## Tutorial Móvil: Usando LiveBindings para Poblar un ListView (iOS y Android)

Este tutorial muestra cómo usar el LiveBindings Designer para poblar un componente FireMonkey ListView desde un TPrototypeBindSource conteniendo algunos datos de ejemplo. El tutorial muestra cómo agregar datos de ejemplo y cómo hacer el enlace entre el prototyping source y el componente list view a fin de llenar la lista.

Como con cada LiveBinding, éste tutorial **no requiere código**. Sin embargo, a fin de crear una aplicación útil, usted necesita agregar manejadores de evento y otro código.

### Paso 1: Creando el Proyecto

1. Crear un nuevo proyecto. Escoja una FireMonkey Mobile Application para este ejemplo. En el asistente, escoja **Blank Application**.



- 2. En el Tool Palette, localize un componente TListView y suéltelo sobre el formulario.
- 3. Agregue un componente TPrototypeBindSource al formulario.
- 4. Sobre el formulario, seleccione el componente ListView1, y entonces en el Object Inspector, setear la propiedad Align a alClient.

Después de colocar los componentes necesarios sobre el formulario móvil, el formulario debería lucor como la siguiente imágen.



### Paso 2: Agregando Campos

1. Click-derecho en el componente TPrototypeBindSource y seleccione Add Field....



2. Desde la caja de diálogo Add Field, seleccione ColorsNames y click en OK.

Add Field					t	x
ColorsName2						
Field Data:						_
Generator			Type			*
AlphaColors BitmapNames Bitmaps Booleans Colors			ftUInteger ftString ftBitmap ftBoolean ftUInteger			m
ContactBitmaps ContactBitmapsL ContactNames			ftBitmap ftBitmap ftString			
Shuffle Values			Repeat	Values		
Sample Value						
Darkgoldenrod						
Value type:	string			Ne	xt Value	
	C	ж	Cano	el	Help	

### Paso 3: Creando LiveBindings

 Abra el LiveBindings Designer, si no está ya abierto, y enlazar la propiedad ColorsName1 del TPrototypeBindSource a la propiedad Item.Text del list view.

El componente list view pobla automáticamente los items con nombres de color desde el componente de datos de prototipos.



# 2. En el Structure View, ubique el componente ListView1 y luego click en la

jerarquía ItemAppearance.

- 3. En el Object Inspector, localize la propiedad **ItemAppearance** y cambie su valor a ImageListItemRightButton.
- 4. Cambie la propiedad **ItemEditAppearance** a

ImageListItemRightButtonDelete.



En este punto, el componente list view está configurado para mostrar una imágen sobre el lado izquierdo del texto del item, y un botón en el lado derecho del texto del item. La imágen y botón son poblados con datos de ejemplo que usted agregará en el siguiente paso.

### Paso 4: Agregando Más Campos

Usted necesita agregar dos campos más con el fin de hacer que el componente list view muestre una imágen y un texto en el botón asociado con cada ítem de la lista.

- 1. Click-derecho en el componente TPrototypeBindSource y seleccione Add Field....
- En la caja de diálogo Add Field, <u>Ctrl+Click</u> para seleccionar los campos de datos Bitmaps y Currency.
- 3. Vaya al LiveBindings Designer y haga lo siguiente:
  - 1. Conecte la propiedad Bitmap1 del prototyping source data a la propiedad Item.Bitmap del componente list view.

Este paso añade un botón representando el color y número de cada item del list view, tales como Blue 19.

2. Conecte la propiedad CurrencyField1 desde el prototyping source data a la propiedad Item.ButtonText del componente list view.

Este paso muestra el valor del campo currency sobre el botón localizado en el lado derecho de cada item del list view.

	47 Cornflo	owerblu	e	43.43
	<sup>19</sup> Blue			60.6
	1: 1	Insert	Modified	
LiveBir	ndings Designer			
12 - De	efault Layer			
	L Start Connect			
Sel	ListView1			
Sele	ListView1 ectedValue			PrototypeBindSource1
Sel Syn	ListView1 ectedValue	]		PrototypeBindSource1 *
Sel Syn Item	ListView1 ectedValue ich n.Text			PrototypeBindSource1 * ColorsName1
Sele Syn Item	ListView1 ectedValue ich n.Text n.Bitmap			PrototypeBindSource1 * ColorsName1 Bitmap1
Sele Syn Item Item	ListView1 ectedValue hch h.Text h.Bitmap h.ButtonText			PrototypeBindSource1 * ColorsName1 Bitmap1 CurrencyField1
Seli Syn Item Item Item	ListView1 ectedValue ich n.Text n.Bitmap n.Bitmap n.ButtonText n.LookupData			PrototypeBindSource1 * ColorsName1 Bitmap1 CurrencyField1
Seli Syn Item Item Item	ListView1 ectedValue ich n.Text n.Bitmap n.ButtonText n.LookupData nHeader.Text			PrototypeBindSource1 * ColorsName1 Bitmap1 CurrencyField1

4. Click en OK.

Ahora el list view muestra algunos datos de color asociado con cada item y también muestra datos de dinero sobre el botón asociado en cada ítem de la lista.

### Paso 5: Agregando el Manejador de Evento onButtonClick

Para crear una aplicación práctica, puede agregar el manejador de evento **onButtonClick** que se dispare cuando usted hace click en un item del ListView.

#### Para añadir elmanejador de evento onButtonClick

- 1. En el formulario de la aplicación móvil, seleccione el componente ListView1.
- 2. En el Object Inspector, abra la ficha Events, y luego doble-click en OnButtonClick.
- 3. En el Code Editor, implemente una manejador de evento apropiado OnButtonClick.

El siguiente código de ejemplo agrega el manejador de eventos que muestra una caja de mensaje cuando usted hace click en un ítem del ListView:

```
procedure TForm1.ListView1ButtonClick(const Sender: TObject;
  const AItem: TListViewItem; const AObject:
TListItemSimpleControl);
begin
   ShowMessage(AItem.Text + ' ' +AItem.ButtonText + ' is
clicked.');
end;
```

### Los Resultados

Ejecute la aplicación en su dispositivo móvil, pulsando F9 o seleccionando Run > Run.

Android (LG - E612	<u>?</u> )
W 💩 🕞 🖻 🖞 🐳 🛛 刘	ıll 💈 16:25
Darkgoldenrod	67.67
46	70 70
Ivory	79.79
<sup>16</sup> Mediumaguamarine	78.78
Mediumvioletred	81.81
" Ghostwhite	37.37
Sandybrown	41.41
" Comflowerblue	43.43
	10110
" Blue	60.6
<sup>25</sup> Greenvellow	71 71
Greenyenow	1.11
<sup>22</sup> Powderblue	15.15

Si usted clickea el item Mediumaquamarine, la aplicación muestra el siguiente mensaje:

