

Eigenschaften

ActiveControl: Sie können ermitteln oder festlegen, welches Steuerelement im Formular den Fokus hat. In einer Anwendung kann immer nur ein Steuerelement aktiv sein.

AutoScroll: Wenn der Wert true ist, erscheinen die Bildlaufleisten bei Bedarf automatisch. Dies ist immer dann nötig, wenn der Benutzer das fensterorientierte Steuerelement so verkleinert, daß einige der enthaltenen Steuerelemente nicht vollständig sichtbar sind. Der Wert false verhindert die automatische Anzeige.

HorzScrollbar: Es wird die horizontale Bildlaufleiste ausgeblendet, eingeblendet und im Aussehen verändert.

VertScrollbar: Es wird die vertikale Bildlaufleiste ausgeblendet, eingeblendet und im Aussehen verändert.

BorderStyle: Sie können Aussehen und Funktionsweise des Formularrahmens festlegen. Die Eigenschaft ist eine Menge mit folgenden Werten vom Typ TFormBorderStyle:

Wert	Bedeutung
bsDialog	Standardrahmen ohne Größenänderung.
bsSingle	Einfacher Rahmen ohne Größenänderung.
bsNone	Kein sichtbarer Rahmen, keine Größenänderung.
bsSizeable	Standardrahmen mit Größenänderung.
bsToolWindow	Wie bsSingle, jedoch mit kleinerer Titelleiste.
bsSizeToolWin	Wie bsSizeable, jedoch mit kleinerer Titelleiste.

Font: Um eine andere Schriftart zu verwenden, geben Sie ein neues TFont-Objekt an. Wenn Sie die Attribute einer Schrift ändern wollen, weisen Sie den Eigenschaften CharSet, Color, Height, Name, Pitch, Size oder Style des TFont-Objekts die gewünschten Werte zu.

Formstyle: Sie können den Stil des Formulars bestimmen. Die Eigenschaft kann einen der folgenden Werte enthalten:

Wert	Bedeutung
fsNormal	Das Formular ist weder ein übergeordnetes noch ein untergeordnetes MDI-Fenster.
fsMDIChild	Das Formular ist ein untergeordnetes MDI-Fenster.
fsMDIForm	Das Formular ist ein übergeordnetes MDI-Fenster.
fsStayOnTop	Das Formular bleibt immer im Vordergrund, es sei denn, bei anderen Fenstern ist FormStyle auch auf fsStayOnTop gesetzt. Wenn ein fsStayOnTop-Formular ein anderes fsStayOnTop-Formular aufruft, bleibt keines der beiden dauerhaft im Vordergrund. Ist das Formular das Hauptfenster einer MDI-Anwendung, muß FormStyle auf fsMDIForm gesetzt werden.

Visible: Sie können bestimmen, ob das Formular angezeigt wird. Hat Visible den Wert true, ist das Formular sichtbar (es sein denn, es wird von anderen Fenstern verdeckt). Soll es nicht angezeigt werden, setzen Sie Visible auf false. Die Methoden Show und ShowModal setzen Visible auf true und zeigen das Formular vor den anderen Fenstern an.

WindowState: Sie können festlegen und ermitteln, ob das Formular normal, minimiert (als Symbol) oder maximiert (Vollbild) geöffnet wird.

Canvas: Canvas stellt ein TCanvas-Objekt bereit, damit das Bitmap gezeichnet werden kann. Modifikationen in der Zeichenfläche wirken sich auf das zugrundeliegende Bitmap aus. Alle Zeichenflächenoperationen (Linien, Rechtecke, Kreise) sind für ein Bitmap zulässig - nicht nur Draw und StretchDraw. Das Bitmap-Objekt wird dabei als Parameter an diese Methoden übergeben. Die Zeichenfläche wird für das Bitmap automatisch erzeugt. Die Eigenschaft kann nur gelesen werden.

Methoden

BringToFront(): Sie können das Fenster ermitteln, das Eigentum von MainForm ist und zuletzt aktiv war, und es in den Vordergrund bringen. Vorher überprüft BringToFront auch, ob das Fenster sichtbar und aktiv ist.

Close(): Schließt eine Datei.

CloseQuery(): Sie können feststellen, ob das Formular geschlossen werden kann. Ist das der Fall, gibt CloseQuery den Wert true zurück, andernfalls false.

SetFocus(): Kann dem Formular der Eingabefokus übergeben werden. SetFocus ruft die Methode SetFocus des aktiven Steuerelements (falls vorhanden) auf. Mit der Eigenschaft Active können Sie feststellen, ob das Formular aktiv ist.

Show(): Sie können ein zuvor ausgeblendetes Steuerelement wieder sichtbar machen. Show setzt die Eigenschaft Visible des Steuerelements auf true und stellt sicher, daß auch das übergeordnete Objekt sichtbar ist.

ShowModal(): Kann ein Formular modal angezeigt werden. Wenn ein Formular modal angezeigt wird, kann die Anwendung erst fortgesetzt werden, wenn das Formular geschlossen wurde. Um ein modales Formular zu schließen, setzen Sie seine Eigenschaft ModalResult auf einen Wert ungleich null. ShowModal gibt den Wert zurück, auf den die Eigenschaft ModalResult gesetzt wird, wenn das Formular geschlossen wird.

Ereignisse

=====

OnActivate: Sie können bestimmte Operationen durchführen, sobald das Formular den Fokus erhält (wenn es beispielsweise angeklickt wird). Bei untergeordneten MDI-Fenstern (FormStyle = fsMDIChild) tritt OnActivate nur auf, wenn der Fokus von einem untergeordneten Fenster zu einem anderen wechselt. Wird der Fokus von einem untergeordneten Nicht-MDI-Fenster zu einem untergeordneten MDI-Fenster verlagert, wird das Ereignis **OnActivate** des übergeordneten MDI-Formulars ausgelöst.

OnClose: Sie können spezielle Verarbeitungen durchführen, wenn das Formular geschlossen wird. Sie können in der zugehörigen Ereignisbehandlungsroutine beispielsweise prüfen, ob in allen Feldern eines Dateneingabeformulars gültige Eingaben vorgenommen wurden. Ein Formular wird geschlossen, nachdem die Methode Close aufgerufen oder im Systemmenü die Option Schließen gewählt wurde. Der Parameter Action bestimmt, ob das Formular tatsächlich geschlossen wird. Er kann die folgenden Werte annehmen:

Wert	Bedeutung
caNone	Das Formular darf nicht geschlossen werden, daher geschieht nichts.
caHide	Das Formular wird nicht geschlossen, sondern lediglich verborgen. Die Anwendung kann nach wie vor auf das Formular zugreifen.
caFree	Das Formular wird geschlossen und sein Speicher freigegeben.
caMinimize	Das Formular wird nicht geschlossen, sondern als Symbol angezeigt. Bei untergeordneten MDI-Formularen ist dies die Standardeinstellung.

OnCloseQuery: Sie können festlegen, unter welchen Bedingungen das Formular geschlossen wird. Der Boolesche Parameter CanClose der Ereignisbehandlungsroutine bestimmt, ob das Formular geschlossen werden darf (Standardwert true). In der Ereignisbehandlungsroutine für

OnCloseQuery können Sie den Benutzer fragen, ob das Formular wirklich geschlossen werden soll. Der Benutzer kann beispielsweise in einem Meldungsfenster aufgefordert werden, vor dem Schließen des Formulars eine Datei zu speichern.

OnCreate: Es können spezielle Verarbeitungen durchgeführt werden, wenn das Formular erzeugt und vom Konstruktor von TCustomForm aufgerufen wird. Implementieren Sie entweder dieses Ereignis, oder überschreiben Sie den Konstruktor des Formulars; führen Sie jedoch nicht beides aus. Wenn ein Formular erzeugt wird und seine Eigenschaft Visible auf true gesetzt ist, werden die folgenden Ereignisse in der angegebenen Reihenfolge ausgelöst:

1. OnCreate
2. OnShow
3. OnActivate
4. OnPaint

OnShow: Sie können spezielle Verarbeitungen durchführen, wenn das Formular angezeigt (seine Eigenschaft Visible auf true gesetzt) wird.