New & Used

By Bill Todd

1stClass

A Set of First-class Delphi Components

If you're looking for a set of controls to give your applications a distinctive look and feel, look no further than 1stClass from Woll2Woll Software. 1stClass includes Shape Button, Image Button, Button Group, Color Combo, Color List, Tree View, DB Tree View, Tree Combo, Font Combo, Image Form, Imager, Label, OutlookBar, and Status Bar components that offer a host of features not found in their Delphi counterparts.

If you like the look of Microsoft Outlook, the *TfcOutlookBar* component will let you duplicate it, and more. Working with the OutlookBar is surprisingly easy. The first step is to drop the OutlookBar on your form and set its *Align* property to position the bar either vertically at the left or right side of the form, or horizontally at the top or bottom. Next, select the *OutlookItems* property and click the ellipsis button to open the Collection Editor. Using the context menu or toolbar in the Collection Editor, add one or more pages to the OutlookBar. The page buttons are actually *TfcShapeBtn* components, so you can assign a graphic to the button's *Image* property and the button assumes the shape of the graphic. Click the active page in the OutlookBar and select its *Images* property to assign an ImageList component that contains the images you want to use for the items on this page. Next, select the *Items* property. Open the Collection Editor for the *Items* property and add the number of items you need for the current page. Now select each item, set its *ImageIndex* property to the correct image in the ImageList component, then switch to the Events page of the Object Inspector and create an *OnClick* event handler. That's all there is to it. Another remarkable feature of the OutlookBar is the ability to embed other components in the bar.

The *TfcImager* control is similar to Delphi's *TImage* component, but with much more power. The *Picture* property of *TfcImager* lets you load a JPEG, bitmap, icon, enhanced metafile, or Windows metafile image. Using other properties, you can change the appearance of the image in various ways. For example, you can adjust the contrast, brightness, and saturation, or give the image an embossed, blurred, or sponged look. You can also flip or rotate the image in any direction and apply a tint. Figure 1 shows an Imager with a digital camera photograph loaded. You can also set the *Imager* property of the *TfcDBTreeView* or *TfcOutlookBar* controls so they'll use the Imager to supply a background bitmap.

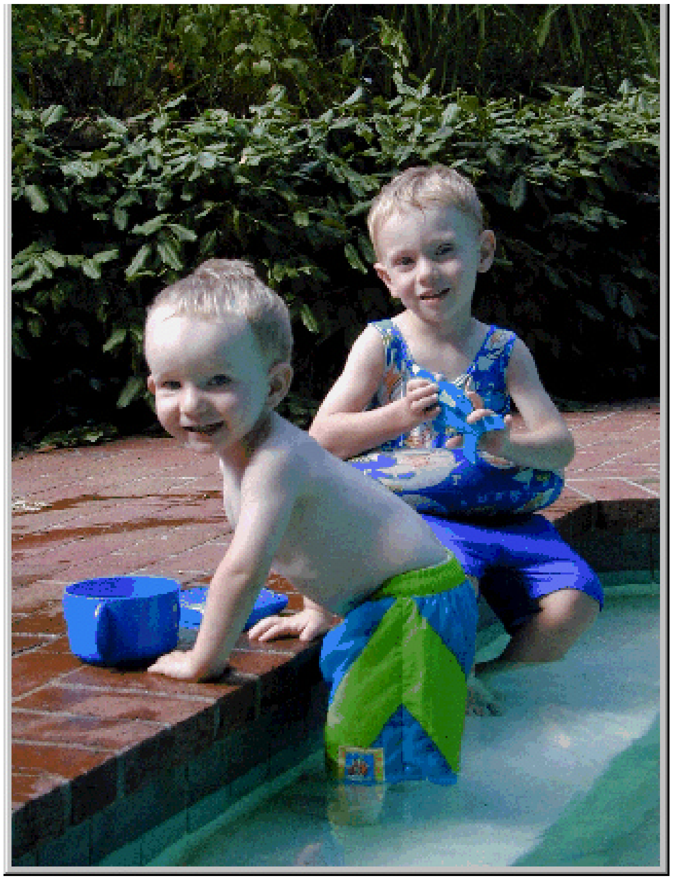
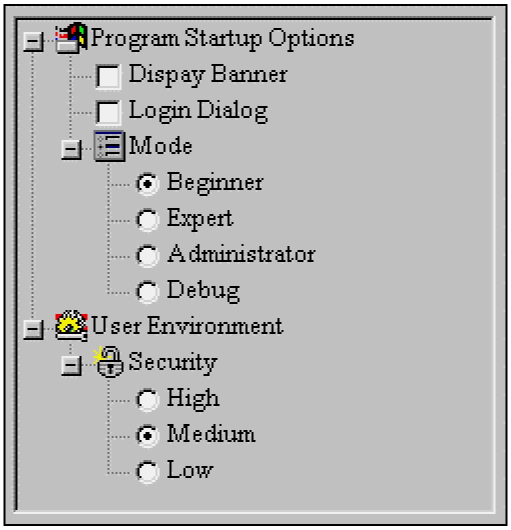


Figure 1: *TfcImager* with a JPEG photo loaded.

1stClass also includes a Tree View component and a Tree Combo component. Each node in the *TfcTreeView* consists of a label, one or more images, and a list of optional checkboxes or radio buttons. Each node can also have a list of subnodes, which in turn can have subnodes, allowing you to build a tree of any depth. The *TfcTreeCombo* shares most of the capabilities of the Tree View in a combobox format. The Tree Combo component can also be embedded in the *TwwDBGrid* component that is part of Woll2Woll's InfoPower Suite. Figure 2 shows a Tree View that contains both checkboxes and radio buttons.

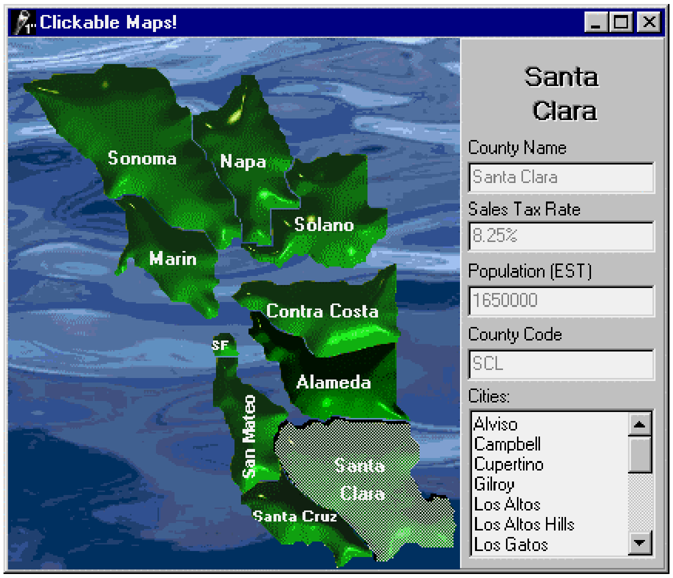


```
⊟ Program Startup Options
    □ Dispay Banner
    □ Login Dialog
  ⊟ Mode
      ● Beginner
      ○ Expert
      ○ Administrator
      ○ Debug
⊟ User Environment
  ⊟ Security
      ○ High
      ● Medium
      ○ Low
```

Figure 2: The *TfcTreeView*.

TfcDBTreeView is a data-aware Tree View component that can automatically model any master/detail relationship regardless of the complexity or the number of datasets involved. The datasets can consist of any combination of *TTable* and *TQuery* components. To use the DB Tree View component, you must set the *DataSourceFirst* property to the DataSource component for the dataset that is the root node of the tree. Next, you must set the *DataSourceLast* property or the *DataSource* property. Setting *DataSourceLast* will cause all DataSources between the root node and the DataSource specified in *DataSourceLast* to appear in the tree. Using the *DataSources* property lets you specify a list of DataSources to display. The *DataFields* property provides control over which fields are displayed at each level in the tree. Figure 3 shows a data entry form that incorporates a DB Tree View. The DB Tree View provides a live view of the data, so navigating from one record to another in the DB Tree View will change the current record in any form that uses the same dataset, and vice versa. Note also the use of an Imager component to provide a graphical background for the DB Tree View in Figure 5.

Figure 3: The sample Customers, Orders, and Items tables in a *TfcDBTreeView*.



Santa Clara

County Name: Santa Clara

Sales Tax Rate: 8.25%

Population (EST): 1650000

County Code: SCL

Cities:
- Alviso
- Campbell
- Cupertino
- Gilroy
- Los Altos
- Los Altos Hills
- Los Gatos

TfcImageBtn is a class with all the features of the standard Delphi *TButton*, *TBitBtn*, and *TSpeedButton* classes, but it gets its shape and appearance from an image. To give the button a different look when it's depressed, assign different images to the button's *Image* and *ImageDown* properties. The up and down images can be completely different so the button will have a different appearance and shape when it's down. You can even create clickable maps, as shown in Figure 4, by using an ImageBtn for each area of the map.

Figure 4: A clickable map using *TfcImage* buttons for the counties.

TfcShapeBtn combines the functionality of *TSpeedButton* and *TBitBtn* with the ability to create buttons of virtually any shape. Figure 5 shows the standard shapes for the ShapeBtn. If the standard shapes don't meet your needs, you can create custom shapes by defining a list of points that describe the shape. You can control the button's color as well as the highlight colors that are used for three-dimensional effects. Both the image button and shape button provide *OnMouseEnter* and *OnMouseLeave* events that you can use to make the button change its appearance when the mouse cursor passes over the button.

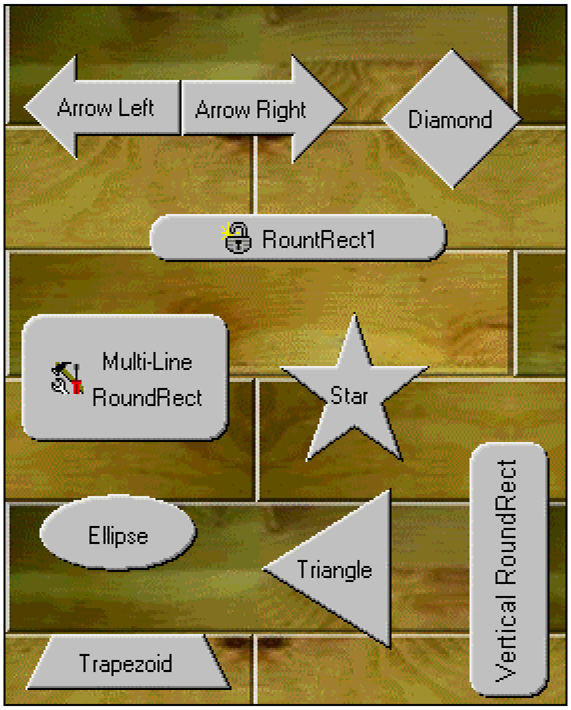


Figure 5: The standard shapes for *TfcShapeBtn*.

TfcButtonGroup lets you easily create a collection of shape buttons or image buttons. You can arrange the buttons in any number of rows and columns, as well as adjust the spacing between adjacent buttons. Three behaviors are available when the buttons are clicked: The buttons can behave as a group of radio buttons, as toggle buttons, or as click buttons. If you set the *ClickStyle* property to *bcsRadioGroup*, only one button can be depressed at a time. Optionally, you can allow all the buttons to be up, or you can require that one always be depressed. The *bcsCheckList* option allows any combination of buttons to be depressed at one time, simulating a group of checkboxes. Choosing *bcsClick* makes the buttons behave as regular buttons so they remain depressed only as long as you hold the mouse button down. The *TfcButtonGroup* class also provides a *Transparent* property. When set to True, all areas of the button group not covered by a button will be transparent. Figure 6 shows a button group with the *Transparent* property set to True so the background graphic fills the area around the buttons.

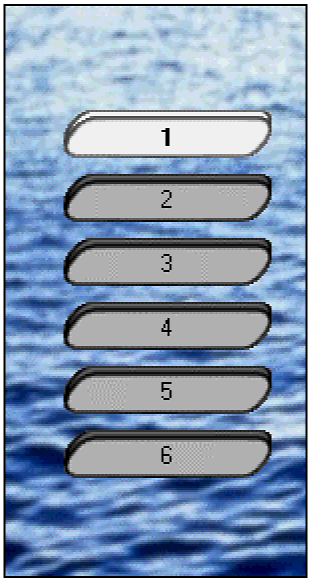


Figure 6: A *TfcButtonGroup* with transparent background.

When you drop the *TfcImageForm* control on a Delphi form, you create a form whose shape is defined by the non-transparent areas of the image specified by the *Picture* property. The *TfcImageForm* also sets the host form's *BorderStyle* to *bsNone* to remove the form's border and title bar. The result is a form that consists only of the graphic assigned to the *Picture* property. Although the form has no title bar, users can still drag it by depressing the left mouse button over any part of the form. You can also assign any control on the form to the ImageForm's *CaptionBarControl* property. Now dragging the assigned control will drag the form, and attempting to drag any other area of the form will not.

The *TfcColorCombo* and *TfcColorList* components provide users with tools to easily select a color. The Color Combo component is data-aware and can be used to store colors in a database table by storing either its name or integer value. You can choose from system or standard color lists, or construct your own custom colors. If you're an InfoPower user, you can embed the ColorCombo and all the other 1stClass combo components into the InfoPower Grid and RecordView components.

TfcStatusBar resembles the Delphi *TStatusBar* but with much more flexibility. You can add any number of panels to the status bar and automatically display the hint properties from controls on the form; the current date, time, or both; the state of [CapsLock], [NumLock], [ScrollLock], and [Insert]; and the computer name or the name of the current user. You can also embed your own components in the StatusBar's panels. You can also size the panels in a status bar proportionally so the panels will occupy the same percentage of the status bar's width when the form is resized.

TfcFontCombo is a combobox for selecting fonts with an optional most-recently-used list at the top. Setting the *MaxMRU* property to a value other than zero will cause the most recently used fonts to appear at the top of the drop-down list. *TfcLabel* is a label component that supports shadow, extrusion, engraved, embossed, and outline effects.

Conclusion

1stClass is an excellent set of components for adding sophistication to the user interface of your Delphi programs. Woll2Woll still believes in providing user manuals, and 1stClass comes with a well-indexed, 178-page, wire-bound manual that provides a complete reference to all the components and their custom properties, methods, and events. 1stClass comes in a Standard version, which supports Delphi 3 through 5, and a Professional version, which supports Delphi 3 through 5 and C++Builder 3 and 4. You can give 1stClass a try by downloading the demonstration version from Woll2Woll's Web site.

Bill Todd is President of The Database Group, Inc., a database consulting and development firm based near Phoenix. He is co-author of four database programming books and over 60 articles, and is a member of Team Borland, providing technical support on the Borland Internet newsgroups. He is a frequent speaker at Borland Developer Conferences in the U.S. and Europe. Bill is also a nationally known trainer and has taught Paradox and Delphi programming classes across the country and overseas. He was an instructor on the 1995, 1996, and 1997 Borland/Softbite Delphi World Tours. He can be reached at mailto:bill@dbginc.com.

Informant Fact File

If you're looking for a set of controls to give your applications a distinctive look and feel, look no further than 1stClass from Woll2Woll Software. 1stClass provides an excellent set of components for adding sophistication to the user interface of your Delphi programs.

Woll2Woll Software
2217 Rhone Drive
Livermore, CA 94550

Phone: (800) 965-2965
Fax: (925) 371-1664
E-Mail: mailto:sales@woll2woll.com
Web Site: http://www.woll2woll.com
Price: Standard version, US$199; Professional version, US$299.

Product review by Bill Todd.