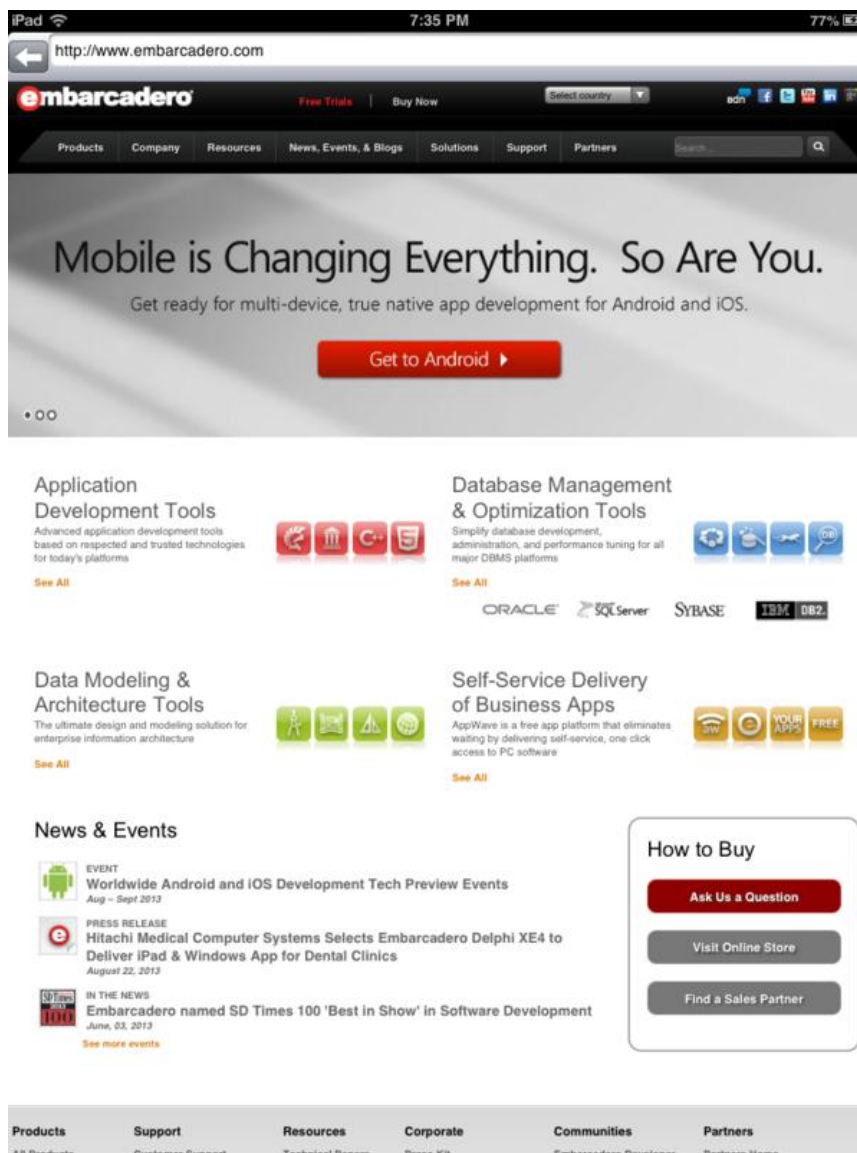


Tutorial Móvil: Usando el Componente Web Browser (iOS y Android)

Usando el componente Web Browser en Aplicaciones Móviles FireMonkey

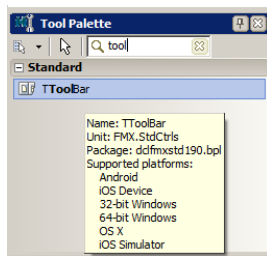
Para plataformas móviles, FireMonkey envuelve el componente Web Browser como el componente **TWebBrowser**. Éste tópico describe cómo crear una simple aplicación FireMonkey Web Browser para plataformas iOS y Android.



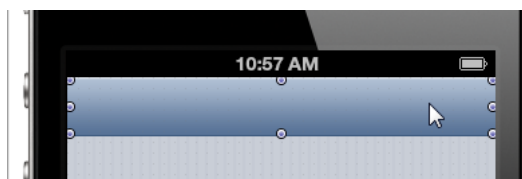
Paso 1: Diseña la Interface de Usuario

1. Seleccione **File > New > FireMonkey Mobile Application - Delphi > Blank Application**.

2. Seleccione el componente **TToolBar** en el **Tool Palette**, y suéltelo sobre el **FireMonkey Mobile Form Designer**. Para encontrar el TToolBar, ingrese los primeros caracteres (como "tool") en la caja de **Búsqueda** del Tool Palette:



3. Luego de soltar el componente, usted puede ver el componente **TToolBar** en la parte superior del Mobile Form Designer:



4. Seleccione el componente **TButton** en el Tool Palette y suéltelo sobre el ToolBar.

5. Seleccione el componente Button en el Mobile Form Designer, y entonces seleccione **priorbuttonbordered** en la propiedad **StyleLookup** en el **Object Inspector**.

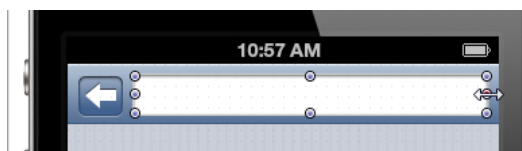
El valor **priorbuttonbordered** StyleLookup para el TButton añade una etiqueta



Back. En dispositivos iOS, es similar a la siguiente imagen:

Para más detalle acerca de seleccionar un estilo en aplicaciones FireMonkey iOS, vea [Tutorial Móvil: Usando un Componente Button con Diferentes Estilos \(iOS y Android\)](#).

6. Seleccione el componente **TEdit** en el Tool Palette y suéltelo en el ToolBar. Asegúrese que el tamaño del control Edit es suficientemente amplia para ocupar el área del ToolBar:



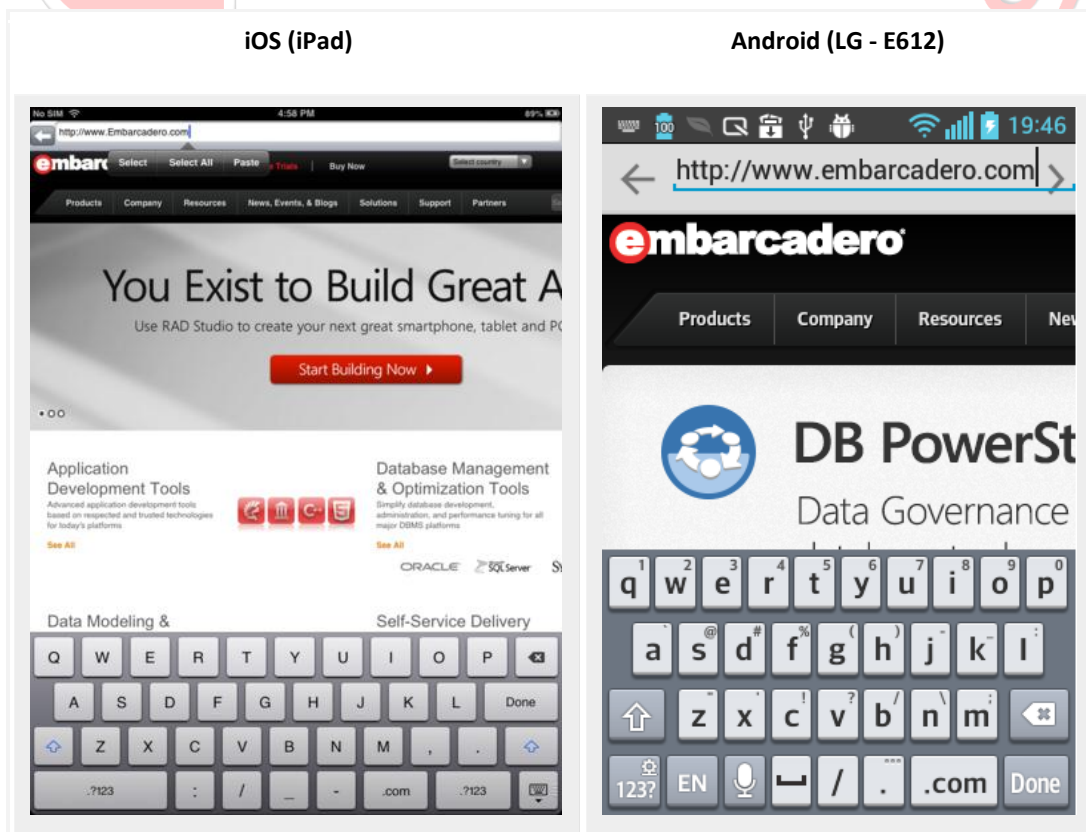
7. Seleccione el componente **TWebBrowser** en el Tool Palette y suéltelo sobre el formulario.
8. Seleccione el componente Web Browser sobre el Mobile Form Designer, vaya al **Object Inspector** y seleccione **alClient** para la propiedad **Align**.

Después que complete estos pasos, el formulario debería lucir como en la siguiente imagen:



Paso 2: Escriba un Manejador de Evento para Abrir una Página Web cuando el usuario Cambia la URL en el Control Edit

A diferencia de las plataformas de escritorio, los dispositivos móviles usan el Virtual Keyboard para ingresar texto como en las siguientes imágenes. El usuario puede completar la acción clickeando el botón "Done".



FireMonkey proporciona varios tipos de manejadores de eventos para cubrir la mayoría de acciones de los usuarios. Después que el botón "Done" es seleccionado, el FireMonkey framework envía un evento `OnChange` al control `TEdit`. Por otra parte, no hay un evento específico para el botón "Back". En esta sección, usted implementará manejadores de evento para soportar ambos escenarios.

Implemente un Método Común para Abrir una Página Web

Antes de implementar manejadores eventos, implementaremos primero un método común para abrir una página web basada en la propiedad `Text` del control `Edit`.

1. En el `Code Editor`, crear la siguiente nueva línea:

```
procedure OpenURL; next to { Private declarations }
type
  TForm34 = class(TForm)
    ToolBar1: TToolBar;
    Button1: TButton;
    Edit1: TEdit;
    WebBrowser1: TWebBrowser;
  private
    { Private declarations }
    procedure OpenURL;
  public
    { Public declarations }
  end;
```

2. Presione **CTRL+SHIFT+C** para crear un marcador de posición al final del archivo:

```
procedure TForm34.OpenURL;
begin

end;
```

3. Implemente el método **OpenURL** como en el siguiente fragmento de código:

```
procedure TForm34.OpenURL;
begin
  WebBrowser1.Navigate(Edit1.Text);
end;
```

Implement an Event Handler for the OnChange Event

1. En el Mobile Form Designer, seleccione el componente `Edit`, y luego en el Object Inspector (Events tab), doble-click en el espacio en blanco al lado del evento **OnChange** para crear el manejador de evento.

El Object Inspector crea un nuevo manejador de evento llamado **Edit1Change**:



2. Complete el manejador de evento añadiendo el siguiente código:

```
procedure TForm34.Edit1Change(Sender: TObject);
begin
  OpenURL;
end;
```

Implementar un manejador de Evento para Soportar la Tecla Enter

No hay un evento específico definido para la tecla Enter. Sin embargo, usted aún puede monitorear dichos eventos a través del evento [OnKeyDown](#).

OnKeyDown da información acerca de la tecla presionada a través de varios parámetros en su manejador de evento. Usted puede implementar éste evento como sigue:

```
procedure TForm34.Edit1KeyDown(Sender: TObject; var Key: Word;
  var KeyChar: Char; Shift: TShiftState);
begin
  if (Key = vkReturn) then
  begin
    OpenURL;

    // Change Focus to Back Button so that virtual keyboard is closed.
    Button1.SetFocus;
  end;
end;
```

Implementar un Manejador de Eventos para el Botón Back

Para implementar un botón **Back** para su Web Browser, usted puede simplemente llamar al método [GoBack](#) del componente Web Browser:

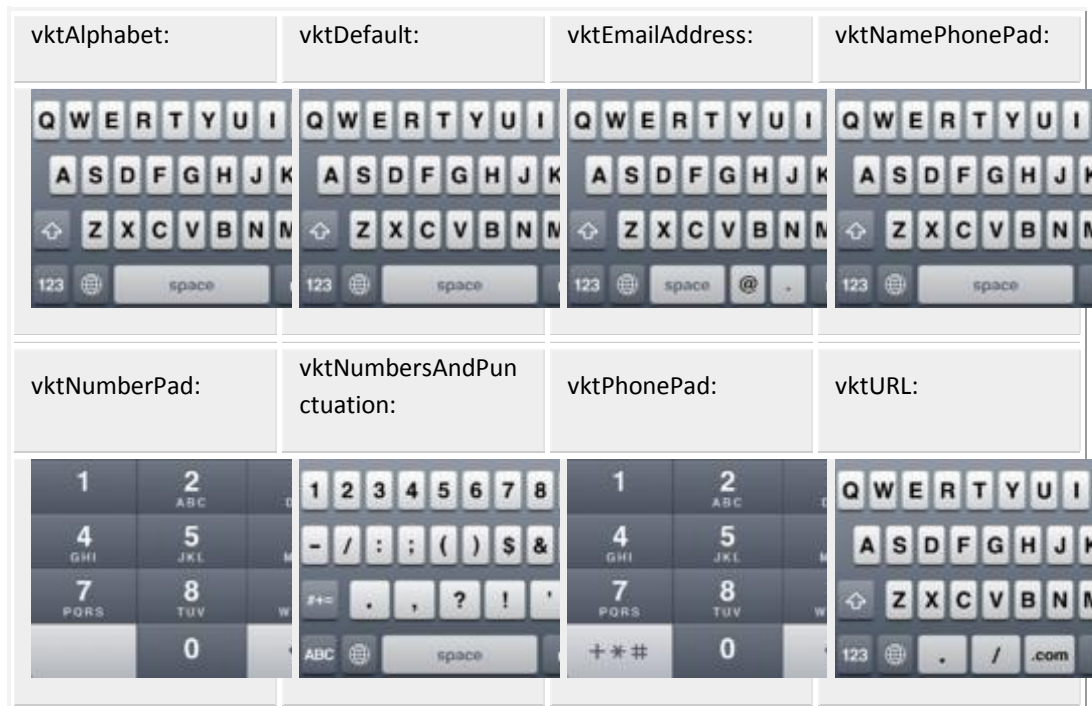
```
procedure TForm34.Button1Click(Sender: TObject);
begin
  WebBrowser1.GoBack;
end;
```

El comportamiento básico está ahora implementado para esta aplicación Web Browser. Intente ejecutar su aplicación ya sea sobre su Simulador iOS o su Dispositivo iOS.

Paso 3: Seleccione el Keyboard Apropriado para la Aplicación Web Browser

Después que usted ejecute su primera aplicación Web Browser, puede darse cuenta que el Virtual Keyboard no está optimizado.

iOS proporciona varios teclados virtuales de la siguiente manera:



Android proporciona varios teclados virtuales de la siguiente manera:



Seleccione **vktURL** como el tipo de teclado virtual adecuado en la propiedad **KeyboardType** para el control Edit, y establezca la propiedad **KillFocusByReturn** a True :

