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: Diseñe la Interface de Usuario

1. Seleccione File > New > FireMonkey Mobile Application - Delphi > Blank Application. Seleccione el componente TToolBar en el Tool Palette, y suéltelo sobre el FireMonkey Mobile Form Designer. Para encontrar el TToolBar, ingrese los primeros caractéres (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 priortoolbuttonbordered en la propiedad StyleLookup en el Object Inspector.

El valor priortoolbuttonbordered StyleLookup para el TButton añade una etiqueta

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

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).

 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:



- 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 imágen:



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".

iOS (iPad)	Android (LG - E612)
No 544 (* 4-50 PM epv, KX http://www.Embarcadero.com Combarc Select All Paste read Products Company Resources News, Events, & Blogs Solutions Support Parmers	Image: Second system of the system of th
You Exist to Build Great A Use RAD Studio to create your next great smartphone, tablet and PC Start Building Now ▶	Products Company Resources New DB PowerSt
Application Development Tools Advanced agesteen development tools to bady platforms See All Data Modeling & Self-Service Delivery	Data Governance q ¹ w ² e ³ r ⁴ t ⁵ y ⁶ u ⁷ i ⁸ o ⁹ p ⁹
Q W E R T Y U I O P A S D F G H J K L Done	a s d f g h j k I \therefore z x c v b n m \checkmark 1237 EN Q \checkmark /com 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: 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

 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:

Desarrollo de Apps Móviles con Delphi (iOS y Android)

📲 Object Inspector	7 8	9													
i dit1 TEdit							10	.57							
Properties Events				_	-	 	 10	ю,	A	()					
LiveBindings LiveBindings		1			5: :					Ĩ.					1
OnApplyStyleLooku				$ \ge $	<u>) </u>	 	 			0	 			 	i (
OnCanFocus															
OnChange Edit1Change		1													
OnChangeTracking OnClick		٦	· · · ·												

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:



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:



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

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

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

vktAlphab	et:	vktDef	ault:			vkt	Ema	ilAddr	ess:		vkt	Nam	nePho	oneP	ad:	
QWE	RTYUI	Q W	ER	ΤY	UI	Q	WE	R	TΥ	UI	Q	w	ER	TY	U	
ASD	FGHJ	KAS	DF	GH	HJ	ĸ	S	DF	GH	IJ	K A	S	DF	G	ΗJ	ĸ
🔶 Z X	CVBN	N 💠 Z	xc	VE	BN	Ŷ	z	хc	VE	N	N 💠	Z	xc	v	BN	N
123 🖶	space	123		space		123	۲	space	@		123	•		space		
vktNumbe	erPad:	vktNur ctuatio	mbers <i>i</i> on:	AndP	un	vkt	Phor	nePad	:		vkt	URL	:			
1	2 ^**	1 2	34	56	78		1		2 ABC		Q	w	ER	TY	U	
4 6H1	5 JKL	- /	::	()	\$ &		4 6H1		5 JKL		A	S	DF	G	ΗJ	ĸ
7 PORS	8 TU V	w #+=	. ,	?	1		7 PORS		8 TUV		Ŷ	z	xc	v	BN	N
	0	ABC 📵		брасе		+	*#		0		123	•		1	.com	

iOS proporciona varios teclados virtuales de la siguiente manera:

Android proporciona	varios teclados virtuale	s de la siguiente mane	era:
vktAlphabet:	vktDefault:	vktEmailAddress:	vktNamePhonePad:
a w e e t y u i o		qwartyui asdfghjk	0 - + Press 1 2.000 3.007
asdfghjk Ozxevbnm.	• z x c v b n m .	• z x c v b n m	. () N 7 Hans 8 TUN 9 Hans
	* / 0	1 0	• 0
vktNumberPad:	vktNumbersAndPun ctuation:	vktPhonePad:	vktURL:
- + , 1 2 3	g w e r t y u i o	- + Peer 1 2 3 -	qwertyuio
* / , 4 5 6		4 5 6 . () N 7 8 9 .	asdfghjk mozxcvbnm,
() = 7 8 9			
	na ≹ /		• • • • • • • •

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



