

Review: Eschalon Power Controls

Reviewed by Dave Jewell

One of the first third party component sets for Delphi was Eschalon's Power Controls, who also market Eschalon Setup Pro (now with BDE support). We'll take a whirlwind tour through Power Controls and see what's on offer. The screenshot shows most of the controls in action.

Doing The Splits

The main part of the screenshot comprises a "splitter" window. The border between the cloud bitmap and the left hand grey panel can be moved to the left or right by using the mouse (which changes to a double-ended arrow over this movable border). The `TEPSplitter` control has an `Orientation` property so you can do a vertical split or a horizontal split. Each of the two 'panes' of the splitter component is a container (just like group boxes, panels, notebooks and so forth) meaning that you can place one or more other components into each pane. It's quite flexible, allowing custom panes and splitter bars if needed.

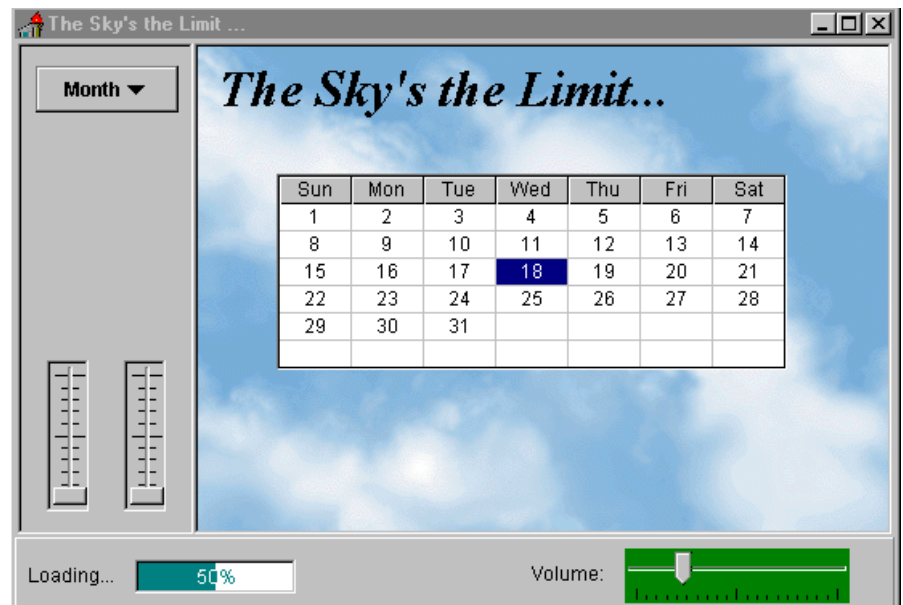
A Menu In A Button...

The `TEPMenuButton` (the 'Month' button in the screenshot) is used to provide the sort of non-standard interface that you often see in Microsoft software [*Wry smile... Editor*]. When you click the button, an associated menu appears.

You set up a list of menu items by defining a list of strings using the `Items` property. An `OnChange` event is generated when the user makes a selection, you can programmatically determine whether or not the menu is in a 'dropped' state and it's also possible to display a small checkmark next to whatever is currently selected.

Sliders A La Carte

Below the menu button, you can see a couple of slider components and another appears on the status bar. These are courtesy of the `TEPSlider` component, which can



➤ Put a real 'Wow!' factor into your app with Eschalon Power Controls

be oriented either vertically or horizontally and (as you can see) it's possible to change the shape of the "thumb". In fact, it's possible to write your own `OnDrawThumb` handler, meaning that you could (for example) have a thumb which changed as you moved it from one end of the slider track to the other.

Quite a number of visual variations are possible. For example, you can control the placement of 'tick marks' or dispense with them altogether.

Improving Your Image

I used the cloud bitmap (from Windows 95) with a `TEPImage` control to produce a nice backdrop. `TEPImage` improves on `TImage` in a number of useful ways. For starters, you can flip the image horizontally, vertically, or both. The flipping is reasonably fast for small images (I discovered, though, that it's unreliable when used with 256-colour bitmaps).

Another nice feature of this control is the ability to specify a particular colour in the property `TransparentColor` to render any matching colour values in the bitmap transparent. There's also a `Style` property which, amongst

other things, allows you to tile an area with a specified bitmap, just as Windows uses bitmaps to tile the desktop wallpaper.

The `GraphicAsBitmap` method allows you to return a bitmap handle to the currently displayed image even if the `Picture` property is set to an `.ICO` or `.WMF` file. This is a useful way of converting metafiles to bitmaps. Finally, an `ImageControl` property can be used to point to a `TImage` component which contains the bitmap to be displayed. This can be useful where you want the same image to be displayed in several places within the same program: rather than copying the bitmap data, you have several `TEPImage` components pointing at the same `TImage`.

Meter Madness

The control on the left hand side of the status bar is a `TEPMeter`. This sort of component is typically used in installer programs and in file copying operations, to give the user visual feedback. Again, you can customise to your heart's content. The component even makes an `OnDrawMeter` event handler available, so you can take over the meter drawing. There's an

`OnSelectColor` event too, which can be used to change the bar colour on-the-fly. Your meter could start off green and go through amber to red. As with `TEPSlider`, you have to manually resize the control after changing the orientation.

Picking Up Some Useful Tips...

A particularly cute inclusion in the Power Controls package [*If you like that sort of thing...! Editor*] is the `TEPTipDialog`. It's becoming fashionable to include a "Tip of the Day" dialog in applications. The Tip dialog is a non-visual component which you simply place on a form and then display by calling its `Execute` method. `TEPTipDialog` will randomly select from up to 16,000 separate tips which can be entered from the string list editor in the normal way or loaded from a text file. If this isn't enough, you can point the dialog at more tips in your application's help file by specifying a valid context number.

As with most of the Eschalon controls, the Tips dialog is very flexible and there's no reason why you couldn't use it for some other

purpose. For example, an enterprising shareware writer might turn it into a "nag screen" complete with 16,000 reasons why it's time you registered the software!

Odds And Ends...

Another 'canned' dialog component is `TEPAboutDialog`. No prizes for guessing what this does!

Two calendar components are included. `TEPCalendar` improves on the built-in `TCalendar` component in a number of ways. There's also a much simpler calendar control, `TEPCalendarPopup` which simply pops up for easy date selection.

Power Controls also sports a pretty good spin button, including an `AssociatedControl` property so that it can be pointed at an edit box. It's a shame that its default behaviour isn't to increment or decrement the value in the edit box: you have to add this code yourself.

There's a `TEPBorder` control which allegedly consumes less system resources than `TPanel` or `TGroup`. Lastly, Eschalon provide you with the source code of a handy splash screen form.

Summary

Eschalon Power Controls are professional and easy to use. They're also reasonably priced, and source code is available, so you'll be able to re-compile them all for the 32-bit version of Delphi once it's released. You can get Power Controls in the UK from Grey Matter (Tel 01364-654100) for £105, or £145 with source code (plus VAT).

By the way, if you want to try out Eschalon Power Controls for yourself, you can download the file `EPCTRLS.ZIP` from the borland/delphi.files conference on CIX or the CompuServe DELPHI forum. The controls in this .ZIP file are fully functional but will only work while Delphi itself is running.

Dave Jewell is a freelance consultant, programmer and technical journalist. He's the author of "Instant Delphi Programming" by Wrox Press and is working on a component-oriented Delphi book due in the Spring. Email Dave as djewell@cix.compulink.co.uk