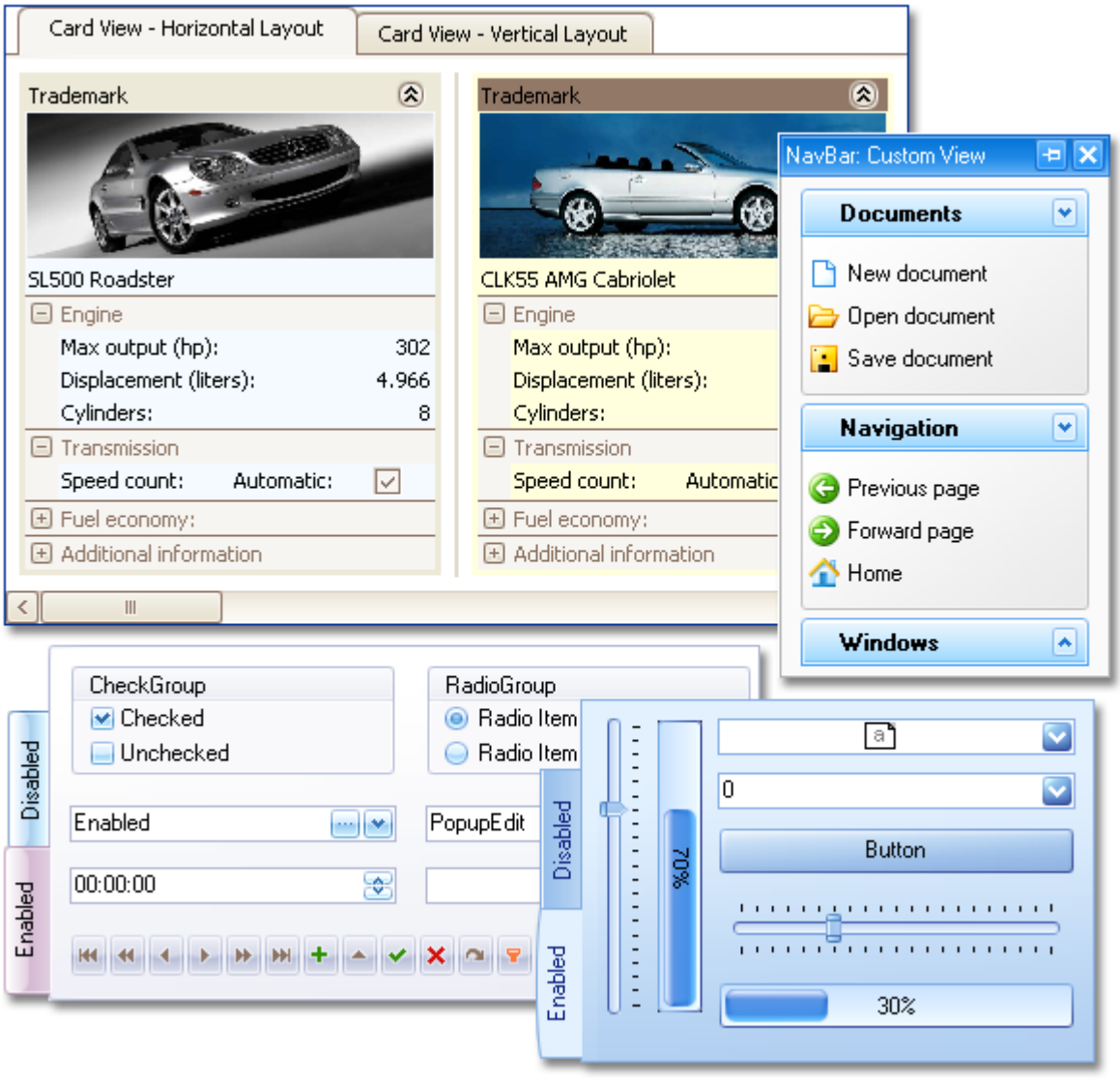
Thank you for your interest in the ExpressSkins Library Beta. This document contains a brief overview of skin options introduced in the **ExpressSkins Library**, as well as step-by-step instructions on how to create custom skins and use them in your applications. Please note that these options are subject to change.

Important Note: The **ExpressSkins Library** requires installation of the Microsoft GDI+ library. Gdiplus.dll is included with Windows XP. GDI+ is available as a redistributable file for Windows NT 4.0 SP6, Windows 2000, Windows 98, and Windows Me. To download the latest redistributable, see http://www.microsoft.com/downloads/details.aspx?FamilyID=6a63ab9c-df12-4d41-933c-be590feaa05a&DisplayLang=en.



New in Beta 3

- Skins for the **ExpressScheduler v2** and **ExpressBars v6** (except for Ribbon elements).
- Three new built-in skins that mimic Microsoft Office® 2007 color schemes: Black, Blue, and Silver.

Introduction to Skins

Skins allow you to create visually exciting interfaces. A skin contains bitmaps showing all control elements in their various states (normal, hot-tracked, selected, pressed, etc.) Since each control is painted using the corresponding bitmap, the appearance of the entire skinned application does not generally depend on the Operating System. By using skins you can introduce the XP look and feel experience into applications that are not running the XP OS.

Skin Store

Skins are stored in resource files along with application projects. To streamline the distribution of skinned applications, skins are added into the compiled executables.

Applying Skins

This Beta version adds skin support to **forms** and the following VCL controls: **ExpressQuantumGrid v6**, **ExpressQuantumTreeList v4**, **ExpressVerticalGrid**, **ExpressPivotGrid**, **ExpressEditors Library**, **ExpressLayout Control**, **ExpressPageControl**, **ExpressNavBar**, **ExpressScheduler v2**, **ExpressBars v6** (except for Ribbon elements), and **Docking Library**.

To apply a skin to an entire application built with these controls, use our **TdxSkinController** or **TcxLookAndFeelController** component, and assign their SkinName property. To apply a skin to one of controls listed above, use the control's LookAndFeel.SkinName property. For Express editors, use the Style.LookAndFeel.SkinName property.

This Beta version ships with seven built-in skins: Caramel, Money Twins, Lilian, iMaginary, Black, Blue, and Silver. You are free to apply them, or create new skins based on our built-in skins.

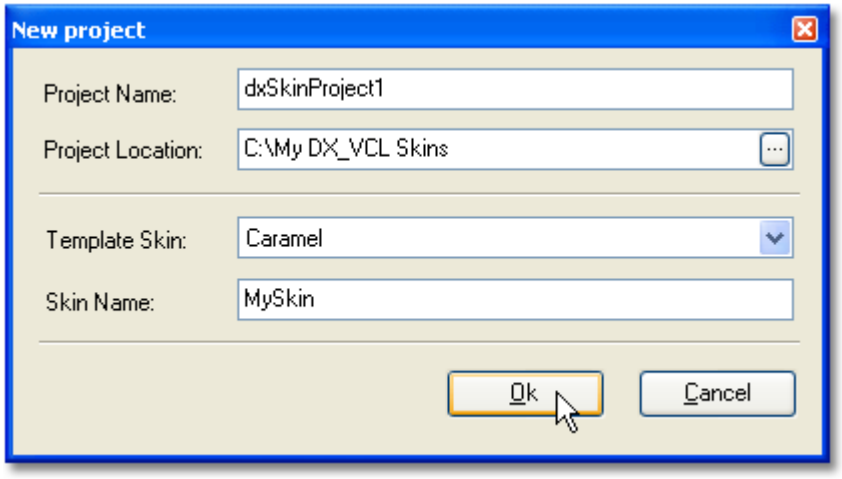
Creating Skins

The **Skin Editor** shipped with the **ExpressSkins Library** allows you to create new skins, and build resource and unit files to incorporate these skins into your applications.
To create a skin from scratch, perform the following steps:

1. Run the **Skin Editor**.

2. Create a new project. To do this, click the **New** button in the toolbar, or select 'New' in the File menu.

3. In the ‘New project’ dialog, specify the project name, its location, the template skin and the name of the new skin. You can have as many skins in a single project as you desire. Use the Project menu to invoke the Project Manager, which will help you manage skins within your project.



4. Customize the appearance aspects of control elements by loading images and glyphs for corresponding elements, and changing image properties as your needs dictate.

5. Once you've completed your modifications, click ‘Build’ in the Project menu to create the resource and unit files that contain the generated skins and skin registration routines. These files will have the same name as the project, and can be found at the project location.

To use a newly created skin, add the created unit into your Delphi/C++ Builder project, and specify the desired skin name (for instance, ‘MySkin’) to the SkinName property as shown in the previous section.