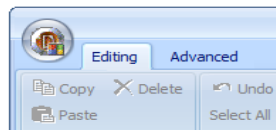


El Menú de aplicación

Para completar nuestra aplicación, para la cual he enlazado diversas acciones personalizadas sin tener que escribir código, debería añadir dos elementos relevantes para el interfaz de usuario Ribbon. Ambos son añadidos usando comandos al editor del componente “Ribbon” (el menú de acceso rápido que aparece en diseño cuando el componente está seleccionado) y puede ser añadido solo si el `ActionManager` está establecido.

El primer elemento en el menú de aplicación, el control redondo en la esquina superior izquierda de la “Ribbon” reemplaza el menú de aplicación tradicional de Windows. Aquí está el componente en ejecución en la demo:



Es característico de este elemento un menú desplegable inicialmente vacío. La idea es usarlo para las operaciones relacionadas con ficheros, y se pueden añadir diversas acciones estándar como Abrir o Guardar, es posible arrastrar acciones a esta barra de herramientas, pero no es fácil, ya que suele cerrarse. He encontrado más fácil seleccionar en la vista de Estructura, añadir Items y enlazar cada cual con su correspondiente acción.

Si el lado izquierdo del menú de la aplicación es simplemente una lista de acciones orientadas al archivo con iconos grandes, el lado derecho podría acoger una lista de los ficheros recientemente usados. El control “Ribbon” tiene un soporte específico para manejar este “most recently used” (MRU)¹³⁶. En esta demostración simplificada he decidido manejar solo “Load” y “Save As”. Cada una de las cuales añade una entrada al MRU llamando al método personalizado, que en turnos, invoca el método de control `AddRecentItem`.

¹³⁶ En el código fuente de la demostración también se encontrará el escrito manualmente para gestionar la lista MRU, usando una acción común y procesando en `Caption` en la selección. Considerando la buena calidad del soporte automático, no creo que sea valioso explorar la manera manual. La razón por la cual lo dejo en el código es para mostrar como dinámicamente extender el interfaz de usuario de la “Ribbon” y utilizar acciones compartidas entre diferentes elementos clientes.

336 - Capítulo 10: Ribbon

Esta operación añade una nueva entrada al principio de la lista, y eventualmente borra una entrada existente referida al mismo fichero.

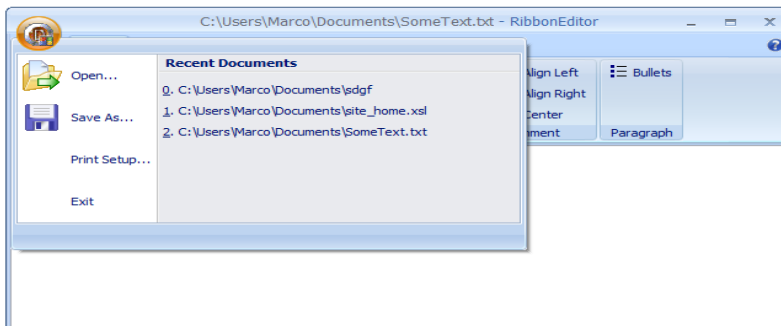
El evento `OnAccept` de las acciones `FileOpen1` y `FileSaveAs1` tiene el siguiente (similar) código, que llama al método `AddToMRU` mostrado debajo de ellos:

```
procedure TRibbonEditorForm.FileOpen1Accept(
  Sender: TObject);
begin
  RichEditor1.Lines.Clear;
  RichEditor1.Lines.LoadFromFile(FileOpen1.Dialog.FileName);
  Ribbon1.DocumentName := FileOpen1.Dialog.FileName;
  AddToMRU(FileOpen1.Dialog.FileName);
end;

procedure TRibbonEditorForm.FileSaveAs1Accept(
  Sender: TObject);
begin
  RichEditor1.Lines.SaveToFile(FileSaveAs1.Dialog.FileName);
  Ribbon1.DocumentName := FileSaveAs1.Dialog.FileName;
  AddToMRU(FileSaveAs1.Dialog.FileName);
end;

procedure TRibbonEditorForm.AddToMRU(
  const strFilename: string);
begin
  Ribbon1.AddRecentItem(strFilename);
end;
```

Cuando un elemento de la lista MRU se selecciona, el control “Ribbon” lanza el gestor de evento `OnRecentItemClick`, el cual se ha codificado de una manera “naïve”, ya que no comprueba si el fichero está actualmente activo en el editor. Además, esta información no es salvada entre sesiones. Todo lo que quiero es mostrar cómo se puede manualmente mostrar los elementos más recientemente usados, obteniendo un efecto como este:



Este es el gestor de evento para la selección de la lista MRU:

```

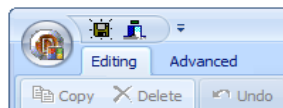
procedure TRibbonEditorForm.Ribbon1RecentItemClick(
    Sender: TObject; FileName: string; Index: Integer);
begin
    RichEdit1.Lines.Clear;
    RichEdit1.Lines.LoadFromFile(FileName);
    Ribbon1.DocumentName := FileName;
end;

```

La página derecha del menú de aplicación puede usarse también para mostrar botones estableciendo la propiedad `CommandType` del objeto `ApplicationMenu` del Ribbon a `ctCommands` (en vez del valor por defecto `ctRecent`). En este caso, cualquier elemento añadido a la colección `RecentItems` se mostrará como un botón. Esto se demuestra en la demostración del menú de aplicación que acompaña a Delphi 2009, un ejemplo interesante ya que funciona de manera similar a Office 2007.

La barra de herramientas de acceso rápido

El segundo elemento gráfico de la “Ribbon” es la barra de herramientas de acceso rápido, una barra de herramientas con operaciones de sistema automáticamente gestionada por el sistema. Se añade a la derecha del selector redondo del menú de aplicación, en este caso mostrando un par de acciones (Save as y Exit):



Al lado de las acciones hay también un botón desplegable para personalización que permite al usuario añadir comandos extra a la barra de herramientas, algo bastante poderoso que puede querer deshabilitar¹³⁷.

¹³⁷ Permitir que los usuarios personalicen sus barras de herramientas es discutible. Un usuario avanzado ciertamente se verá beneficiado pero se puede generar una cantidad importante de llamadas a soporte de los novatos que no puede imaginarse porque los botones de su barra han desaparecido. Como el 80% de los usuarios utilizan el 20% de las funcionalidades la personalización no es siempre buena idea.