\files.cab /l c:\office any.exe</code>

Product Identification

The product ID for Office is the unique identifier for each license of Office. The complete product ID consists of 20 digits and is handled slightly differently depending on the media used to install Office.

Floppy Disk

A portion of the product ID is stamped on the floppy disks at manufacturing time and a portion is generated at the time the software is installed from the floppy disks. The complete ID is stamped into the Office main executable files on the user's hard disk (Winword.exe, Excel.exe, Msaccess.exe, Powerpnt.exe, and Schdp132.exe).

If multiple installations are done from the same set of floppy disks, a different product ID is generated for each installation and is associated with the license for that particular installation. Each product ID consists of a common root (the first 15 digits) that were stamped on the floppy disk at manufacturing time. This common root identifies each installation as coming from the same floppy disk set.

When installing from floppy disks, Setup keeps track of the number of different installations performed from the same floppy disk set and displays a warning if the legal number of installations is exceeded. Selecting Remove All from Setup will not decrement the installation counter. If you hold multiple licenses, you can use a command line option for Setup to specify the number of licenses you have. For more information about this option, see Appendix A, "Setup Command Line Options and File Format."

CD

CDs aren't stamped at manufacturing time as floppy disks are, so a portion of the product ID is printed on the CD package: the user running Setup will be asked to enter that portion manually during the installation process. The ID is stamped into the Office main executable files on the user's hard disk ((Winword.exe, Excel.exe, Msaccess.exe, Powerpnt.exe, and Schdpl32.exe).

Select CD

When installing Office from a Select CD (from the Select customer program), the user is not prompted to enter any product ID information. A portion of the ID is stamped on the CD and includes the product code and a Select ID. The remainder of the total product ID is generated at the time the software is installed and is unique for each installation. Multiple client installations of Office from a single Select CD have a common root in the product ID that identifies each installation as coming from the same Select CD.

Network

Every installation done from the administrative installation point is given a unique, 20-digit product ID with a common root (the first 15 digits). The ID is stamped into the Office main executable files on the user's hard disk (Winword.exe, Excel.exe, Msaccess.exe, Powerpnt.exe, and Schdpl32.exe)

Compliance Checking

Microsoft Office for Windows 95 is available as an upgrade for those who have a previous version of Office, or for those who are moving to Office from a competitive product. If you purchase Microsoft Office for Windows 95 as an upgrade, client Setup performs a *compliance check* to verify that the upgrade is being installed in accordance with the upgrade agreement. It verifies that you have an appropriate older version or competitive product installed before it proceeds with the upgrade installation.

The verification procedure is performed according to a table defined by Microsoft for each specific upgrade and that may vary for each application, for each version, and for each language depending on international competitive markets. The table consists of a defined list of products and product versions that satisfy the compliance check, along with a *signature* for each product that defines how Setup can tell if the product is installed.

The signature can consist of one or more of the following items:

- Product files, including a filename, size, and folders to search. The signature can indicate whether to search one or more default folders only, or to search the entire disk.
- INI file entries, including an .ini filename, key, and value.
- Registry entries key and value.

At the beginning of the client installation process, Setup works through this table, examining the client computer for indications that at least one of the products on the list is installed. The first product it finds satisfies the search and Setup proceeds with the installation. If the table instructs Setup to search the entire disk, Setup display a message to this effect and lists the products it is looking for.

If no products are found from the list, Setup stops and give the user a chance to insert the installation floppy disk for a complying product so that it can be verified. If the user doesn't have such a disk, Setup ends and does not install the upgrade. In some cases, the compliance checking table for an upgrade may specify that Setup can continue without finding a compliant product, in which case Setup displays a warning and then continues with the upgrade. This depends on how Microsoft defined the compliance checking rules for a specific product upgrade.

The following list shows the products that are searched for by the initial release of the upgrade version of Microsoft Office for Windows 95, US version.

Presentation Graphics

Aldus Persuasion v 2.x and 3.0 for Windows
 Harvard Graphics 1.0, 2.0, and 3.0 for Windows
 Harvard Graphics 2.3 and 3.0 for DOS
 Lotus Freelance 1.x and 2.x for Windows
 Lotus Freelance 4.0 for DOS
 PowerPoint 4.0 or earlier
 WordPerfect Presentations 2.0 and 3.0 for Windows
 WordPerfect Presentations 2.0 for DOS

Word Processors

Ami Pro 1.0, 1.01, 1.1, 1.1B, 1.2, 2.0, 3.0, 3.01, and 3.1 for Windows
 Display Write for DOS 1.0, 1.1, 2.0, and 2.1
 First Choice for DOS 1.01
 Legacy for Windows 1.0
 Lotus Manus for DOS 1.0

Word Processors *(continued)*

Lotus Manusc for DOS 2.1
 Mass 11 6C
 Multimate 1.0 and 4.0
 Multimate for DOS 1.0, 3.3, 3.6, and 4.0
 Prof Write+ for Windows 1.0
 ProfWrite for DOS 1.0
 Q&A for DOS 1.0
 Samna Word for DOS IV
 Sprint for Windows 1.01
 Volks W III for DOS 1.0
 Word for Dos 2.0, 3.0, 4.0, 5.0, 5.5, and 6.0
 Word for Windows 1.0, 1.1, 2.0c, 6.0, and 6.0c
 WordPerfect 5.1, 6.0, 6.0A, and 6.1 for Windows
 WordPerfect* 4.0, 4.1, 4.2, 5.1, and 5.2 DOS
 Wordstar 1.0, 1.5, and 2.0 for Windows
 Wordstar 2000 for DOS 3.0, 3.5, and 6.0
 Wordstar for DOS 1.0, 1.5, 5.5, 6.0, 7.0, and 7.0 Upgrade
 Wordstar Pro for DOS 1.0 and 4.0
 XyWriteIII+ for DOS 3.06, 3.52, and 3.56

Spreadsheets

Lotus 1-2-3 for Windows 1.0, 4.0, 4.01, and 5.x
 Lotus for DOS 1-2-3 2.01, 2.2, 2.3, 3.1, and 4.0 Upgrade, 3.4 and 3.4A
 Microsoft Excel 3.0, 4.0, 4.0a, 5.0, 5.0a, and 5.0c
 Multiplan all versions
 Quattro Pro for DOS 1.0, 2.0, 3.0, 4.0, and 5.0
 Quattro Pro for Windows 1.0, 5.0, and 6.0
 Quattro Pro Special Ed 1.0
 Supercalc for DOS 3.1, 3.2, 4, 5.1, and 5.5 B

Databases

Alpha Five
 Approach for Windows 2.1
 Clipper 5.01
 Clipper 5.02c, 5.02b, 5.02a, and 5.02
 Clipper 5.2 DOS

Databases *(continued)*

Datebase 1.1, 4.5, and 4.53 Exec

Dbase 1.0, 1.1, and 1.5

dBASE 5.0 DOS

dBASE 5.0 WIN

Dbase IV 2.0

FileMaker Pro 2.0 and 2.1 WIN

FoxPro 2.0 DOS

FoxPro 2.5, 2.5a, 2.5b, and 2.6 DOS

FoxPro 2.5, 2.5a, 2.5b, and 2.6 WIN

FoxPro 2.6 WIN

Microsoft Access 1.0, 1.1, and 2.0

Microsoft Office Std (all Windows versions including Office 95 Std)

Paradox 1.0, 3.0, 3.5, and 3.5 Update, 4.0 and 4.5

Paradox 5.0 DOS

Paradox 5.0 WIN

Powerbuilder 3.0

Q&A 2.0 and 3.0 DOS

Q&A 4.0 Win (Checking for Q&A or QAWrite)

R:Base 2.11, 3.0, 3.1, and 4.0

R:Base 3.1, 4.0, 4.5, and 4.5+

Superbase for Windows 4 1.3

Superbase4 1.2 and 1.3 WIN

Suites

Borland Office

Microsoft Office (all Windows versions)

Microsoft Works (all Windows versions)

Perfect Office

Smart Suite

Contact Microsoft Product Support for additional help with specific upgrading situations.

Licensing

Each user must have a Microsoft Office license, no matter how your users install Office. One license is provided with each retail package of Microsoft Office.

If you don't want to purchase a complete copy of Office for every user, you can purchase a License Pak. Each License Pak allows you to make and use one additional copy of Office. You don't need to store extra copies of Office or the documentation, and it's less expensive than purchasing a retail package. For quantities of 13 licenses or more, ask your reseller about the Microsoft Open, Variable, and Enterprise licensing options.

Client Installation Example

The following example demonstrates how an administrator can prepare and perform a basic installation of Office. For this example, no customization is needed. If you need to customize setup, see Chapter 9, "Customizing Client Installations."

Corporation 1 is a mid-sized publishing firm that has just purchased licenses to upgrade their 100 users to Windows 95 and Office for Windows 95. The users are well-trained and usually install their own software, although the internal support group can assist them if needed. The administrator believes that all the machines have sufficient hard disk space. Users have had training on the features of Office for Windows 95 and understand the features of Office enough to know what components they should install. As the software was purchased through a volume license agreement, each user does not have a copy of Office disk, and the administrator wants to avoid the time and cost of distributing physical media.

The administrator decides to create an administrative installation point on a central file server, and then the users can run the Office Setup program from there.

Steps for Installation

- Install Office for Windows 95 on the server using administrative Setup
- Inform the users that Office is available for installation, and set a deadline for the upgrade.
- Distribute installation instructions to users including the path to the administrative installation point
- Confirm that users completed the installation process

CHAPTER 12

Support and Troubleshooting

This chapter covers issues for the Office applications that will help you support your users. It also includes information about customizing templates in Office so your users can follow the standards that exist in your organization for creating documents.

Also in this chapter is information about porting 16-bit Office-based solutions to 32-bit.

In This Chapter

Supporting Microsoft Office	309
Supporting Microsoft Excel	320
Supporting Microsoft PowerPoint	330
Supporting Microsoft Schedule+	349
Supporting Microsoft Word	351
Registry Settings and .Ini Files	359
Long Filenames	360
Porting Your 16-Bit Office-Based Solutions to 32-Bit Office	360
Frequently Asked Questions	377

Supporting Microsoft Office

This section includes information for supporting and customizing templates for your organization, creating shortcuts on the Office Shortcut Bar to access the Microsoft Network, and supporting Office Binders.

Supporting Custom Templates

Office 95 has several features that make it easy to create files based on templates. When a user clicks the **Start A New Document** button on the Office Shortcut Bar or clicks **New** on the File menu in any Office application, the **New** dialog box is displayed. This dialog box has tabs listing templates available for the application, or in the case of the Office Shortcut Bar, for all documents. When the user selects a template, a new document is created with the attributes of the template. When the user subsequently saves this document, the user is prompted to name it, and the template remains intact.

You can customize the templates presented in several ways. You can:

- Create templates customized for your site and add them to any of the tabs in the **New** dialog box.
- Add tabs to the dialog box to help users find the customized templates.
- Use shortcuts to add templates stored on network shares to the dialog box. If you have a set of shared templates used within a workgroup, you can designate a workgroup template location for inclusion in the **New** dialog box.

The following sections explain how to add templates to the **New** dialog box and customize this dialog box to help users find the templates they need.

Adding Custom Templates

To add a template to the **New** dialog box, copy it to the appropriate subfolder of the **Templates** folder within the **Office** folder. Templates added directly to the **Templates** folder are listed on the **General** tab in the **New** dialog box. The **General** tab is usually reserved for blank documents.

Office creates the following tabs for the **New** dialog box by default:

Letters & Faxes	Other Documents	Presentation Designs
Memos	Spreadsheet Solutions	Binders
Reports	Presentations	Databases
Publications		

Whenever possible you should add your templates to one of these categories. The **Other Documents** tab is provided as a location for miscellaneous templates.

To add a template stored in another folder or on a network share, create a shortcut to the template in the **Templates** folder. For more information about creating shortcuts, see the Windows 95 documentation or online help.

If you want to list the template on one of the other tabs in the New dialog box, copy it to the subfolder of the Templates folder that has the same name as the tab you want. For instance, you can copy a memo template used in your company to the Memos folder, so that it appears on the Memos tab.

You can create special-purpose templates that run programs other than the Office applications by setting up entries in the Windows registry. For information about creating these entries, see "Template Settings in the Windows Registry" in Appendix B, "Registry Keys and Values."

Adding Tabs to the New Dialog Box

In most cases, using the existing tabs is sufficient, but you can add tabs to the New dialog box if you need them to identify and organize your templates.

To add a tab to the dialog box, create a subfolder in the Templates folder. Give the subfolder the name you want shown on the tab. Then copy your templates to the subfolder.

- For a subfolder to be available as a tab in the dialog box, it must contain at least one template. When a user clicks New on the File menu from an application, the New dialog box displays a tab for each subfolder of the Templates folder that contains at least one template for that application. Clicking the Start A New Document button from the Office Shortcut Bar displays tabs for all subfolders that contain at least one template for any Office application.

If the folder that you want to be available as a tab is on a network share or other folder, create a shortcut to the folder within the Templates folder.

Note Only folders and folder shortcuts directly under the Templates folder appear as tabs. Templates stored in subfolders all appear on the same tab as the parent folder. For example, if you create a folder named Company Forms within the Templates folder, and Company Forms has subfolders that contain templates, all of these templates appear on the Company Forms tab in the New dialog box.

Setting the Workgroup Templates Location

In addition to the Templates folder, you can specify an additional folder location for shared templates used within a workgroup. Templates in this folder and its subfolders become icons on tabs in the New dialog box, along with the templates and folders on the user's local hard disk. Subfolders with the same name in both locations are merged; for example, if both the local Templates folder and the Workgroup Templates folder have subfolders named Memos, all of the templates in both subfolders appear on the Memos tab in the New dialog box. Templates stored directly in the Workgroup Templates folder appear on the General tab in the New dialog box.

Local templates override templates in the Workgroup Templates location. If a template file has the same name in both locations, the New dialog box presents the file from the user's local disk instead of the workgroup version.

► **To set the Workgroup Templates location**

1. Using the right mouse button, click the Office Shortcut Bar, and then click Customize.
2. Click the Settings tab.
3. Under Item, select Workgroup Templates location, and then click Modify.
4. Type the path to the network share designated for shared templates, or click Browse to search for the share.
5. Click OK.

Note The setting for the User Templates location is required. Do not delete this setting when you add a Workgroup Templates location.

Using the Office Shortcut Bar with Microsoft Network Shortcut Files

Users who create shortcuts to Microsoft Network forums on the desktop cannot add these shortcuts to the MSN toolbar by dragging and dropping. Links to MSN forums are .mcc files, but dragging them onto the toolbar creates a .lnk file, which does not appear on the MSN menu in the user's applications.

Instead, have the user copy the .mcc file to the MSN toolbar folder. The default location for this folder is Office\Shortcut Bar\MSN within the Office installation folder. For information about registry settings affecting toolbars and their locations, see "Office Shortcut Bar Settings in the Windows Registry" in Appendix B.

Supporting Office Binders

Office binders provide a means of combining different types of documents into a single storage unit. Documents in a binder preserve their order, open and save together, print together, and are stored as a single file. Each document becomes a section of the binder and can be manipulated using the original application via OLE in-place editing. If needed, a binder can be broken apart into separate documents.

Binders allow users to print the collection of documents, in sequence, with consecutive page numbers, as long as headers or footers are set in each document to print page numbers and the initial page number is not reset within an individual document.

Users create new binders from the Office Shortcut Bar or by using the New Binder command on the File menu. Users add sections to binders by dragging and dropping documents from the file system or by using the commands on the Section menu. The Binder menus provide commands to reorder, rename, and delete sections.

Office Binder includes templates to help you assemble common sets of business documents such as client billings, proposals, and presentations. These templates are stored in the Binders folder of the Template folder.

Note If you have previously installed a Beta copy of Office 95, such as the Office Preview Program, make sure that you and all your users completely remove the early copy of Office and then install Office 95 before you create or use Office Binders.

The following sections provide information to help the administrator support users of Office Binders.

* * * Required Files

— The following DLLs and other files support the Office Binder capability

Filename	Location	Purpose
Docobj.dll	Windows folder\System	Provides document object capability
Explode.dll	Office folder\Office	Provides unbind capability
Bdreg.dll	Windows folder\System	Supports binder registration
Mso95.	Windows folder\System	

Binder Documents, Templates, and Wizards

Binders use the following file extensions:

Extension	File type
.obd	Binder document
.obt	Binder template
.obz	Binder Wizard, a binder template that includes macros

There is no difference in file format among the three file types above; only the extensions and icons are different to indicate how the files are used. A binder file with the .obt extension uses the binder template icon; files with the .obz extension have the Binder Wizard icon.

To create a template from a binder document, users can save the binder as a template using the **Save As** command on the **File** menu, or just rename the file to change the extension from .obd to .obt. To create a new binder based on a template, use the **New Binder** command on the **File** menu. Saving the binder does not overwrite the template. To edit the template itself, you must use the **Open** command on the **File** menu in Office Binder.

You can create binder templates based on sets of documents in use in your organization. You can add your own templates to the files installed on users' computers by editing the setup scripts your organization uses. For information about custom setup scripts, see Chapter 9, "Customizing Client Installations."

Use a wizard to tell users how to use a template or provide information they'll need to complete the project.

► **To create a wizard for an Office Binder template**

1. Write macros that gather information from, or give information to, users of the template. Set up the macros to run whenever the template is opened.
2. Save the macros.
3. Place the document containing the macros in a binder document by dragging its icon into the binder, and then click the icon in the binder to make it the active document.
4. Save the binder document as a template.

Note You can edit macros only in the original document before you place it in the binder. Create, debug, and test your macros first, and then assemble the binder. The macros run when the document containing them is opened.

Binder Templates in the Office Value Pack

If your system uses A4 paper size, you can take advantage of a set of binder templates designed to use your paper size. Run the Setup program for the Office Value Pack to install these templates.

Editing Binder Sections

Users edit a document in a binder by clicking its icon on the left side of the Binder window. The binder then activates the application originally used to create the document. Because the documents in the binder are OLE objects contained in the binder frame, the application menus are combined with the Binder menus. The Binder **File** menu replaces the application **File** menu, and the application file-based commands are not available. Other commands that would normally appear on the application **File** menu appear on the **Section** menu.

In most Office applications, the Help menu includes the selections from both the Binder Help menu and the application Help menu.

Users who prefer to edit a document in its own window, similar to OLE 1.0-style object editing, can use the **View Outside** command on the Section menu. Using **View Outside** provides the usual menus for the application, without the binder selections.

Binder Security

No password protection is available for binders. When a document that is password protected is added to a binder, the binder prompts for the password. However, after the document becomes a binder section, the password protection is lost.

To protect a binder from changes, you can place it on a read-only network share. Users can open and view the binder on the read-only share, but cannot make changes to it or its sections.

Binder-Compatible Applications

For an application to produce documents that can be included in binders, the application must implement the Microsoft document objects interfaces in addition to OLE 2.0. Currently Word 7.0, Microsoft Excel 7.0, and PowerPoint 7.0 documents can be included in binders.

Developers of third-party applications who want to create documents that can be placed in binders must do three things:

- They must participate in the Office-compatible program. For more information about this program, contact Microsoft Customer Support.
- They implement OLE 2.0 in-place editing.
- They must support the required document objects interfaces.

For files that don't support document objects and OLE 2.0, drag and drop, insertion as a section, and dropping the document icon on a binder file in the Windows Explorers are not possible.

Unbinding into Separate Files

Users can break a binder up into its component document files using the **Unbind** command on the shortcut menu. This command creates a separate file for each binder section, using the section name as the filename. The files are copies of the sections and are stored in the same folder as the binder. The original binder remains intact.

Note The **Unbind** command is not available under Microsoft Windows NT.

Users can also save a copy of an individual binder section in a separate file using the Save As File command on the Section menu.

Using Office Binder to Simplify Collaborating on Documents

Briefcase, a file synchronization program included with Windows 95, can transform Office Binder into a multi-user authoring tool. By using Briefcase and Office Binder together, different users can work on different sections in a binder while keeping the other sections current.

► To use a binder with the Briefcase

1. Place a binder file on the network where others can access it.
2. Instruct those using the binder to copy it to their local Briefcase. Each person can then open the copy of the binder in his or her Briefcase and work on individual sections. For more information about Briefcase, see Windows online Help.
3. Each person working on the binder can update his or her copy by clicking Update All on the Briefcase menu. This command copies the individual's work to the binder on the network, and copies any changes in the network copy of the binder to that person's personal Briefcase.

Note This command is not available under Microsoft Windows NT

Normally, Briefcase is installed only with laptop configurations of Windows 95.

► To install Briefcase

1. In the Control Panel, click Add/Remove Programs.
2. Click the Windows Setup tab.
3. In the Components box, check Accessories and then click Details.
4. In the Components box, click Briefcase.

You can also edit your organization's setup script for Windows 95 to include Briefcase as part of all installations. For details, see the *Microsoft Windows 95 Resource Kit*.

Briefcase Reconcilers Keep Files Synchronized

The Briefcase allows for reconciliation of the original file and the file in the Briefcase, updating either file with the most recent version. While the standard reconciliation handler for Windows 95 does this on a file basis, the Office Binder reconciler updates at the section level.

The binder reconciler examines the time stamps on the corresponding sections of the two binders and determines when each was edited. The options possible are:

Section that changed	Reconciliation
Neither	None.
Section in the Briefcase	Section in the Briefcase is copied over the section in the original file.
Section in the original file	Section in the original file is copied over the one in the Briefcase.
Both sections	Reconciler for the application used to create the section is used to merge changes between both sections. If no reconciler is available for that application, the user is prompted to make the decision.

Currently, Microsoft Access 7.0 is the only Office application that provides its own Briefcase reconciler support. Office-compatible third party applications may provide reconciliation handlers; see the documentation for your application for information.

Troubleshooting Binder Problems

The following sections provide information to help the administrator solve problems that users of Office Binders may encounter.

Problems with Printing

Users can print a whole binder, an individual binder section, or a group of sections. The application that created each section does the printing for the section. The binder dispatches each section in sequence, passing control to the originating application for each section and receiving control back as each section finishes printing. The result is a single print job.

The Print Binder dialog box, displayed by the Print command on the File menu, lets users make print settings that apply to the entire binder. In some cases the originating application for a section can override these settings. For example, if a Word document resets page numbering to page 1, that section starts printing at page 1 regardless of the starting page number for the binder. The originating application controls page layout, including orientation, margins, headers, and footers. For page numbers to print, the user must include them on the page in the originating application.

Print preview is not available for binders or individual sections.

The following are solutions for common printing problems.

Binder too large to print

Possible Cause Being collections of documents, binder files can grow quite large. Because a binder is printed as a single print job, it can become too large for the print spooling system.

Solution If a binder is too large to print in one job, print it in sections. Use the Selected Section(s) option in the Print Binder dialog box to print smaller groups of sections, or use the Print command on the Section menu to print one section at a time. Consecutive page numbering should be correct, provided the binder is correctly set up to print with consecutive page numbers.

Page numbers don't print

Possible Cause Page numbers are not enabled for each section in the application that created the document.

Solution In each section, make sure that page numbers are in place in the header or footer. For example, if the section is a Microsoft Excel workbook, the user can use the Page Setup command on the Section menu to add a header or footer that includes the page number.

Wrong page numbers print

Possible Cause When the user selects consecutive page numbering, the binder passes the start page number to the application for the first section. When the section is finished printing, it passes the number of pages printed and the last page number printed back to the binder. The binder passes the next page number to the application for the next section, and continues dispatching until all sections are printed.

If the user resets the page numbering explicitly in the originating application, the binder uses that page number instead of the number passed from the preceding section. For example, if you reset the first page in a Word document to page 1, the binder does not override this setting.

Solution To locate a section that is resetting page numbering, you can print each section individually in sequence, using the Print command on the Section menu, and see how each section is numbered.

Problems with E-mail

Possible Cause Because binder files often grow quite large, sending them through electronic mail may not be feasible, depending on the limitations of your mail system.

Solution Consider alternative ways to distribute a large binder file: place it on a network share, post it to a public folder, or use the Unbind command on the shortcut menu to place the sections in separate files that you can mail or distribute individually.

Controlling File Size

Binders that contain many document files can easily become quite large. To maintain the best possible performance for large binder files, users should use the Save As command from time to time to save the binder as a new file, and delete the old binder file. Doing so defragments the binder file and can reduce its size.

Problems with Linked Objects

Binders do not support external OLE links. You can embed but not link documents in binders. You can, however, create links between two sections of the same binder.

Problems Opening a Binder File or Section

If you are having problems opening a Binder file or section, load the binder, then unbind it into separate files using the Unbind command on the shortcut menu. If that doesn't work, open the individual files. If you identify a problem section, delete that section from the binder.

Problems with Macros Stored in a Section

Users can record and run macros in individual sections by using the macro language for the originating application. Users cannot, however, edit macros that are stored within the sections while the document is in the binder. To edit a macro, save the section to a file using the Save As File command on the Section menu. Edit the macros in the file, and then reincorporate it into the binder.

Problems with Spell Checking

Spell checking across an entire binder is not provided. To spell check a binder, activate and spell check each section individually.

Commands Unavailable on Windows NT Systems

With the Windows NT 3.1 shell, the Unbind command is not available, and the Briefcase reconciler for binders is not available because Windows NT does not include the Briefcase feature. As an alternative to the Unbind command, users can use the Save As File command on the Section menu to save each section out to a file.

Supporting Microsoft Excel

This section describes common problems you or your users may encounter with Microsoft Excel 7.0.

File is locked when you save or close a shared file

Possible Causes

- The file you are saving or closing is open in Microsoft Quick View. Quick View is an application that ships with Microsoft Windows 95 that you can use to view a document without opening it in the application that was used to create the document. When a non-shared file is open in Microsoft Excel, you cannot open the same file in Microsoft Quick View, because the file is locked by Microsoft Excel. However, when a shared file is open in Microsoft Excel, the file is not locked (to allow for sharing) and can be viewed in Microsoft QuickView.
- You opened a shared file in Microsoft Excel, and then set the file attributes to read-only while the file is still open.

Solution

To avoid receiving an error message that the file is locked when you save or close a shared file in Microsoft Excel, you must close the file in QuickView

Open command without SHIFT opens template for editing

When you open a template file (.xlt), the file is opened for editing, even if you do not hold down the SHIFT key. This behavior is different from earlier versions of Microsoft Excel. In earlier versions of Microsoft Excel, when you open a template file without holding down SHIFT, a copy of the template was opened, rather than the template file itself. This change in Microsoft Excel 7.0 is consistent with the way templates act in Microsoft Word.

To create a new worksheet, chart, macro, or workbook file based on a template file in Microsoft Excel 7.0, do the following:

1. On the File menu, click New.
2. Click either the General tab or the Spreadsheet Solutions tab.
3. Click the template file.

Issuing the Shared Lists command results in an error message

The error message "Can't access read-only document *filename.xls*" appears when you click the Shared Lists command on the File menu. The Shared Lists dialog box does not open, and you cannot show the Conflict History in the file when you click Shared Lists on the File menu.

Possible Cause The file is open as read-only and you cannot change the status of the Multi-User Editing feature. When you change the status of the Multi-User Editing feature in a file, Microsoft Excel requires that you then save the file. When a file is open as read-only, you cannot display the Shared Lists dialog box.

Solution Although you cannot display the Shared Lists dialog box in a read-only file, you can display the Conflict History by running the following procedure in a Visual Basic module:

```
Sub Conflict_History()
' In the following commands, replace BOOK1.XLS
' with the name of the file that is open read-only
Workbooks("BOOK1.XLS").Activate
Workbooks("BOOK1.XLS").ShowConflictHistory = True
End Sub
```

The Conflict History is a worksheet that contains information—such as Date, Time, Cell Location, and Value—about conflicting cells in the workbook. Conflict history is created when more than one person edits the same file. To display the Conflict History, do the following (the file must be open with read-write protection):

1. On the File menu, click Shared Lists.
2. Click the Editing tab.
3. Select the Show Conflict History check box. A Conflict History worksheet is inserted in your workbook with the information about the conflicting cells on the worksheet.

Save As Type box contains only the Excel Workbook option

Possible Cause You selected the Allow Multi-User Editing check box in a workbook that has not been saved or is saved in a file format other than Microsoft Excel workbook. This is by design, as you cannot share a file in Microsoft Excel unless it is saved in the Microsoft Excel workbook file format. Additionally, you cannot save a file in Microsoft Excel to a file format other than Microsoft Excel workbook and still maintain the Allow Multi-User Editing feature.

Solution Save the file as a Microsoft Excel workbook file.

Save As dialog box saves the file without multi-user editing

When you save a file that has the multi-user editing feature turned on, this feature may not be turned on in the saved file. This occurs when you use the Save As command on the File menu to save the file, and you change either the file format or the filename. If you change the file format, the multi-user editing feature is not turned on in the new file, because you can only share a Microsoft Excel Workbook file. If you change only the filename, the multi-user editing feature is not turned on in the new file. This behavior is by design, because normally, when you save a shared file, you do so to make your own local copy. Therefore, the multi-user editing feature is not saved to the new copy of the file.

Saving a shared workbook causes unexpected results

You see the Conflict Resolution dialog box when you save a shared workbook and the mouse pointer changes to an hourglass; when this happens, Microsoft Excel may or may not appear to have stopped running.

≠ ≠
— —

Changes you made to the shared file conflict with the changes saved in the same file by another user. When a conflict occurs, the Conflict Resolution dialog box appears, asking how you want to resolve the conflicts. If you click the Use All Remaining Changes From My Session button or the Use All Remaining Changes From The Other Session button, the conflicts are resolved without prompting you for each conflicting cell. If there are many changes to resolve, this process can take a long time, perhaps as long as a few minutes. If you choose one of the Use All Remaining Changes buttons, there is no indication that Microsoft Excel is processing any changes while the conflicts are being resolved. If you press CTRL-ALT-DELETE to display the Close Program dialog box, the text "Not responding" follows Microsoft Excel in the list of programs.

Solution

Switch to another application in Windows while Microsoft Excel resolves the conflicts in the workbook. Windows 95 allows you to work in another application while your original application is processing in the background.

The Save As dialog box does not replace an existing file if it is currently open by another user

When you save a shared workbook using the Save As command, you get a message that the filename already exists and you are asked whether you want to replace the existing file. If you use the Save As method in a Visual Basic for applications procedure to save a shared workbook, you see a message asking if you want to replace the existing file.

This behavior occurs as expected when you save a file to the same location and with the same name as an existing file. However, when you are editing a shared file, this message is misleading because another user may be editing the file. In this case, if you answer Yes to replace the existing file, you are not truly replacing the file. You are still prompted with the Conflict Resolution dialog box if a change you made conflicts with a change saved in the workbook by another user. The Save As dialog box does not replace an existing file if it is currently open by another user.

Solutions

When you use a Visual Basic for applications procedure to save a shared workbook, you must use the Save As command if you want to use the Conflict Resolution argument to determine whose changes to accept in case of a conflict. When you use the Save command instead of the Save As command, you cannot use command arguments to specify information about the workbook, such as any of the following: the filename, whose changes to accept in case of a conflict, whether to save the file exclusively, and so on.

- Click Yes in the dialog box asking if you want to replace the existing file. This method saves the file the same way it does if you click Save on the File menu. The Conflict Resolution dialog box will still appear if there is a conflict with a change you made to the workbook.
- If you are not using a Visual Basic procedure to save the shared workbook, click Save on the File menu, instead of Save As, to avoid receiving this message when you save a shared workbook.

In Microsoft Excel, when you share a workbook, you may make a change to the workbook that conflicts with a change that another user has made to the workbook. The conflicting change appears in the Conflict Resolution dialog box, and also appears on the Conflict History worksheet box when a conflict occurs. A conflict occurs when you enter data in a cell that has already been used. When this occurs, the Conflict Resolution dialog box appears and notifies you that you made a change that conflicts with a change made by another user who has already saved the workbook. The type of change that causes the Conflict Resolution dialog box to appear is displayed in the Your Changes On Sheet box and the Conflicting Changes By box in the Conflict Resolution dialog box, and in the Change column on the Conflict History worksheet. The following list shows the change type values that can appear in these locations:

- Cell Change
- Sheet Rename
- Range Move
- Sort Cells

Commands and features unavailable in a shared workbook

When you use the shared lists feature, some of the commands and features that you normally use in your workbook are not available. This behavior is by design, because of the way that a shared file is saved. For example, you cannot change cell formatting in a shared workbook. The following information describes the features and commands that are not available while you are editing a shared workbook.

The following table contains the menu commands that are unavailable (dimmed) in a shared workbook:

Menu	Command
File	Properties
Edt	Delete Sheet Links Object
Insert	Worksheet Chart (submenu items dimmed) Macro (submenu items dimmed) Function Name (submenu items dimmed) Picture Map Object
Format	Cells AutoFormat Style
Tools	Scenarios Protection (submenu items dimmed)
Data	Subtotals Table Group And Outline (submenu items dimmed) PivotTable®

You cannot use any of the buttons on the Formatting, Forms, Drawing, and Chart toolbars in a shared workbook.

On the Standard Toolbar, the AutoSum, Function Wizard, Chart Wizard, Map, and Format Painter buttons are disabled.

Formulas When you enter a formula on a worksheet in a shared workbook, you receive a message that new or copied formulas won't be saved while the file is being shared. You can turn this message off to prevent it from repeating on the current computer for the current session.

Objects You cannot select any objects on a worksheet in a shared workbook, including OLE objects, drawing objects, and dialog box controls.

Visual Basic Modules You cannot edit Visual Basic for applications modules in shared mode; however you can run a macro on a module sheet in a shared workbook. If you record a new macro in a shared workbook, the macro is recorded in a new workbook. In the Record New Macro dialog box, the New Workbook option is selected by default. This Workbook option is unavailable in a shared workbook.

Chart Sheet or Dialog Sheet Although you can select a chart sheet or dialog sheet in a shared workbook, you cannot select any of the items on the chart or custom dialog box.

Macro Sheet You can select a macro sheet in a shared workbook. However, because commands on a macro sheet generally begin with an equal sign (=) (as when you enter a formula), you receive an error message. You can run a macro from a macro sheet in a shared workbook.

Unable to insert a cell not in a shared workbook

You cannot insert a cell note in a worksheet when the workbook you are using is shared. When you use the shared lists feature, the Note command on the Insert menu is available and you can enter text in the Text Note box in the Cell Note dialog box. However, this text is not saved in a cell note, and the error message that a cell cannot be inserted appears. Clicking OK closes both the error message and the Cell Note dialog box without saving the entered text.

New sort order in Microsoft Excel 7.0

In Microsoft Excel 7.0 and Microsoft Excel 5.0 for Windows NT (32 bit), the order used for sorting is different from that used for Microsoft Excel 5.0 for Windows and the Macintosh (16 bit). This occurs because Windows 95 and Windows NT use a sort order different from the Microsoft Windows version 3.x operating system. The differences in sorting are described in the following sections.

Special Characters The following table displays the differences in the sort order of special characters.

Microsoft Excel 7.0 Microsoft Excel 5.0 for NT	Microsoft Excel 5.0 for Windows and the Macintosh
-	!
!	#
#	\$
\$	%
%	&
&	(
()
)	+
*	-
.	~
?	.
@	=
\	?
^	@
-	\
+	^
=	-

Extended Characters When you sort extended characters (used, for example, in international versions) in Microsoft Excel 5.0 for Windows NT or Microsoft Excel 7.0, the international characters are sorted as US English alphabetic characters. In Microsoft Excel 5.0 for Windows or the Macintosh, extended characters are sorted before alphabetic characters. The following table displays some of these differences.

List	Result in Microsoft Excel 7.0 and 5.0 for NT	Result in Microsoft Excel 5.0 for Windows or the Macintosh
A1: b	A1: b	A1: e*
A2: e*	A2: *	A2: b
A3: f	A3: f	A3: f

* Lowercase e with the Umlaut character (ALT+0235)

List	Result in Microsoft Excel 7.0 and 5.0 for NT	Result in Microsoft Excel 5.0 for Windows or the Macintosh
A1: a	A1: a	A1: â**
A2: â**	A2: â**	A2: ê***
A3: e	A3: b	A3: a
A4: ê***	A4: e	A4: b
A5: b	A5: ê***	A5: e

** Lowercase a with the Caret character (ALT+0226)

*** Lowercase e with the Caret character (ALT+0234)

Case Sensitive When you select the Case Sensitive check box in the Sort Options dialog box (Ascending order, Top To Bottom), in Microsoft Excel 5.0 for Windows NT and Microsoft Excel 7.0, your list is sorted with the lowercase entries at the top of the list. Under the same conditions in Microsoft Excel 5.0 for Windows, the list is sorted with the uppercase entries at the top of the list. The following table displays some of the differences.

List	Result in Microsoft Excel 7.0 and 5.0 for NT	Result in Microsoft Excel 5.0 for Windows or Macintosh
A1: A	A1: a	A1: A
A2: AAA	A2: A	A2: a
A3: a	A3: AAA	A3: AAA
A1: Add	A1: add	A1: ADD
A2: ADd	A2: Add	A2: ADd
A3: ADD	A3: ADD	A3: Add
A4: add	A4: ADD	A4: add

DataMap objects do not automatically update

DataMap is a new OLE object included with Microsoft Excel 7.0 that allows a geographical representation of data. Unlike other objects, such as charts, DataMap objects do not automatically update when associated data is changed, due to the potentially prohibitive amount of time required to update DataMap objects associated with large amounts of data.

Solutions

- Manually update by activating the object and clicking the Update button.
- Programmatically update through a Visual Basic for applications procedure. The following example outlines a Microsoft Excel Visual Basic for applications macro that updates a DataMap object on a worksheet.

```
Sub Update_DataMap_Object()
'Assign DataMap Object Name to a Variable
MapName = "Picture 1"
'Update DataMap Object
ActiveSheet.OLEObjects(MapName).Object.Refreshmap
End Sub
```

You receive the error message “A column of data in your selection must contain geographical data” when creating a DataMap object based on ZIP Codes

Possible Cause

A column of non-formatted zip codes is selected and you insert a DataMap object.

Solution

Format the numeric postal codes (zip codes) with the special ZIP Code number format. This format is designed to prevent the possible misinterpretation of numeric data as ZIP Codes.

DataMap Objects Are Programmatically Limited

The only Visual Basic command available for DataMap Objects is the RefreshMap command. This command updates a DataMap Object on a worksheet. The following example illustrates the use of the RefreshMap command:

```
ActiveSheet.OLEObjects("Picture 1").Object.
```

You receive the error message “You must select cells from the workbook *workbook name*” when using the Template Wizard

When you use the Template Wizard to create a template, if the name of the workbook from which you create the template contains an apostrophe character ('), you receive this error message. In Microsoft Excel, the apostrophe character is reserved for use in references. This problem does not occur if the name of the template or the database that you create using the Template Wizard contains an apostrophe character. However, as a general rule, to avoid problems with linking, you should not use the apostrophe character in a workbook or worksheet name in Microsoft Excel.

Solution

Rename your workbook to a name that does not contain an apostrophe, and then use the Template Wizard to create a template from this file.

Array formulas return #NUM error value**Possible Cause**

You are using an array greater than 5458 elements in a function. The maximum array size is 5458 elements (approximately 75 rows by 75 columns) in Microsoft Excel 7.0. In Microsoft Excel 3.0, 4.0, and 5.0, the maximum is 6553 elements. In Microsoft Excel 7.0, the maximum array size is still 64K, as it is in earlier versions of Microsoft Excel. However, because Microsoft Excel 7.0 is a 32-bit application and uses 12-byte data types (instead of 10-byte data types found in the 16-bit versions of Microsoft Excel), the maximum array size in Microsoft Excel 7.0 holds fewer elements than the earlier versions of Microsoft Excel. The following is a list of some of the functions in Microsoft Excel that use arrays:

- LINDEX()
- MDETERM()
- MINVERSE()
- MMULT()
- SUMPRODUCT()
- TRANSPOSE()
- TREND()

Solution

Recalculate your function with a smaller array.

Microsoft Excel 5.0 file size larger after saving in Microsoft Excel 7.0

If you open and save a file in Microsoft Excel 7.0 that you created in Microsoft Excel 5.0, the file size increases by 4 to 6K. Even though Microsoft Excel 7.0 uses the same file format as Microsoft Excel 5.0, this occurs because of an optimization in Microsoft Excel 7.0 that allows Microsoft Excel to save files faster than in the previous version. When you create and save a file in Microsoft Excel 7.0, the file is not necessarily larger than the equivalent file that you create and save in Microsoft Excel 5.0.

Run-time error 438: Object doesn't support this property or method

When you use a Visual Basic for applications procedure to create an Office Binder OLE Automation object, and you edit a Microsoft Excel worksheet object in the Binder, you may receive the error message that the object doesn't support the property or method.

Possible Cause

This error message occurs when you edit a Microsoft Excel worksheet object in a Binder using OLE Automation if you attempt to access a single sheet (such as a worksheet) or an item on a single sheet (such as a cell value) from outside the Application object without using the Parent object. This occurs because the Microsoft Excel object is a worksheet, even though the section in the Binder is a workbook.

Solution

Use the Parent object to access a Microsoft Excel workbook object, and then use the property or method of the Worksheet object as in the following example: The following command adds a worksheet to section 1 (a Microsoft Excel workbook) in the Binder MyBinder:

```
MyBinder.Sections(1).Object.Parent.Worksheets.Add
```

Supporting Microsoft PowerPoint

This section describes supporting some new features in PowerPoint, including the AutoContent Wizard, the Pack and Go Wizard, using and managing ClipArt in PowerPoint and other applications, customizing the style checker, and setting up self-running presentations. At the end of this section are common issues you or your users may encounter with PowerPoint 7.0.

The AutoContent Wizard

- — The AutoContent Wizard creates a presentation and suggests an agenda for a variety of different topics, such as "Communicating Bad News" and "Reporting Progress." Some topics are only available after clicking the Other button in the AutoContent Wizard. For many topics, the Wizard asks questions about presentation length, style, type of output, printing, and customization. When you click the Finish button, the Wizard creates a pre-formatted presentation with a suggested agenda based on the selected topic. Each slide includes suggestions about when, where, and how to present the information. You simply replace the tips with the content of your presentation.

Note The starter presentation templates used by the AutoContent Wizard are stored in the \MSOffice\Templates\Presentations folder. You can also access them using the New command on the File menu, or the Start a Document button on the Office Shortcut Bar.

You can create new AutoContent templates for your users and make them available through the wizard by storing them in the \MSOffice\Templates\Presentations folder. Make sure to save the presentation in template format. Choose the Save As command on the File menu and select the Presentation Template option in the Save As Type list. To access these templates in the AutoContent Wizard, click the Other button when the Wizard displays a list of presentation types to choose from. The Select Presentation Template dialog box appears, which displays the contents of the Presentations folder.

When you use the Other button to select a custom template, the subsequent steps in the Wizard that provide options regarding length, style, and output are not displayed. Thus, the only element that the AutoContent Wizard is able to control in a custom template is the creation of the title slide.

If you want a custom title slide, make sure you create one when you create a template for use with the AutoContent Wizard. If you don't, the Wizard automatically inserts one for you, formats it using the default title slide format, and applies the master formatting to the background. If you want a custom look for the title slide, include one in the template and format it the way you want. Because the title slide already exists, a new one is not inserted. Custom title slide text entered in the AutoContent Wizard is added to the existing title slide without disturbing the formatting.

The Pack And Go Wizard

Presentations are frequently created and then given on different computers, sometimes by different people. The Pack and Go Wizard assembles all the files required to give the presentation and ensures that they're all bundled together. The Wizard clones the presentation, copying the presentation file and all of its source files to a destination selected by the user.

Tip You can pack multiple presentations by holding down CTRL while selecting files in the Pack and Go Wizard's Browse dialog box.

Here's how the Pack and Go Wizard works:

- When packing the active presentation, the Pack and Go Wizard saves a copy of the presentation to the Windows\Temp folder. If the presentation has not yet been saved, it is saved as "Untitled.ppt."
- The temporary copy of the presentation is loaded into memory. (If the Other Presentations option is selected in the Wizard, the original file is loaded into memory.) If your presentation contains links to two or more files with the same name but stored in different folders, the Wizard renames the second and subsequent files using the Windows GetTempFileName API function. This is done because all files packed by the Pack and Go Wizard are unpacked together in the same folder at the destination.
- If the Embed TrueType Fonts option is selected in the Wizard, a flag is set in the presentation to embed the fonts. The font embedding code is identical to the code used when Embed TrueType Fonts is selected in the Save As dialog box.
- If the Package Links option is selected, the Wizard looks through the presentation for linked files. When a linked file is found, that file is copied to the Temp folder and the link source is reset accordingly in the presentation. Because both the presentation and the link source are in the same folder when the link source is reset, the link is preserved after the files are moved to another folder.

- If the Wizard detects an embedded Media Player object, the user is asked to browse for the linked file. (Embedded Media Player object links are not maintained by OLE. The Wizard must ask for user intervention to identify the source.)
- The presentation is saved to the folder used as the temporary folder (for example, Windows\Temp), ensuring that the original presentation remains untouched. If the original presentation was stored in the temporary folder, the new copy will overwrite it. All files that are copied to the temporary folder automatically overwrite existing files of the same name.

Note The location of the temporary folder is normally specified in the Autoexec.bat file, if one exists. By default, Windows 95 sets \Windows\Temp as the temporary folder.

- If the Include PowerPoint Viewer option is selected, the Wizard checks to see which version of the PowerPoint Viewer is installed. The latest version of the Viewer will always be packed with the presentation. When you unpack the Viewer, you don't need to worry about installing an older version over a newer version; the Wizard checks for this.
- The PNGSetup.ex_ file is copied to the destination disk and renamed to PNGSetup.exe. PNGSetup.exe is used to unpack the presentation.
- A function in the PNGComp.dll file is called to create a cabinet file on the destination disk. Cabinet files are compressed files that are designed to span multiple disks. Each cabinet file is the same size, and the size is calculated based on the amount of space remaining on the destination disk at the time the first cabinet file is created. Thus, when packing to floppy disks, the space remaining on Disk 1 will determine how much space is used on all subsequent diskettes. If the user is packing to a hard disk or network location, only one cabinet file is created, and the size limit is determined by the remaining space on the disk. Make sure, when packing to floppy disks, that Disk 1 is clean, to ensure that the minimum number of disks are necessary.
- In addition to the presentation, linked files, and the PowerPoint Viewer files, the Wizard also packs a text file called Playlist.lst that contains the names of all the presentations that were packed. This list is used to run the slideshow feature from the PNGSetup program. (This playlist file has no relationship with the playlist files used with the 16-bit Viewer.)
- Once all files are copied into the cabinet files, the temporary files are removed from the Temp folder and the working copy of the presentation is removed from memory. The Pack and Go Wizard always leaves PowerPoint in the same state as when it started.

Note An updated 32-bit version of the PowerPoint Viewer application will be available after PowerPoint 7.0 ships. The new version will support all of the new PowerPoint 7.0 electronic presentation features. To obtain a copy, contact Microsoft Customer Service, download it from the Microsoft Network, the Microsoft forum on CompuServe, or Microsoft Select. For more information, see the list of resources in Appendix F, "Resources."

Setting Up a Packed Presentation

To give a presentation that has been packed with the Pack and Go Wizard, the user must first decompress the packed presentation and its linked files. The Pack and Go Wizard provides a setup program to decompress the presentation files and install them to a specified directory.

► To unpack a presentation

1. Insert the disk containing PNGSetup.exe, or navigate to PNGSetup.exe in the destination folder.
2. Double-click PNGSetup.exe
3. Specify a destination folder for the unpacked presentation. If the folder doesn't exist, PNGSetup will create it.
4. If PNGSetup detects that PowerPoint or the PowerPoint Viewer is installed when the presentation is unpacked, it offers the option of running the slideshow immediately. If you choose this option, PNGSetup launches the slideshow and shows each presentation (if more than one was packed) in sequence.

Removing an Unpacked Presentation

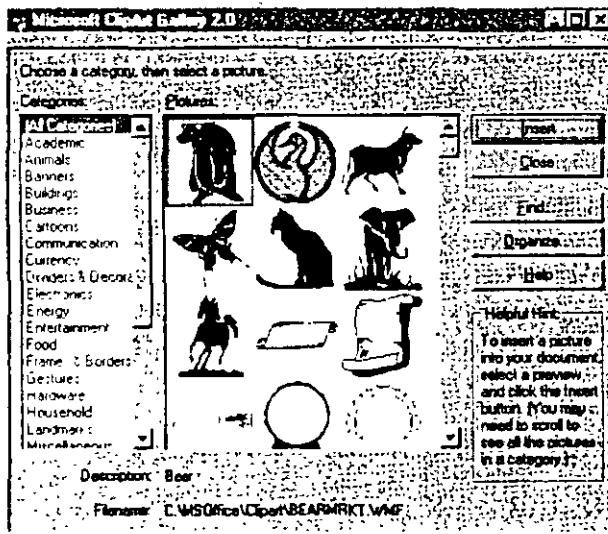
To remove the presentation from the computer when you're finished with it, delete the folder to which it was unpacked. Except for the Viewer files, all files installed by the Pack and Go Wizard are installed to this folder.

Note If any objects in a packed presentation have Interactive Settings applied, these settings will be preserved, but any programs that are specified in the Run Program edit box in the Interactive Settings dialog box will not be packed with the presentation. The user must ensure that these programs are installed on the destination system.

Clip Art in PowerPoint

Microsoft ClipArt Gallery 2.0 is an OLE tool that ships with Microsoft Office. You activate the ClipArt Gallery by clicking the Clip Art command on the Insert menu in PowerPoint, or by clicking the Object command on the Insert menu and double-clicking Microsoft ClipArt Gallery in the Object Type list. ClipArt Gallery displays graphics stored in Picture Store files. By default, the Microsoft ClipArt Gallery application is installed in the \Program Files\Common Files\Microsoft Shared\Artgalry folder. By default, the Picture Store files are installed in the MSOffice\Clipart\Pcsfiles folder.

The version of PowerPoint supplied on floppy disks includes a selection of clip art spanning a variety of categories. The version of PowerPoint supplied on CD provides additional clip art in the Office Valupack directory.



Using Clip Art in Other Applications

ClipArt Gallery .psc file images are not necessarily available in individual graphic formats, such as Windows Metafile (.wmf) or Tagged Image File Format (.tif). Therefore, it might not be possible to directly insert ClipArt Gallery images into Office documents by clicking Picture on the Insert menu. To access, view, and insert graphic images from the .psc files, you must insert the image as an object from the ClipArt Gallery. On the Insert menu, click Object. In the Object Types list, choose Microsoft ClipArt Gallery. Select the image you want, and click OK.

You can also copy a ClipArt Gallery image already inserted into one document and paste it into another document.

Managing the Clip Art Gallery

Click the **Organize** button in the ClipArt Gallery dialog box to display the Organize dialog box. Here you can add or update pictures in the gallery. When you add art to the gallery, a thumbnail sketch is created and added to the category of your choice. With the **Add Pictures** button, you can specify particular art files to add. Many formats are recognized, including .wmf, .tif, .eps, and .bmp, as well as .cag (ClipArt Gallery 2.0 database format) and proprietary formats such as CorelDRAW® and Kodak® Photo CD, among others. The **Update Pictures** button searches your computer for readable picture files, and updates the gallery to include all the pictures it finds. This button is generally not a good choice if you have artwork installed that you don't want to include in the gallery.

Note The CD releases of Microsoft Office and PowerPoint include a ValuPack directory containing additional clip art you can install using the methods described above (the floppy-disk release of PowerPoint includes a separate CD with similar contents). You can install the files from the Clipart folder onto your hard disk, or you can conserve space by leaving the clip art files on the CD. If you install the clip art files on your hard disk, you must copy both the .jps and .cag files to the same folder. If you decide to leave the art on CD, you need to insert the CD when you insert clip art from the Office ValuPak.

To change the categories in which pictures appear in the gallery, to create new categories, or to change the text description of pictures in the gallery, click the right mouse button on the thumbnail image in the ClipArt Gallery dialog box and click the **Picture Properties** command. To delete a picture, click the **Delete Picture** command. To Delete or rename the categories themselves, click the right mouse button on the category name in the **Choose A Category** list and click the **Edit Category List** command.

Note When you add or remove pictures in the ClipArt Gallery, information such as category, thumbnail sketch, and description is updated in the master clip art database file, named **Artgalry.cag**, located in the Windows directory.

Installing ClipArt Gallery 2.0 Over ClipArt Gallery 1.0

Previous Versions of ClipArt Gallery created .idx and .thm database files, containing index information, category information, and thumbnail art used by the ClipArt Gallery to locate and organize the actual clip art files. ClipArt Gallery 2.0 creates a single database file to accomplish the same objectives, identified by the extension .cag. If you install ClipArt Gallery 2.0 over a previous version, the .idx and .thm files are merged into the new .cag file, and then they are deleted.

Note When you install PowerPoint 7.0 over PowerPoint 4.0, settings in the `Artgalry.ini` file used by ClipArtGallery 1.0 are transferred to the Windows registry, to be used by ClipArt Gallery 2.0.

Relocating and Renaming .Cag Files

By default, each user's .cag database file is named `Artgalry.cag`, and is created in the user's `\Windows` folder. You can specify where the ClipArt Gallery creates this file by adding a new string value in the Windows registry, to the following key:

```
HKKEY_LOCAL_MACHINE\Software\Microsoft\ClipArt Gallery\2.0
```

The string value name should be "Database", and the value should be a string containing the full path and filename for the ClipArt Gallery database file. For example, setting this value to

```
"C:\Program Files\Common Files\Microsoft Shared\Artgalry\My Database.cag"
```

would cause ClipArt Gallery to call the database "My Database.cag," and to store it in the same folder with the ClipArt Gallery application. If you choose to customize this property, you should ensure that the folder you specify exists, is writable, and is exclusive to the user (i.e. not shared). This value must be set before ClipArt Gallery is launched for the first time.

Installing Clip Art on a Server

To conserve hard-disk space on users' machines, the ClipArt Gallery .pcs files (installed in the `Msoffice\Clipart\Pcsfiles` folder), and any additional art files you want to add to the gallery, can be installed on a network server. The .cag files created by the ClipArt Gallery must be stored on the local machine, or in a private user directory on the server. These files cannot be shared.

Some clip art packages come pre-indexed with their own .cag files. For example, the Cartoons package on the PowerPoint ValuPak CD includes two files: `Cartoons.pcs` and `Cartoons.cag`. When this package is imported into the ClipArt Gallery, the .cag file is merged with the user's master .cag file, then deleted from the hard disk. Until a .cag file is merged, it must be stored with the associated clip art images. If you do not want the ClipArt Gallery to delete the .cag file after importing it (if it is stored on a server where other users need access to it, for example), simply make the folder where they are stored read-only.

For information about setting up a custom installation, see Chapter 9, "Customizing Client Installations."

Optimizing Custom Artwork for the ClipArt Gallery

You can add custom artwork such as company logos to the ClipArt Gallery. Use the following guidelines when creating art for the ClipArt Gallery:

- Use a vector graphic format such as .wmf (preferred) or .cgm. These formats are readily scalable and have transparent backgrounds, allowing the art to “float” when placed on a colored background. Don’t use raster graphic formats such as .bmp or .tif. These formats save background pixels as well as the main body of the art, causing a “white box” effect when placed on a colored background.
- Make sure that you use colors that work well on all the computer displays used in your workgroup. For example, if the lowest-common-denominator is 16-color VGA, optimize the art for that display format.
- When art is added to the ClipArt Gallery, a thumbnail is created for display in the ClipArt Gallery dialog box. This thumbnail uses the display settings of the computer that is used to add the clip art to the gallery.

How AutoClipArt Works

The AutoClipArt command on the Tools menu suggests specific clip art you can add to a presentation, based on the text contained in the presentation. This feature uses a search mechanism that looks at every word in the presentation and finds any matches against a master list of concepts that PowerPoint stores internally. If a match is found, AutoClipArt suggests one or more pieces of clip art for each slide in the presentation that contains text matching the internal concept list. The following guidelines apply to AutoClipArt:

- You must first run the ClipArt Gallery and import the *PowerPoint Popular Clip Art* package, which is supplied with PowerPoint, before AutoClipArt will work.
- The presence of the AutoClipArt feature is indicated by the appearance of the AutoClip key in the following Windows registry location:
HKEY_CURRENT_USER\Software\Microsoft\Office\PowerPoint\7.0\Addins
- AutoClipArt uses the value stored in the following key in the Windows registry to locate clip art files on disk.
HKEY_CURRENT_USER\Software\Microsoft\Office\PowerPoint\7.0\AutoClipArt
- AutoClipArt requires the Microsoft ClipArt Gallery version 2.0, whose presence is indicated by the appearance of the following Windows registry key:
HKEY_LOCAL_MACHINE\Software\Microsoft\ClipArt Gallery\2.0

For more information about the Windows registry, see Appendix B, “Registry Keys and Values.”

Making Your Own Art Available to AutoClipArt

By adding concept keywords to images you add to the ClipArt Gallery, you can make them available for AutoClipArt searches. To do so, activate the ClipArt Gallery, click the Organize button, and then click the Picture Properties button. Type the concept keywords into the Description box. Follow these guidelines for adding concept keywords:

- You can add multiple concept keywords to each piece of clip art.
- The first entry you type in the Description box should be a space character. For example, if you type "war" as the first entry in the Category box for a piece of art, AutoClipArt might suggest the picture when searching for the keyword "reward." Adding a space before the word (" war") ensures that the search would use only the entire word.
- You must type a single space character between each concept keyword.
- Do not use punctuation.
- Use only valid concept keywords, which are listed in the file named "AutoClipArt Concepts.txt" in the \Documentation\PowerPoint folder on Office Resource Kit CD. Keywords other than those listed here are not recognized.

Customizing the Style Checker

You can specify a number of different style and punctuation preferences that the PowerPoint Style Checker can look for when reviewing a presentation. For example, you can specify that all body text should end with a period (or with any other character). If the Style Checker finds a body text item that does not end with a period, you can have the Style Checker add one automatically.

To set these options, click Style Checker on the Tools menu and click the Options button. The settings in the Style Checker Options dialog box are preserved in the Windows registry. For more information about the specific registry settings, see Appendix B, "Registry Keys and Values." These options can also be set during installation using an installation script. For more information, see Chapter 9, "Customizing Client Installations."

Setting Up Self-Running Presentations

Presentation kiosk is a term used to describe a special presentation implementation where a presentation runs in an unattended setting, such as a trade-show booth or an information desk. Presentation kiosks can be interactive, if allowing the audience to have access to the keyboard is acceptable, but you can also hide the keyboard and mouse to prevent interaction and possible disruption of the presentation.

The keys to creating a presentation kiosk are using the slide timing and presentation looping features. You can set options for kiosk presentations so that the slides change automatically and the presentation repeats continuously. First, click the Rehearse Timings button on the Slide Sorter toolbar in Slide Sorter view. This starts the slide show with the Rehearsal dialog box visible, shown in the following illustration.



With the Rehearsal dialog box visible, go through your presentation slide by slide and click the Next button (with the right-pointing arrow) when you want to advance to the next slide. PowerPoint records the amount of time the slide was displayed. Be sure to allow plenty of time to read the entire slide. When you have gone through the entire presentation, PowerPoint returns to Slide Sorter view, where the timings you recorded appear below the corresponding thumbnail for each slide.

Tip In the Rehearsal dialog box, instead of waiting for the time to elapse for each slide, you can type the amount of time you want in the edit box.

Next, choose Slide Show from the View menu, click the Use Slide Timings and Loop Continuously Until ESC options, so that the presentation will repeat indefinitely. Click OK to start the slide show, and then press ESC. Then save the presentation. This way, the options you set in the Slide Show dialog box are preserved in the presentation file.

Creating a Special Shortcut Icon

You can create a duplicate PowerPoint shortcut icon that will automatically start PowerPoint and launch a slideshow with no further user interaction. This can be handy for presentation kiosks where you want to allow audience interaction using the keyboard.

Tip For interactive kiosks, set up the kiosk computer with the special shortcut icon displayed in the center of the screen and hide the taskbar. Click the Windows Start menu, choose Settings, Taskbar command, and click the Auto Hide option. Give the icon a name like "Double-click to start." Then, if an audience member somehow shuts down the program, getting the kiosk up and running again is no problem.

First, you should create a duplicate PowerPoint program icon to use specifically for this purpose.

► **To create a shortcut icon that automatically starts a slideshow**

1. In the MSOffice\Powerpnt folder, select the PowerPoint program icon (Powerpnt.exe) and click the right mouse button to display the shortcut menu.
2. Click Create Shortcut.

A new icon appears, entitled "Shortcut to Powerpnt." Rename it using the Rename command on the shortcut menu.

3. Click the right mouse button on the new shortcut icon and click Properties on the shortcut menu.

In Windows NT, click Properties on the File menu.

4. Click the Shortcut tab.

5. In the Target edit box, after the path string, type a space then /s and then type the name and path of the presentation file you want to start. For example, the complete string should resemble the following:

```
C:\MSOffice\Powerpnt\Powerpnt.exe /s c:\My Documents\MyKiosk.ppt,
```

≡ ≡

In Windows NT, edit the string in the Command Line edit box.

--

6. Click OK.

After you create the special shortcut, double-clicking the icon or choosing it from the Start menu starts PowerPoint directly in slideshow view and starts the indicated presentation. All evidence of the PowerPoint program, including the user interface and the startup logo screen remain hidden. Clicking the ESC key in this mode not only stops the presentation, but also exits PowerPoint.

The Special Slide Show File Type

Another way to directly launch a presentation is to rename the three-character extension of a presentation file. When you change the extension from .ppt to .pps, and then click the presentation file, the presentation launches directly into a slide show.

Using Batch File Lists

You can create a batch file list if you want to show more than one presentation at a time. The presentations listed there are displayed as one continuous slide show without interruptions. This feature can be useful with kiosks, allowing you to string together different groups of presentations for different audiences.

To create a batch file list, start Microsoft Word, Notepad, or any other word processor. Type the filenames of each presentation you want to show on a separate line. Then, save the document as a text file, but give it a .bat extension, and save it in the same folder along with the presentations it opens.

To use the batch file list, click Open on the File menu in PowerPoint, and double-click the batch file list document.

Using Playlists With The 16-Bit Viewer

You can also use batch files with the 16-bit PowerPoint Viewer (shipped with PowerPoint 4.0 and the initial release of PowerPoint 7.0). Instead of naming the text files using .bat extensions, use the .lst extension instead. The files are otherwise identical to batch files described above.

To use a playlist with the PowerPoint Viewer, double-click it in the Windows Explorer, select the playlist file, and click the Show button.

Note An updated 32-bit version of the PowerPoint Viewer application will be available after PowerPoint 7.0 ships. The new version does not support playlists. To obtain a copy of the updated Viewer, contact Microsoft Customer Service, download it from the Microsoft Network, the Microsoft forum on CompuServe, or Microsoft Select. For more information, see the list of resources in Appendix F, "Resources."

PowerPoint Command-Line Startup Switches

-- The following startup switches can be added to end of the Target string in the Windows 95 Properties dialog box (or the Command Line box in Windows NT) to cause specific things to happen whenever you start PowerPoint. In each case you must type a space before and after the switch, followed by the full path to a file.

- /s Starts PowerPoint in Slide Show view and runs the specified presentation or presentations. For more information, see "Using Batch File Lists" earlier in this section
- /p Prints the specified presentation.
- /n Creates a new presentation using the specified presentation as a template
- /i Creates a new presentation by inserting the slides of the specified presentation

Files Created Using Beta Releases of PowerPoint 7.0

Because PowerPoint 7.0 uses a file format that is different from that of previous versions of PowerPoint, and because the file format itself underwent incremental changes throughout the beta release cycle, you might experience problems with presentations created with a beta release of PowerPoint 7.0. Problems are more likely to occur in a presentation that was created in one beta version of PowerPoint, and then edited and saved in later beta releases. Some problems that might indicate a file-format problem include the following:

- Embedding Word tables produces unpredictable results.
- In Outline view, selecting, dragging, or reordering slides sometimes produces unpredictable results

- In Notes view, slide images disappear or are incorrect.
- In Notes printouts, slide images are incorrect.
- When printing, bullets are repeated on a particular slide.

To alleviate these problems:

- Everyone in your workgroup should be using the final release version of PowerPoint 7.0. Do not allow a person who has the final release and another who is still using a beta version to work on the same presentation. This can cause file corruption, to the extent that the file may not even open in PowerPoint 7.0.
- If you notice unusual behavior with presentations that were created using beta versions of PowerPoint 7.0, create the files again using the final version. When you do so, you can copy embedded objects and paste them into the new presentation, but don't copy entire slides or groups of slides.

PowerPoint 7.0 Printing Issues

-- Printing in PowerPoint is somewhat different from printing in the other Office applications. Specifically, PowerPoint:

- Uses landscape mode by default.
- Uses large text by default.
- Allows gradient fills.
- Allows layering of graphics.
- Allows recoloring of graphics on-the-fly (recolor picture, black and white view).
- Allows scaling of documents on the fly (scale to fit paper).
- Positions output on the page starting at the center of the paper and working outwards in all directions.

Slides print individually, but fail if all printed together

This problem usually occurs with a particular presentation file rather than all presentation files

Possible Causes

- Insufficient printer memory.
- Insufficient disk space.
- Timeout problem.

Solutions

- Check for old temporary files and delete them.
- Use an emulation printer driver.
- Clean up disk space by deleting old files and defragmenting:

- Change the Windows 95 spooler settings.
- Reduce printing resolution.
- Switch HPPCL/Raster mode.
- Print to a file and copy the file to the printer.

Certain slides don't print at all

This problem usually occurs with a particular presentation file rather than all presentation files.

Probable Causes

- Corrupted graphic/OLE object.
- Corrupted font.
- Incorrect printer driver.

Solutions

- Copy problem slides to new presentation to see if they print.
- Select or clear the Black And White check box in the Print dialog box.
- Use an emulation printer driver.
- If there something about that specific slide, such as the composition or content that is unique and causing the problem, try reformatting the slide.

Complex slides don't print

This problem usually occurs with any presentation files containing complex slides.

Probable Causes

- Insufficient printer memory.
- Insufficient disk space.
- Timeout problem.
- Bad emulation printer driver
- Corrupted graphic that is commonly used (such as a logo).
- Driver has a problem with landscape printing.
- Hardware problem (printer or cable).

Solutions

- Free disk space and remove old temporary and spool files.
- Print using a different printer driver.
- Print to a different printer.
- Switch printer cables.
- Print to a file and copy the file to the printer at the command prompt.
- Look for common graphics that are used in all affected slides.
- Copy the slide to another application, such as WordPad, and then try to print from there.

All slides don't print

This problem usually occurs with any presentation file, rather than just one.

Possible Causes

- Printer is not connected correctly.
- Inappropriate or corrupt printer driver.
- Insufficient disk space.
- Hardware problem (printer or cable).

Solutions

- Print from another application such as WordPad.
- Print a portrait mode slide from PowerPoint.

Other applications don't print, either

When you cannot print in any application:

Possible Causes

- Printer not connected correctly.
- Inappropriate or corrupt printer driver.
- Hardware problem (printer or cable).

Solutions

- Print a Windows 95 test page.
- Use the Print Troubleshooter in Windows 95 or the Print Troubleshooter included on the *Office Resource Kit* CD.

Whole slide image is too large or too small

This problem usually occurs with all presentation files.

Possible Causes

- Slide Setup is incorrect
- You used the Scale To Fit Paper command.
- Corrupt graphic/OLE Object (particularly an EPS file).

Solutions

- Check PowerPoint configuration.
- Check Slide contents (does removing any specific element clear this up?)
- Make sure that you have the current or best printer driver.

Whole slide image is in the wrong place

This problem usually occurs with all presentation files.

Possible Causes

- Printer has uneven unprintable regions (Deskjet 550, 560c).
- Printer driver reports center of page in the wrong place (Epson Stylus, Tektronics 200i)
- Printer driver is set up for wrong paper size.

- Solutions**
- Select or clear the Scale To Fit Paper check box in the Print dialog box.
 - Print to legal size paper and trim off unwanted excess.
 - Use an emulation printer driver.

Portions of the slide image are in the wrong place

This problem usually occurs with all presentation files.

- Possible Causes**
- Corrupted graphic or OLE Object (often an EPS graphic).
 - Corrupted font.
 - Insufficient printer memory.
 - Communications problem (for example, timeout or a bad cable).

- Solutions**
- Copy the slide to a blank presentation to check for a corrupted object on the slide master.
 - Replace fonts.
 - Print to a different printer.
 - Use a different driver.
 - Adjust the Windows 95 spooler settings.
 - Adjust the Windows 95 printer driver settings.

Graphic prints incorrectly or not at all

This problem usually occurs with all presentation files.

- Possible Causes**
- Corrupt graphic.
 - Printer Driver issue.
 - Display Driver issue.

- Solutions**
- Print the graphic from another application.
 - Use an updated driver or an emulation driver.
 - Use standard VGA display.
 - Adjust the Windows 95 spooler settings.
 - Adjust the Windows 95 printer driver settings.

Text prints incorrectly or not at all

This problem usually occurs with all presentation files.

Possible Causes

- Corrupt font.
- Incompatible font.
- Font substitution causing unexpected output.
- Font not available to printer.

Solutions

- Print the text from another application.
- Reinstall the font from the original location.
- Make sure that the printer driver provides the font to the printer.
- Make sure that the display driver doesn't prohibit the font from getting to the printer.

PowerPoint 7.0 OLE Troubleshooting

-- Here are some troubleshooting tips you can use when you encounter problems working with embedded objects or OLE servers.

Problem is PowerPoint-Specific

Make sure that the problem is application-specific by trying other OLE applications (such as other Office applications).

Check the registry Make sure that the application is properly registered in the Windows registry. PowerPoint will reregister when launched, if it finds a problem with the current registration.

Compel Office to reregister This is more effective than compelling PowerPoint to self-register alone. To do this, run Office Setup using the Run command on the Windows Start menu with the /y switch (type **setup /y**). This starts Setup maintenance mode. After selecting Reinstall, Setup rewrites all the registry keys for the last installation.

Compel PowerPoint to self-register Delete the key for the .ppt extension from Hkey_Classes_Root, and then launch PowerPoint.

Maximize resources Minimize other system variables such as memory usage and background tasks.

Problem Is Not PowerPoint-Specific

Determine whether the problem is application-specific by trying other OLE applications (such as other Office applications).

Check the registry Make sure that the application is properly registered in the Windows registry. Windows 95-compatible applications will reregister if run from outside of an application. Double click to reregister the application.

Try alternative approaches If the application has a toolbar button or a specific menu item associated with it, try those methods. If they don't work, try using the Object command on the Insert menu. If the object doesn't appear, it is not registered. If it appears, but does not work, it is improperly registered. If it appears and works using the Insert command, but does not work using the toolbar or menu, it is partially improperly registered and re-registration should fix it.

Check the path to the OLE server Check the path of the server in Hkey_Clases_Root. Trace the path to make sure that the OLE server executable is found.

Delete the server key Delete the key for the server from Hkey_Clases_Root. If necessary, remove and reinstall the server application itself. This should refresh all the files associated with the server. You must uninstall all of the applications that are registered as clients while uninstalling the OLE server application. Then reinstall them all. Otherwise, the OLE server application will not be completely uninstalled.

Problem Affects All OLE Applications

If none of your OLE functionality works from any OLE application, the problem may be with Windows. Call Windows product support or check your Windows technical documentation.

Word-to-PowerPoint Interoperability

Here are some tips that may be useful when you encounter problems inserting OLE objects from Word onto a PowerPoint slide.

Inserting a Word table doesn't work

- Make sure that Word is installed and properly registered.
- Make sure that Word objects (not tables) can be inserted.
- Make sure that registration is complete.

Inserting Word objects will not work

Try inserting a Word object into another application. If that doesn't work, Word is not properly registered.

Word objects in PowerPoint cannot be edited

- If the object is linked, make sure that the linked object is available and the link is correct. Use the Links command on the Edit menu.
- If the object is embedded, make sure that Word is properly registered and available. (The Document Object command appears on the Edit menu if a Word object is embedded.)

PowerPoint-to-Word Interoperability

Here are some tips that may be useful when you encounter problems inserting OLE objects from PowerPoint into a Word document.

Write-Up doesn't work

- Make sure that Word 7.0 is installed and registered.
- Make sure that you can put PowerPoint objects into Word without Write Up. If you cannot, reregister PowerPoint and Word.

Report-It doesn't work

- Make sure that Word 7.0 is installed and registered. If it is not installed, your presentation is exported to WordPad instead.
- Make sure that you can put PowerPoint objects into Word without Report-It. If you cannot, reregister PowerPoint and Word.

Copying and pasting a slide doesn't work

Make sure that PowerPoint is correctly registered.

Inserting an object in PowerPoint doesn't work

Make sure that PowerPoint is correctly registered.

PowerPoint objects in Word cannot be edited

- If the object is linked (check to see if the Links command on the Edit menu is active), make sure that the linked object is available and the link is correct.
- If the object is embedded (check to see if the Object command on the Edit menu is active) make sure that PowerPoint is properly registered and available.

Microsoft Excel-to-PowerPoint Interoperability

Here are some tips that may be useful when you encounter problems inserting OLE objects from Microsoft Excel onto a PowerPoint slide.

Inserting a Microsoft Excel worksheet doesn't work

- Make sure that Microsoft Excel is installed and properly registered.
- Make sure that Microsoft Excel objects (other than worksheets from the toolbar) can be inserted.

Inserting Microsoft Excel objects will not work

- Make sure that Microsoft Excel is installed and properly registered.
- Make sure that Microsoft Excel worksheets can be inserted using the Insert Microsoft Excel Worksheet button on the Standard toolbar. Microsoft Excel objects in PowerPoint cannot be edited.
- If the object is linked (check to see if the Links command on the Edit menu is active), make sure that the linked object is available and the link is correct.
- If the object is embedded (check to see if the Object command on the Edit menu is active), make sure that Microsoft Excel is properly registered and available.

PowerPoint-to-Microsoft Excel Interoperability

Here are some tips that may be useful when you encounter problems inserting OLE objects from PowerPoint into a Microsoft Excel workbook.

Copying and pasting a slide doesn't work

Make sure that PowerPoint is correctly registered.

Inserting a PowerPoint object doesn't work

Make sure that PowerPoint is correctly registered.

PowerPoint objects in Microsoft Excel cannot be edited

- If the object is linked (check to see if the Links command on the Edit menu is active), make sure that the linked object is available and the link is correct.
- If the object is embedded (check to see if the Object command on the Edit menu is active), make sure that PowerPoint is properly registered and available.

Supporting Microsoft Schedule+

This section contains information about possible Schedule+ user questions.

For information about the support implications of using Schedule+ 7.0 with Schedule+ 1.0, and with different mail servers and gateways, see Chapter 13, "Running Multiple Versions of Microsoft Office."

Additional support issues are discussed in the Schedule+ readme file, Sreadme.txt, located in the Microsoft Office Schedule+ folder.

Using Schedule+ with a Microsoft Mail 3.x/PROFS Gateway

PROFS® users connected to Schedule+ via a Microsoft Mail 3.x/PROFS gateway receive formatted meeting requests from Schedule+ 1.0 users. However, because of enhancements made to the Schedule+ 7.0 meeting request form, Schedule+ 7.0 meeting requests are transmitted to PROFS users as a plain text email message.

An update to the gateway software that fixes this limitation may be available at a later date through the Microsoft Product Support Quick-Fix Engineering (QFE) program.

Schedule+ 7.0 and the Timex Data Link Watch

Schedule+ 7.0 supports downloading schedule data to the Timex Data Link watch on Windows 95 only. The current release of Schedule+ 7.0 running on Windows NT does not support the Timex Data Link watch downloader.

Timex Data Link watch users can download text with international characters (for example, ê or Ç) to the watch. Due to physical limitations of the watch display, however, accented characters will download as their non-accented equivalent (for example, ê will download as e and Ç will download as C).

Calibrating the Timex Data Link Watch Manually

If a user is unable to download data consistently, or does not hear the watch's confirming beep when lines are displayed on the workstation screen during the download process, the display may need to be calibrated manually.

- ▶ **To manually calibrate the workstation display for Timex Data Link downloading**
 1. On the File Menu, click Export, and then click Timex Data Link Watch.
 2. Answer the Export Wizard questions and click Finish.
 3. In the Export To Watch dialog box, hold down the CTRL key and click Cancel. The Manual Calibration dialog box appears.
 4. In the manual calibration dialog box, click OK. Adjust the spacing between the lines on the screen with the UP ARROW and DOWN ARROW keys. When the watch starts beeping at regular intervals, press ENTER to begin the data download process.

Supporting Microsoft Word

This section describes common problems you or your users may encounter with Word 7.0.

General Troubleshooting Tips

The cause of many Word issues can be isolated and/or corrected by performing a few standard procedures described below.

Isolate the Problem When possible, try to isolate whether the problem exists in all cases or just in a specific case. For example, does the problem happen with all Word documents, or with just one document? Do you receive error messages when you try to convert any type of word processing document, or only when you try to convert a specific type of document? Are all inserted graphics printing incorrectly, or just one graphic? Answering the "all or one" question will help you determine whether it's a system or Word problem, or if the single document or portion of a document is at fault.

Start Word without Loading templates or Add-ins You can start Word without loading any global templates, add-ins, or settings. On the Windows Start menu, click Run, type `winword /a` and then click OK.

If the problem disappears when you start Word this way, the problem may be in your Normal.dot template, another global template, or one of the WLLs loading from the Startup folder.

Rename Normal.dot There are times when the Normal.dot template, which is the global template that comes with Word, may become corrupt, and adversely affect Word. Try quitting Word and then renaming this file to Normal.old. Restart Word as you normally would. Word will not locate the Normal.dot file and will revert to the original settings stored in the application program. If the problem does not reappear, it is likely that a setting, macro, or some problem in the Normal.dot template is causing the problem.

Tip If the problem is corruption in Normal.dot, you can copy macros, styles, and AutoText entries to a new Normal.dot by using the Organizer. On the File menu, click Templates, and then click Organizer.

Use Standard VGA Video Display Incompatible or corrupt third-party video drivers can create problems in Word. Display problems are an obvious symptom of this, but there are other problems that can be caused by a corrupt or incompatible video driver, such as printing problems, system stops, or even out-of-memory error messages when trying to start Word. Switching the video driver to the standard VGA video driver is a quick way to determine if it's a video driver problem.

If switching to another video driver works, it may be necessary either to reinstall the third-party video driver or to update the incompatible driver to the latest version. Switching to standard VGA most often won't be an acceptable fix, but it can provide a temporary workaround until you can reinstall the original video driver or obtain an update.

Use the Latest Video and Printer Drivers Word is very printer and display-centric. Word polls both the printer and video drivers for information more extensively than most Windows applications. As a result, other applications may not exhibit a particular problem, while Word does. Third-party video and printer drivers, as well as some of the printer drivers distributed by Microsoft, are updated fairly frequently. Updating your drivers to the latest versions will often correct display, printing, and other system-related problems. For information about obtaining updated video and printer drivers from Microsoft, see Appendix F, "Resources."

Try using WordPad When a problem is encountered in Word, it is wise to see if the problem is a more global Windows problem, so you don't waste all your time focusing on it as a Word-specific problem. WordPad is an excellent application in which to test problems you encounter in Word, whether they are printing, screen display, or other issues. While other applications such as Microsoft Excel may also be good test applications, WordPad is an excellent choice because it is more similar to Word than most Windows applications. Microsoft Excel, for example, isn't as concerned about printer driver information as is either Word or WordPad. So, if you can, use WordPad as a test application.

Specific Troubleshooting Areas

Word cannot open a document

Follow the suggested solutions in the order listed until the problem is solved.

Possible Causes

- The document is not on a properly formatted disk.
- The document is not in a format that Word supports.
- Not enough free disk space is available.
- The document has been corrupted.

Solutions

- If working with a document on a floppy disk that is not properly formatted for Windows, go back to the source of the document and copy it to a properly formatted disk.
- If the document is not in a format that Word supports, see if Word or the *Office Resource Kit* CD-ROM includes a converter and install it. For more information, see Chapter 20, "Switching to Microsoft Word." If no converter is available for the document's format, go back to the originating application and save the document in a format that Word supports.
- If not enough free disk space is available, free some. Try emptying the recycle bin, moving old, unused files to floppy disks, or deleting old, unused programs.
- If you still cannot open the document, check to see if the document is corrupt. Try opening it in the originating application if other than Word 7.0. If the originating application can open the document, resave it and try opening it in Word 7.0 again. If the originating application is Word 6.0 or Word 7.0, see "Determine if a document is damaged" later in this section.

Word cannot save a document

Follow the suggested solutions in the order listed until the problem is solved.

Possible Causes

- Not enough free disk space.
- The document is really a template.

Solutions

- If not enough free disk space is available, free some. Try emptying the recycle bin, moving old, unused files to floppy disks, or deleting old, unused programs.
- If the document is really a template, select all the contents of the document and paste it into a new document, and then save the new document.

When importing, Word fails to convert a graphic correctly

Follow the suggested solutions in the order listed until the problem is solved.

Possible Cause

The graphics filter Word needs is not installed.

Solutions

- If the graphics filter Word needs is not installed, install it. For more information, see Chapter 20, "Switching to Microsoft Word." If no filter is available for the graphic format, go back to the originating application and save the graphic in a format that Word supports.
- Try opening the graphic in the originating application. If the originating application cannot open the graphic, the graphic file may be damaged.
- Instead of opening the graphic with the Open command on the File menu, try inserting the graphic with the Picture command on the Insert menu.
- Try copying the graphic in the originating application and pasting it into Word.

Determine whether a document is damaged

If you experience any of the following symptoms when working with a document, it is possible the document is damaged. Follow the Solutions in the order listed until the problem is solved.

Tip When troubleshooting what appears to be document corruption, it is important to first rule out the possibility that it is not document corruption at all, but rather some unusual formatting or document setting. Save the document in RTF format or copy and paste all but the last paragraph mark into a new document.

Symptoms include:

- Incorrect document layout and formatting, unreadable characters on the screen, error messages during processing, Word stops running when you load or view the document.
- You are unable to save the document.

Possible Causes

- Word add-ins or other programs are interfering with Word.
- The template attached to the document is corrupted.
- The active printer driver is interfering with Word.
- The current video display is interfering with Word.

Solutions

- Check the paragraph, character, section and style formatting.
- Disable add-ins
- Disable terminate-and-stay-resident (TSR) programs and third-party software, particularly font driver packages.
- Attach the document to a different template.
- Switch to a different printer driver.
- Change the video display to standard VGA.
- Try working with the document on another computer.

If none of these Solutions corrects the problem, it's likely that the document is damaged. See "Recover a damaged document in Word" in the following section.

Recover a damaged document in Word

Follow the suggested solutions in the order listed until the problem is solved.

Possible Causes

- Word was interrupted while saving the document.
- The document became corrupted while stored on disk.

Solutions

- If you can open the damaged document, save it in RTF format, close the document, then open the RTF document, and save that in Word format.
- If you can open the damaged document, highlight a small portion of the document and copy it to a new document based on the Normal template. If this portion of the document responds normally, switch back to the original document and copy another portion to the new file. Repeat this process until an error occurs. At that point, you have isolated the problem to a specific part of the document. Recreate that part of the document in the new file.
- If you cannot open the damaged document in Word, try opening it in WordPad. Start WordPad, click the Open button, and in the Files Of Type box, make sure that Word for Windows 6.0 is selected. (Word 6.0 and Word 7.0 have the same file format.) If you can open the document in WordPad, save it in RTF format and then try opening that in Word 7.0.
- If you cannot open the damaged document in WordPad, try opening it in Notepad. Start Notepad, then click Open on the File menu. In the Files Of Type box, make sure that All Files is selected. The document will open with gibberish characters at the top of the document, followed by your unformatted text, followed by more gibberish characters. Copy the unformatted text out of Notepad and paste it into a new Word document based on the Normal template.
- Write a macro that opens the document, turns off screen updating and change other display options that might affect the document, as follows.

► **To create a macro to open a possibly corrupted document**

1. On the Standard toolbar, click New.
2. On the Tools menu, click Macro.
3. In the Macro Name box, type **screen**
4. Click Create.
5. In the macro editing window, click between the Sub Main line and the End Sub line.
6. Type the following:


```
ScreenUpdating 0
FileOpen.Name = c:\docs\mydoc.doc
ViewNormal
ToolsOptionsView.DraftFont = 1, .PicturePlaceHolders = 1
ToolsOptionsGeneral.Pagination = 0
ScreenUpdating
```
7. On the File menu, click Close, and then click Yes when prompted to keep changes to the macro Global.Screen.
8. On the Tools menu, click Macro.
9. Click Screen, and then click Run.

10. If the document opens, click Save As on the File menu.
11. In the Save File As Type box, select Rich Text Format.
12. In the File Name box, give the file a new name, and click Save.

Word 7.0 cannot complete a macro that Word 6.0 or earlier could complete

If your workgroup uses a custom macro that calls a 16-bit Windows application programming interface (API) or 16-bit Windows dynamic-link library (DLL), you will have to rewrite the macro for Word 7.0, which is a 32-bit application.

Possible Cause A 16-bit WordBasic macro calls a 16-bit Windows API or 16-bit Windows DLL.

Solution Rewrite your 16-bit WordBasic code to make Win32[®] API calls when the WordBasic code is executed from 32-bit Word. For more information, see "Porting Your 16-bit Office-Based Solutions to 32-bit Office" later in this chapter.

≡

Word prints a document, but incorrectly

--

Follow the suggested solutions in the order listed until the problem is solved.

Possible Causes

- Various settings in Word don't match your printer's options.
- Out-of-date or corrupt printer or video driver.
- Problems with fonts.

Solutions

- If the printed output is missing graphics or underlines, make sure that Word is not set to print draft output. On the Tools menu, click Options and then click the Print tab. Clear the Draft Output check box. If a graphic is still missing from the printed output, see "When importing, Word fails to convert a graphic correctly" earlier in this section.
- If the printed output is missing portions of text or individual letters, make sure that the text is not formatted as hidden text or as the color white.
- Create a new document in Word with just one line of text. Try printing this document. If it prints correctly, return to your original document and carefully check its formatting. If it still doesn't print properly, the document may be damaged. See "Determine whether a document is damaged" earlier in this section.
- Start WordPad and try to print the document from WordPad.
- Change the video settings to standard VGA.
- Turn off background printing in Word. On the Tools menu, click Options, and then click the Print tab. Clear the Background Printing check box.

- Use a printer font. On the Edit menu, click Select All, and then choose a font from the Font box on the Formatting toolbar that has a printer symbol beside it. If this corrects the problem, you may have a problem with the fonts you were originally using.
- Rename Normal.dot and then restart Word.
- Verify that you have the correct printer driver installed and it is set as the default.
- Disable TSRs and third-party software, particularly font driver packages.
- If you're using a font cartridge, turn your printer off, remove the cartridge, and turn your printer back on.
- Double-click Printers in the Windows Control Panel, and then double-click the printer you are using. In the Properties dialog box, click the Fonts tab, and then click Print TrueType As Graphics. For this option, you need about 300K of free memory per page. Some printers such as dot matrix, DeskJet, and BubbleJet do not support this option.
- The problem may be hardware related. Try removing any switch boxes, using a new printer cable, and, if you are on a network, connecting the printer locally. If this doesn't fix the problem, you may have a mechanical problem with the printer, the port connection on your printer, or the port connection on your computer.

Word cannot print a document

Follow the suggested solutions in the order listed until the problem is solved.

Possible Causes

- Word does not have the correct paper size or paper source settings specified for the printer you are using
- The printer does not have enough memory.

Solutions

- If an error message prints or appears on the computer screen, follow the instructions it contains
- Check the paper size and paper source settings in Word. On the File menu, click Page Setup. Check the settings on the Paper Size and Paper Source tabs.
- Reset the printer to clear the printer buffer.
- Restart the print queue by selecting Printers in the Windows Control Panel, double-clicking the printer you are using, and then clicking Resume Printing on the Document menu.
- Clear the print queue by selecting Printers in the Windows Control Panel, double-clicking the printer you are using, and clicking Cancel Printing on the Document menu

Page faults in Word

This section describes page faults in Word 7.0 running under Windows 95, which are comparable to General Protection (GP) faults in Word 6.0 running under Windows 3.x. Follow the Solutions in the order listed until the problem is solved.

Possible Causes

- Something unexpected has happened within the Windows environment, typically an improper memory address. For example, an application or a Windows component might read or write to a memory location that has not been allocated to it (memory that it does not own), potentially overwriting and corrupting other program code in that area of memory.
- Parameters that are not valid have been passed between applications and the Windows environment. Invalid parameters can cause invalid instructions to be executed, resulting in page faults. This is usually the result of the application's internal program code incorrectly passing specific data that could not be correctly interpreted by Windows or a Windows-based application.

Solutions

- Change the video display to standard VGA.
- Switch your printer driver to another compatible printer driver, or temporarily switch it to the Generic\Text Only printer driver.
- Temporarily remove any items located in the Windows Startup folder.
- Start Word without loading any global templates, add-ins, or settings. On the Windows Start menu, click Run, type **winword /a** and then click OK.
- Temporarily remove all templates and add-ins from the Word Startup folder.
- Quit Word, then delete the following key in the registry:
 \Hkey_Current_User\Software\Microsoft\Word\7.0\Data and then restart Word. For information about using the Registry Editor, see Appendix B, "Registry Keys and Values."
- Use the Find command on the Windows Start menu to locate any AutoSave files (look for "*.asd"). If you find any files with the .asd extension, rename the extension. This prevents Word from loading the AutoSave file when it starts.
- Use the Find command on the Windows Start menu to locate any AutoCorrect files (look for "*.acl"). If you find any files with the .acl extension, rename the extension. This prevents Word from loading the AutoCorrect file when it starts.
- Try removing Word through the Add/Remove programs icon in the Windows Control Panel, and then reinstalling Word.

Registry Settings and .Ini Files

Some Office for Windows 95 applications continue to use .ini files, even though they operate based on settings in the Windows registry. For backward compatibility, Office applications allow the use of the /ini command line option. Except for Word, Setup does not migrate .ini settings from previous versions of the applications when Office for Windows 95 is installed. Leftover .ini files have no effect on Office for Windows 95 applications.

When Setup installs Word 7.0 onto a system with a previous version of Word, everything in Winword6.opt and Winword6.ini is registered in the user's Hkey_Current_User registry keys. All shared components currently registered in the user's Win.ini (text converters, filters, proofing tools, and so forth) are registered in the Hkey_Local_Machine registry keys.

The following table lists Office applications, whether they use .ini files or the registry only, and information about how the applications use existing .ini files.

Application	Registry or .ini Files	Use of existing .ini files
Office	Registry only.	When upgrading previous versions of Office, buttons included in Msoffice.ini are migrated to the Old Office toolbar.
Microsoft Access	Registry only.	None.
Microsoft Excel	Registry only.	None.
Microsoft PowerPoint	Registry only, except where compatibility with PowerPoint 4.0 and other 16-bit applications requires use of .ini files. For example, PowerPoint 7.0 to PowerPoint 4.0 translator entries are stored in an .ini file.	None.
Microsoft Schedule+	Registry only.	Schedule+ 7.0 reads the Schedule+ 1.0 .ini file the first time it runs to find existing .cal file, migrates the .cal file to an .scd file, and sets registry entry for LocalPath.
Microsoft Word	Registry and .ini files.	Msfntmap.ini and Mstxtcnv.ini are not made into registry entries.

Long Filenames

When users transfer a file from an Office for Windows 95 application to a previous version and then back to the original application, the long filenames supported by Windows 95 may or may not be preserved. Each application handles preservation of long filenames slightly differently.

Microsoft Access Microsoft Access does not preserve long filenames, as users cannot open Microsoft Access 7.0 database files in a previous version of Microsoft Access and then move them back to version 7.0.

Microsoft Excel When opening a Microsoft Excel 7.0 workbook in Microsoft Excel 5.0, the older version of Microsoft Excel cannot update links or find link sources. However, when the workbook is opened again in Microsoft Excel 7.0, the long filename is preserved, and links are preserved and can be updated normally.

Microsoft PowerPoint When opening a PowerPoint 7.0 file using the Macintosh installable filters, the Windows 95 long filenames are preserved.

When using the Save As command in PowerPoint 7.0 to save a file in PowerPoint 4.0 format, a dialog box appears, prompting the user for a new filename. The user must consider the destination platform of the file when naming it. For example, if the file is destined for PowerPoint 4.0 running on Windows 95 or the Macintosh, long filenames are supported. If the file will be used in PowerPoint 4.0 for Windows 3.x, a long filename will be truncated, so the user might want to create an 8.3 filename, to avoid confusion.

Microsoft Schedule+ Schedule+ does not preserve long filenames, as users cannot open a Schedule+ 7.0 schedule file in a previous version of Schedule+.

Microsoft Word In Word, the long filename of the document itself is not preserved when the file is modified in Windows 3.x, then reopened in Windows 95. However, document names in links are updated, and the links still work.

Porting Your 16-Bit Office-Based Solutions to 32-Bit Office

The introduction of 32-bit Microsoft Windows 95 (and Windows NT) brings 32-bit Office applications into common use and results in users switching their Microsoft Office files, macros, and solutions to the 32-bit versions. This section assists you in porting solution code—that is, code written in the Microsoft Excel macro language (XLM), WordBasic, Visual Basic for applications, or Access Basic—to 32-bit versions of Office running on 32-bit operating systems.

Changes to your existing code are required if your 16-bit Office solution (including Microsoft Access or Microsoft Project) calls a 16-bit Windows application programming interface (API) or 16-bit Windows dynamic-link library (DLL), and you are porting that code to a 32-bit Office application (again, including Microsoft Access 95 or Microsoft Project 95).

Porting your solution code is necessary because 16-bit API calls and 16-bit DLL calls (referred to in this section simply as API calls) do not execute correctly when the solution code containing those calls is run in a 32-bit Office application. This section applies to solution code that uses APIs in the following products: Microsoft Access, Microsoft Excel, Microsoft Project, and Word for Windows.

Note 16-bit solutions ported to 32-bit Windows are not affected. For example, existing 16-bit Office solutions, 16-bit Visual Basic, and 16-bit FoxPro applications, even if they call 16-bit APIs, will run just fine on Windows 95 or Windows NT. It is only when users want to run solutions code that includes 16-bit API calls on a 32-bit application that porting is required.

Solution providers and corporate developers face an important task in ensuring that their Microsoft Office-based solutions run successfully under Windows 95 and Windows NT.

The rule they must follow is this: neither a 32-bit compiled application nor solution code called from a 32-bit Office application can make direct 16-bit API or DLL calls. In addition, neither a 16-bit compiled application nor solution code called from a 16-bit application can make direct 32-bit API or DLL calls. This inability to make calls back and forth between 16-bit and 32-bit layers occurs in both the Windows 95 and Windows NT environments because of their advanced flat-memory-model management systems, as well as the way in which they load DLLs.

To prepare for Office 95, you must change your solution code to make Win32 API calls when the solution code is executed from 32-bit Office applications. If this is not possible (for example, you don't have access to the source code of the DLL), you must change the solution code to *thunk* through an intermediate DLL to make the 16-bit API call. Updating solution code to support Win32 API calls is a relatively simple mechanical process. A more significant task is to write code that is operating-system-independent (that is, so the solution code will run on both 16-bit and 32-bit Office applications). This section discusses both of these tasks, as well as other 16-to-32-bit API issues you may need to handle.

Note Although you must update API calls when porting solution code to 32-bit operating systems, you do not need to change code that uses OLE Automation or dynamic data exchange (DDE). All OLE and DDE code will continue to work regardless of whether the applications are 16-bit or 32-bit. OLE and DDE insulate automation calls, so all combinations of containers (clients) and servers (16/16, 16/32, 32/16, and 32/32) work under Windows 95 and Windows NT.

How This Section Is Organized

What you need to know depends on your situation; therefore, this section is organized in terms of complexity, from the easier issues to the more complex ones.

- "Which API Should Your Solution Code Call?" is a quick overview of which API you should be using, according to your application needs.
- "Calling the Win32 API" describes what an API is and discusses the issues involved in converting existing 16-bit API calls to Win32 API calls, finding Declaration statements, and testing the Declare statements.
- "Writing a Single Code Base for 16-bit and 32-bit Office Applications" supplies code samples for writing solution code that will run on both a 16-bit and 32-bit Office application.
- "Determining Whether a 32-bit Application Is Running" describes how to determine whether your Office application is 16-bit or 32-bit and how to select the appropriate 16-bit or 32-bit API call.
- "Recompiling DLLs" tells you what you need to do to make the DLL and solution code work on Windows 95 and Windows NT if your solution code calls a custom DLL.
- "Thinking" tells you how, if you cannot recompile your DLLs, you can add an intermediate DLL.
- "Advanced Programming Topics" explains: translating C-API declarations to Visual Basic or Visual Basic for applications.

Which API Should Your Solution Code Call?

When you write solution code for your own use, you write it for the version of the Office application you have and for your own operating system. Distributing this solution to others means that you have to make it also work on their computers, which may use different versions of Windows and Office applications than you used when you wrote it. While the operating system isn't an issue, whether the Office application is 16-bit or 32-bit is important. The following table shows that the application, and not the operating system, determines which API you use in porting your solution code.

Microsoft product	Windows 3.x	Win32s	Windows NT	Windows 95
16-bit applications	16-bit API	16-bit API	16-bit API	16-bit API
32-bit applications	N/A	32-bit API	32-bit API	32-bit API

Note Microsoft Office (including 32-bit Microsoft Access and 32-bit Microsoft Project) products do not run on Win32s, but because Microsoft FoxPro does, the Win32s column was added to show that FoxPro programmers should use the same rules for choosing the API. Also, Win32s, Windows NT, and Windows 95 do not have identical sets of API calls. For more information, see the Win32 SDK documentation in the Development Library (in particular, see the Compatibility Tables in the *Win32 Programmer's Reference*, Vol. 5).

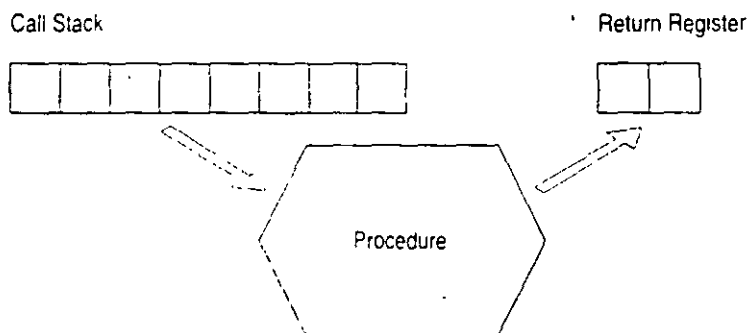
Calling the Win32 API

To write calls to the Win32 API, you must do the following:

- Understand what Windows API calls are.
- Understand the differences between 16-bit and 32-bit Windows APIs.
- Use Win32api.txt to find the correct **Declare** statement.
- Test an API **Declare** statement.

What Is an API Call?

An API call in C, Visual Basic, or other languages places a series of values (parameters) at a location in memory (the stack) and then requests the operating system or DLL to execute a function (the procedure call) using the values provided. The function reads the values (call stack) and executes its function code using those values or the data that the values point to. If a result is returned, it is placed at another location (return register) for the calling application to use. This is shown in the following illustration. To ensure accuracy, the number of bytes of data on the stack is verified before and after the procedure is called. The message "Bad DLL calling convention" appears when the wrong number of bytes are on the stack.



In practical terms, Windows API calls are how applications request services (screen control, printers, memory) from the operating system. There are approximately 300 API calls in Windows 3.0, over 700 API calls in Windows 3.1, and over 1,000 API calls in Windows 95. These API calls are packaged in executables and DLLs found in the Windows folder—User.exe, Gdi.exe, and one of the following KERNEL files: Kernal286.exe, Kernal386.exe, or Kernel32.dll

To call an API from your solution code, use these four steps:

1. Identify the file containing the API.
2. Determine the parameters required by the API.
3. Create a **Declare** statement for the API.
4. Call the function with valid parameters.

The following is a simple example for the **GetVersion** API call that obtains the version of Windows that is running. The **GetVersion** API call is located in KERNEL under 16-bit Windows and does not use any parameters (so the **Declare** statement has empty parentheses). The following **Declare** statement is written for 16-bit Windows for use by Visual Basic for applications:

```
Declare Function GetVersion Lib "KERNEL" () As Long
```

By comparison, here is the same function as it would be used by an Office 95 application running on 32-bit Windows:

```
Declare Function GetVersion Lib "KERNEL32" () As Long
```

Although the Windows API name stays the same, the location of the API has changed to KERNEL32. Because you are calling from a 32-bit application, you must make a 32-bit API call. The parameter data type, on the other hand, did not change (it remained a **Long**). In general, the function parameters will change more and require more attention than the parameters of the return value. Understanding the differences between 16-bit API calls and 32-bit API calls is essential to porting your solution code to Windows 95.

What Are the Differences Between a 16-Bit and a 32-Bit Windows API?

As shown in the previous example, most 32-bit Windows API calls have the same name or a very similar name to the 16-bit API calls. In fact, the documentation may show the same arguments, with the only apparent difference being the library name change from KERNEL to KERNEL32. However, the code must handle changes in addition to the name change:

- Case-sensitivity
- Unicode or ANSI options
- Change of parameter data type (shown in the previous example)

These bulleted items can require subtle changes in the **Declare** statements that are not always easy to identify.

Case sensitivity

The first issue in moving to 32-bit Windows API calls is case sensitivity in the name of the function. API calls under 16-bit Windows are not case sensitive and work if you enter the function name as `GetVersion`, `GetVersion`, or `getversion`. In other words, in 16-bit Windows the following statements are equivalent:

```
Declare Function GetVersion Lib "KERNEL" () As Long
Declare Function GetVersion Lib "KERNEL" () As Long
```

- — API calls under 32-bit Windows are case-sensitive for the function call and must be correctly entered in the **Declare** statement. In other words, the following statements are not equivalent in 32-bit Windows:

```
Declare Function GetVersion Lib "KERNEL32" () As Long
Declare Function GetVersion Lib "kernel32" () As Long
```

The easiest way to handle this change is to always use the **Alias** control word. The contents of an **Alias** string map to the actual API call name (which is case sensitive), but the function name used in code, which appears between "Function" and "Lib," is not case sensitive and will not change if you type it different ways in your code or use the same name for variables or procedures. Using the **Alias** control word, the `GetVersion` function (32-bit Windows) would be entered as:

```
Declare Function GetVersion Lib "kernel32" Alias "GetVersion" () As Long
```

Now the case of API names won't matter when writing code: as long as you spelled and typed the function name correctly in the **Alias** string and you spell the function name in code the same way as in the **Declare** statement, the function will be mapped by Visual Basic or Visual Basic for applications back to the correct **Declare** function automatically.

Note The **Alias** control word is the single most important thing you can use in preparing to switch to 32-bit operating systems because it means you will only have to change the contents of the **Declare** statement and not every instance of the function being called in your code.

Unicode or ANSI Options

Both Windows NT and Windows 95 have two API interfaces. One interface is based on the American National Standards Institute (ANSI) character set, where a single byte represents each character. The other interface was created for the Unicode character set, where two bytes represent each character. All 16-bit Windows operating systems and applications use the ANSI character set. All 32-bit versions of Windows added Unicode to allow foreign language characters to be represented. C programmers handle this by setting a flag in their **include** file (*.h). The flag causes hundreds of macros throughout the C **include** files to select the correct Unicode or ANSI functions.

All western language versions of Office products use ANSI for Visual Basic for applications code. Therefore, programmers using current versions of Visual Basic for applications or macro languages will always use the ANSI version of the API call. When using the Win32api.txt file, this choice is made for you. For more information about this file, see "Change of Parameter Data Type," next in this section.

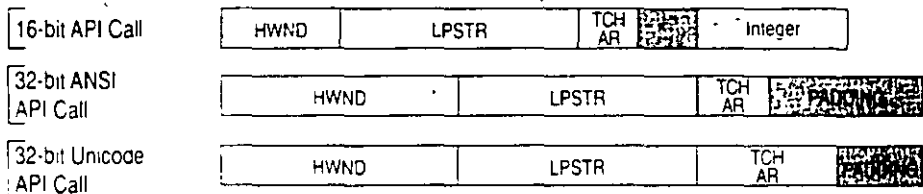
≠ ≠
— —

To distinguish the ANSI version from the Unicode version, the ANSI version adds an "A" to the end of the API name, and the Unicode version adds a "W". W is for wide, as in the width of the bytes provided for characters. The name of an API call includes the characters A and W at the end of the API name only if the API requires parameters with string (character) data types.

The Win32 SDK documentation in the Development Library does not record the permutations of the name of the API call. The documentation gives only the name of the root function and its library name. The actual name of the API in the library can be one of three possibilities:

- **MyAPICall**, which uses no character strings in the call.
- **MyAPICallA**, which uses ANSI character strings in the call.
- **MyAPICallW**, which uses Unicode character strings in the call.

To understand the differences, see the following diagram showing the amount of data the API expects to find on possible call stacks for an example function (the 16-bit version is padded because 16-bit Windows always pads the stack to 16 bits).



The three possible declarations for **MyAPICall** are shown next, formatted to make comparison easier. All of the statements use the **Alias** control word so that the function name used in code (**MyAPICall**) does not have to change even if the name of the function called is appended with an "A" or a "W":

```
'16 bits
Declare function MyAPICall Lib "MYDLL.DLL" Alias "MyAPICall" (
    ByVal hwndForm As Integer,
    ByVal lpstrCaption$,
    ByVal hAccKey As String,
    ByVal iMagicNumber As Integer
) As Integer
'32-bit ANSI
Declare function MyAPICall Lib "MYDLL32.DLL" Alias "MyAPICallA" (
    ByVal hwndForm As Long,
    ByVal lpstrCaption$,
    ByVal hAccKey As String,
    ByVal iMagicNumber As Long
) As Long
'32-bit UNICODE * For illustration only.
Declare function MyAPICall Lib "MYDLL32.DLL" Alias "MyAPICallW" (
    ByVal hwndForm As Long,
    ByVal lpstrCaption$,
    ByVal hAccKey As String,
    ByVal iMagicNumber As Long
) As Long
```

Any one of these declarations would add the function **MyAPICall** to your application; you can only have one **MyAPICall** function.

Note This code sample introduces the **ByVal** keyword, which enables you to pass Visual Basic parameters to a API function by value. *By Value* is the default for functions written in C, and is therefore the default for Windows API calls. The reason you must use **ByVal** is Visual Basic and Visual Basic for applications default to **ByRef** (*By Reference* which passes a pointer to the value rather than the value itself) which is not what API calls expect. **ByVal** can also be used to convert a Visual Basic string to a null-terminated C string. **ByVal** is included in the Declare statements in `Win32api.txt` so you will know when to use it, but for more information about **ByVal**, see the MSDN article Q110219 "How to call Windows API from VB," or published references such as "The Visual Basic Programmer's Guide to the Windows API" by Dan Appleman, published by Ziff-Davis Press.

Change of Parameter Data Type

The easiest way to learn what the new required parameter data types are for 32-bit API functions is to have somebody else give them to you. Included on the *Office Resource Kit* CD is a Visual Basic declaration file called `Win32api`. All you need to do is copy the appropriate API **Declare** statement from `Win32api.txt` into your source code.

Another source of information is the *Win32 Programmer's Reference* on the MSDN Development Library CD, which is discussed in "Advanced Programming Topics" later in this section. This reference may occasionally be required to resolve questions about the inclusion or exclusion of **ByVal** in the declaration or the need to put parentheses around the actual value passed.

If you use the *Win32 Programmer's Reference*, however, you must be careful to properly convert C to Visual Basic data types. For example, don't mistake a C **int** for a Visual Basic for applications **Integer**. Many Windows data types and Visual Basic **Integer** data types are no longer the same size, as shown in the following table. It is critical to remember that the sizes of many API parameters have changed, and you must not assume they are the same.

Visual Basic data types	Size of variable	16-bit Windows data types	32-bit Windows data types
Integer	2 bytes	int , short , WORD , HWND , HANDLE , WCHAR	short , WCHAR
Long	4 bytes	long , LPSTR	int , long , HANDLE , HWND , LPSTR

Finally, whether you use `Win32api.txt` or the *Win32 Programmer's Reference*, judicious use of the **Alias** control word may assist you with changing parameter data types by allowing existing 16-bit *code* (italicized to point out that code does not include the **Declare** statement, which must change to point to a 32-bit API) that calls the API to be left unchanged. The **ByVal** control word and automatic type conversion in Visual Basic, Access Basic, WordBasic, and Visual Basic for applications change the size of parameters for you in many cases (**Integer** to **Long**, for example). Alternatively, type conversion will extend integers with a sign (+/-) that may lead to incorrect long parameters and cause overflows on conversion from **Long** to **Integer**. Again, the best solution is to check `Win32api.txt` or the *Win32 Programmer's Reference* to get the correct functions.

What Types of Errors Can Occur with an API Declare Statement?

After you create a **Declare** statement, it may not work. While there are many mistakes possible in a **Declare** statement, the following are the most common errors:

Error 453: Function is not defined in specified DLL Either you misspelled the function name or you have a problem with case in the function name. Functions are case-sensitive in Win32; they are not case-sensitive in 16-bit Windows.

Error 48: Error in loading DLL Usually, this error is caused by having the wrong size or arguments, but may also occur for some of the reasons described under Error 53.

Error 53: File Not Found Windows checks the loaded libraries for matches, and if the DLL is not loaded, it will attempt to load the DLL from disk. Many functions available in the 16-bit Windows on Windows (WOW) layer on a Windows NT system are not available directly from Windows NT. Calling the 16-bit Windows and Win32 `GetProfileString` function from a 16-bit and a 32-bit solution will give a confusing set of error messages. The 16-bit application call will find `KERNEL` and fail to find `KERNEL32`, while the 32-bit application will find `KERNEL32` and fail to find `KERNEL`. The general cause of this error is a mismatch of calls and environment. The solution is to write code that works in both 16-bit and 32-bit environments.

Writing a Single Code Base for 16-Bit and 32-Bit Office Applications

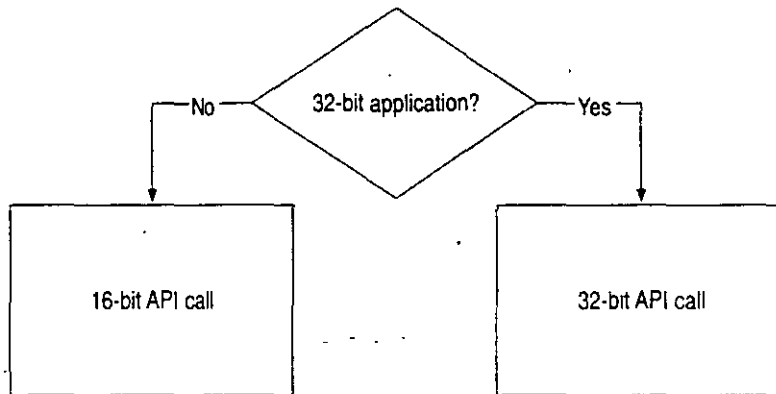
If your users are running both 16-bit and 32-bit versions of Microsoft Excel, should you put the 16-bit API call or the 32-bit API call in your solution? Microsoft Excel is not like Visual Basic version 4.0, which allows conditional compilation of code into executables, but instead runs solution code in workbook files which may be opened in either the 16-bit or the 32-bit version. The solution is that you put both the 16-bit API call and the 32-bit API call inside an **If...Then...Else** control structure.

With 32-bit applications using the same solution code as 16-bit applications, you do not know which API to call in the solution code. Your code must determine whether the application is a 16-bit one or a 32-bit one. How does the code answer the following questions:

- Microsoft Access: Is the host application 16-bit Microsoft Access 2.0 or earlier (make 16-bit API calls), or 32-bit Microsoft Access for Windows 95 (make 32-bit API calls)?
- Microsoft Excel: Is the host application 16-bit Microsoft Excel 5.0 or earlier (make 16-bit API calls) or 32-bit Microsoft Excel 5.0 for Windows NT or 32-bit Microsoft Excel for Windows 95 (make 32-bit API calls)?
- Microsoft Project: Is the host application 16-bit Microsoft Project 3.0 (make 16-bit API calls) or 32-bit Microsoft Project version 4.0 or later (make 32-bit API calls)?

- Word for Windows: Is the host application 16-bit Word 6.0 or earlier (make 16-bit API calls) or 32-bit Word 6.0 for Windows NT or Word for Windows 95 (make 32-bit API calls)?

If you make the wrong API call, an error will occur. The solution code must determine whether the application is a 16-bit application or a 32-bit application, so it can make the appropriate call.



The solution is to put every API call into a *wrapper*—a Visual Basic procedure or a Microsoft Excel 4.0 macro. This wrapper routine checks the “bitness” of the application and selects the appropriate API call. Place these wrappers in separate modules so that your code can be easily reused. Some API calls (for example, **GetPrinterDriveDirectory** and **GetWinMetaFileBits**) are not available in all 32-bit operating environments, which means that the structure of an API wrapper can become as complex as this:

```

Function MyAPICall$(ByVal Args)
    If Engine32() Then
        'Select is rarely needed
        Select Case OS32() 'Based on GetVersionEx API
            Case 0 'Win32s
                ....
            Case 1 'NT 3.1
                ....
            Case 2 'NT 3.5
                ....
            Case 3 'Windows 95
                ....
        End Select
    Else '16-bit
        ....
    End If
End Function
  
```

An API wrapper this complex is the exception and not the rule.

Compiled languages, such as FoxPro and Visual Basic, build 16-bit or 32-bit application executables. The executable targets either 16-bit API calls or 32-bit API calls. You can determine the appropriate API calls while building the application. You can select the calls either by having all the 16-bit declarations in one file and all the 32-bit declarations in another file and manually switching them in a project, or by using the **#IF... #ELSE...** directives and conditional compilation supported by Visual Basic 4.0. If you must support Visual Basic 3.0 and Visual Basic 4.0 applications concurrently, separate files may reduce code maintenance. If you support FoxPro, you will have no problem using 16-bit API calls from compiled 32-bit FoxPro solutions because the **RegFN** functions will automatically think from the 32-bit layer to the 16-bit layer if needed.

Compiled 32-bit languages may require some minor differences in API calls depending on the 32-bit operating system. For example, developers must program context menus differently for Windows 95 than for Windows NT.

Determining Whether a 32-Bit Application Is Running

- - In the previous section, writing application independent code was covered by adding code for both 16 and 32-bit scenarios. However, you still need to determine in source code whether the application is a 32-bit application or a 16-bit application without doing any API calls. You cannot do an API call because you do not know if a 16-bit API call or a 32-bit API call will work. The following code will determine if the application is a 32-bit application.

Note The **Application.OperatingSystem** property in Microsoft Excel and Microsoft Project does not return the version of Windows you have installed, but the layer of Windows that the application is running on, for example, the 16-bit subsystem in Windows NT.

Microsoft Excel 5.0, Microsoft Excel 95, Microsoft Project 4.0, Microsoft Project 95

```
Function Engine32%()
    If Instr(Application.OperatingSystem,"32") then Engine32%=True
End Function
```

Word 6.0, Word 95

```
Function Engine32
    If Val(GetSystemInfo$(23)) > 6.3 Or Len(GetSystemInfo$(23)) = 0 Then
        Engine32 = 1 Else Engine32 = 0
    End Function
```

Microsoft Access 1.1 or higher

```
Function Engine32% ()
    If SysCmd(7) > 2 Then Engine32% = True
End Function
```

An API wrapper this complex is the exception and not the rule.

Compiled languages, such as FoxPro and Visual Basic, build 16-bit or 32-bit application executables. The executable targets either 16-bit API calls or 32-bit API calls. You can determine the appropriate API calls while building the application. You can select the calls either by having all the 16-bit declarations in one file and all the 32-bit declarations in another file and manually switching them in a project, or by using the **#IF... #ELSE...** directives and conditional compilation supported by Visual Basic 4.0. If you must support Visual Basic 3.0 and Visual Basic 4.0 applications concurrently, separate files may reduce code maintenance. If you support FoxPro, you will have no problem using 16-bit API calls from compiled 32-bit FoxPro solutions because the **RegFN** functions will automatically thunk from the 32-bit layer to the 16-bit layer if needed.

Compiled 32-bit languages may require some minor differences in API calls depending on the 32-bit operating system. For example, developers must program context menus differently for Windows 95 than for Windows NT.

Determining Whether a 32-Bit Application Is Running

- In the previous section, writing application independent code was covered by adding code for both 16 and 32-bit scenarios. However, you still need to determine in source code whether the application is a 32-bit application or a 16-bit application without doing any API calls. You cannot do an API call because you do not know if a 16-bit API call or a 32-bit API call will work. The following code will determine if the application is a 32-bit application.

Note The **Application.OperatingSystem** property in Microsoft Excel and Microsoft Project does not return the version of Windows you have installed, but the layer of Windows that the application is running on, for example, the 16-bit subsystem in Windows NT.

Microsoft Excel 5.0, Microsoft Excel 95, Microsoft Project 4.0, Microsoft Project 95

```
Function Engine32%()
    If Instr(Application.OperatingSystem,"32") Then Engine32%=True
End Function
```

Word 6.0, Word 95

```
Function Engine32
    If Val(GetSystemInfo$(23)) > 6.3 Or Len(GetSystemInfo$(23)) = 0 Then
        Engine32 = - 1 Else Engine32 = 0
    End Function
```

Microsoft Access 1.1 or higher

```
Function Engine32% ()
    If SysCmd(7) > 2 Then Engine32% = True
End Function
```


Putting It All Together

For examples of code for particular applications, you should consult the Microsoft Developers Network (MSDN), and find the article entitled "Corporate Developer's Guide to Office 95 API Issues." For information about connecting to MSDN, see Appendix F, "Resources." The following simple example may help you understand some issues.

```

Declare Function GetTickCount32 Lib "KERNEL32" Alias "GetTickCount" ()
➔ As Long
Declare Function GetTickCount16 Lib "USER" Alias "GetTickCount" ()
➔ As Long

Function GetRightTickCount() As Long
If Engine32%() Then
    GetRightTickCount = GetTickCount32()
Else
    GetRightTickCount = GetTickCount16()
End If
End Function

```

The **GetTickCount** API has the same name for both 16-bit Windows and 32-bit Windows, so you must use an **Alias** control word to change the function name in at least one of the **Declare** statements. In the previous example, the names in both **Declare** statements were changed, to **GetTickCount32** and **GetTickCount16**. Next, depending on the application's bitness, **GetTickCount** is mapped to the correct API function name (**GetTickCount32** or **GetTickCount16**) and its associated API call. In this example, **GetTickCount** in your code will be mapped to **GetTickCount32** (in the **GetTickCount** function), which is mapped to **GetTickCount** in **KERNEL32**, when **Engine32%** is **True**.

Word Sample Declare-method Solution

Word has a different **Declare** format and syntax. The Word solution is more complex because you cannot place both the 16-bit and 32-bit **Declare** statements in the same macro. The solution is to create three macro libraries: **APICALL16** and **APICALL32**, that contain the **Declare** statements for each operating environment, and a 16-bit/32-bit interoperability macro, **APICALLS**.

First, create a macro library called **APICALL16**. This macro contains all the 16-bit API **Declare** statements.

```

' This is APICALL16 -- all 16-bit Declare statements are placed here.
Declare Function GetTickCount16 Lib "USER" Alias "GetTickCount"()
➔ As Long
Function GetTickCount
GetTickCount = GetTickCount16
End Function

```

Second, create a macro library called **APICALL32**. This macro contains all the 32-bit API **Declare** statements.

```
'This is APICALL32 -- all 32-bit Declare statements are placed here.
Declare Function GetTickCount32 Lib "KERNEL32"() Alias "GetTickCount"
  As Long
Function GetTickCount
  GetTickCount = GetTickCount32
End Function
```

Third, create a macro library called **APICALLS**. This macro contains **Engine32** and the procedures your solution code will call.

```
'This is APICALLS -- no Declare statements may be in this macro.
Function Engine32
  Engine32 = 0
  If Val(AppInfo$(2)) > 5 Then
    OS$ = GetSystemInfo$(23)
    If Val(OS$) > 6.3 Or Len(OS$) = 0 Then Engine32 = - 1
  End If
End Function
```

```
Function GetTickCount
  If Engine32 Then
    GetTickCount = APICall32.GetTickCount
  Else
    GetTickCount = APICall16.GetTickCount
  End If
End Function
'Other API function calls are placed here.
```

You can now call this function from your solution code. You must preface your calls with **APICALLS**. For example:

```
Sub MAIN
  MsgBox Str$(APICALLS.GetTickCount)
End Sub
```

> **To convert Word solutions to run on 16-bit and 32-bit products**

1. Create a new module called **APICALLS**.
2. Create the **Engine32** function in **APICALLS**.
3. Create a new module called **APICALL16**.
4. Locate all the 16-bit **Declare** statements in the solution and move them to **APICALL16**.
5. Create a new module called **APICALL32**.
6. Create the equivalent 32-bit **Declare** statements and put them to **APICALL32**.

7. Using the template above, create functions for each API in each of the three macro libraries.
8. Add **APICALLS** before all calls to the API in your solution code.
9. Test each function.

This process allows existing calls in other modules to be left untouched. After the developer defines and tests these macros, the developer can add them to `Normal.dot` and reuse the macros in other solutions so as to cut conversion time.

Recompiling DLLs

This section has so far focused on the issue of updating Windows API calls—but the issues for solution code that calls 16-bit DLLs that you have bought, developed, or simply used is exactly the same. The developer must change all 16-bit DLL **Declare** calls in solution code to 32-bit calls. This requires creating a 32-bit version of the DLL (at least) and possibly changing the **Declare** statement (in Microsoft Excel 4.0 macros, the **Register** function).

This also means a separate DLL must exist for both the 16-bit application and the 32-bit application. For file management, the name of the 32-bit DLL should include "32" at the end. The developer must recompile the DLL as a 32-bit `Ansi.dll`. The parameters passed to the DLL must use the **stdcall**-passing protocol to talk to 32-bit Visual Basic for applications, instead of the **PASCAL**-passing protocol used with 16-bit Windows. Place the calls for the 16-bit and 32-bit versions of the DLL in a wrapper similar to the API wrapper described previously.

For information about recompiling applications, see Chapter 1, "Porting 16-bit Code to 32-bit Windows," in *Programming Techniques* from the Visual C++ 2.1 documentation in the Development Library, or see your C compiler documentation.

Thinking

When you do not have the source code of a DLL, your solution is to use *thinking* to port your solution to Windows 95. Thinking enables direct 16-bit and 32-bit calls but requires much more work than simply changing the Windows API call. If you cannot change or recompile the 16-bit DLL, you must write a new 32-bit DLL wrapper to access the 16-bit DLL. The 32-bit application calls to this 32-bit wrapper DLL, which then calls the original 16-bit DLL.

Thinking allows parameters to be pushed correctly on the stack, enables a DLL of a different bitness to load in your process, and converts memory addresses from **offset** (32-bit) to **segment::offset** (16-bit). This means, however, there are some issues even if you do the thinking work. For example, pointers to pointers to memory locations require additional work in 16-bit and 32-bit scenarios.

There are different ways to think depending on your operating system. Windows 95 and Windows NT think differently. For an overview of thinking across the Windows platforms and pointers to more information about thinking, see "Diving into the Requirements for the Windows 95 Logo" in the Development Library.

Advanced Programming Topics

Most developers writing solution code know the C language, and the following information is provided to assist them in using their knowledge of C to create Declare statements for Visual Basic and Visual Basic for applications using the tools they already have.

Working from C Declarations

Apart from the API location changing (from KERNEL to KERNEL32), the main issue in moving from 16-bit API calls to 32-bit API calls is the change in the size of parameter data types. Some background information may help you understand what has changed and why. Windows 3.0 was designed for the Intel 80286 CPU, where the hardware handles data two bytes at a time or in 16-bit words. Windows 95 was designed for later CPUs, where the hardware can handle data four bytes at a time or in 32-bit words. The following list shows how Visual Basic represents an **Integer** versus how Windows represents an **int**:

- **Integer** and **int** are each two bytes in the 16-bit Windows operating system and in 16-bit Microsoft Excel, Visual Basic, Microsoft Access, Word for Windows, and Microsoft Project.
- **Integer** is two bytes in 32-bit Microsoft Excel, Visual Basic, Microsoft Access, Word for Windows, and Microsoft Project, the same as in the 16-bit versions of these products.
- **int** is four bytes in the 32-bit Windows operating systems, Windows 95, and Windows NT.

To illustrate how this change of size can change a call, recall the fictional **MyAPICall** API used earlier. The **MyAPICall** call needs the handle to the application's window (**HWND**), a string, a character, and an integer to be placed on the stack. In C, the function would be:

```
int MyAPICall (HWND hForm, LPSTR lpstrCaption, TCHAR tchAccKey,
int iMagicNumber)
```

Each parameter has two parts: the data type (**HWND**, **LPSTR**, **TCHAR**, **int**) and the field name (**hwndForm**, **lpstrCaption**, **tchAccKey**, **iMagicNumber**). Each data type requires a specific number of bytes to represent it. Each field name has some odd-looking characters as a prefix—these characters (known as Hungarian notation) indicate the data type, such as **int** or **lpstr**.

Windows has many data types that API calls use as parameters. The following table shows some of the more significant data types used by Windows 95 API calls. Many Windows data types use the C data type of **int**. When **int** changed from 16-bits to 32-bits, the related Windows data types also changed.

C data type	Windows 3.x; Windows for Workgroups 3.x (16-Bit)	Win32s, Windows NT, and Windows 95 (32-Bit)
unsigned int, UINT, int	2 bytes	4 bytes
short	2 bytes	2 bytes
long	4 bytes	4 bytes
char, CHAR	1 byte	1 byte
WORD	2 bytes	2 bytes
Handle (hWnd, hDC, hMenu)	2 bytes	4 bytes
LPSTR	4 bytes	4 bytes
WCHAR	2 bytes	2 bytes
TCHAR (ANSI or Unicode)	1 byte	1 or 2 bytes
POINT	4 bytes	8 bytes

Thus, converting our **MyAPICall** API call from C, the declarations for **MyAPICall** using Visual Basic for applications, Access Basic, or WordBasic would be as follows (formatted to make comparison easier):

```
'16 bits
Declare Function MyAPICall Lib "MYDLL.DLL" Alias "MyAPICall" (
    ByVal hwndForm As Integer,
    ByVal lpstrCaption As String,
    ByVal hAccKey As String,
    ByVal iMagicNumber As Integer
) As Integer

'32 bits
Declare Function MyAPICall Lib "MYDLL32.DLL" Alias "MyAPICall" (
    ByVal hwndForm As Long,
    ByVal lpstrCaption As String,
    ByVal hAccKey As String,
    ByVal iMagicNumber As Long
) As Long
```

A final tool you may find useful is the following table that maps C language declaration data types to their Visual Basic equivalents:

C language declaration	Visual Basic equivalent	Call with
Boolean	ByVal B As Boolean	Any Integer or Variant variable
Pointer to a string (LPSTR)	By Val S As String	Any String or Variant variable
Pointer to an integer (LPINT)	I As Integer	Any Integer or Variant variable
Pointer to a long integer (LPDWORD)	L As Long	Any Long or Variant variable
Pointer to a structure (for example, LPRECT)	S As Rect	Any variable of that user-defined type
Integer (INT, UINT, WORD, BOOL)	ByVal I As Integer	Any Integer or Variant variable
Handle (32 bit, HWND)	ByVal H As Long	Any Long or Variant variable
Long (DWORD, LONG)	ByVal L As Long	Any Long or Variant variable
Pointer to an array of integers	I as Integer	The first element of the array, such as I(0)
Pointer to a void (void*)	V As Any	Any variable (use ByVal when passing a string)
Void (function return value)	Sub Procedure	n/a
NULL	As Any	ByVal 0&
Char (TCHAR)	ByVal Ch As String	Any String or Variant variable

Using the Win32 Programmer's Reference

The two primary sources for Win32 API information are the *Win32 Programmer's Reference* and a list of Microsoft-supplied Win32 Declare statements for Visual Basic, such as Win32api.txt. The Development Library contains a listing with explanations of the entire Win32 API set in the *Win32 Programmer's Reference*. For more information about the Development Library, see "Microsoft Developers Network," in Appendix F.

Frequently Asked Questions

Automated answers to common questions, solutions to technical problems, and tips are available at no cost, 24 hours a day, 7 days a week, by telephoning Microsoft FastTips at (800) 936-4100. These tips change frequently to respond to the most common issues.