

CodeRage® X

Pura Escovação de Bits com Delphi

Criando Controles VCL Compostos

Fernando Rizzato

Lead Software Consultant, Latin America

Agenda

- Breve Introdução a Controles Compostos
- Preview
 - Componente Exemplo
- Técnicas de Criação
 - Gerenciando Componentes Embutidos
- Provendo Acesso a Componentes Embutidos
 - Propriedades Agregadas
 - Sub-Componentes
- Criando Controles Compostos Visualmente

Controles Compostos

- Algumas vezes chamados de “compound controls” ou “user controls”
- Trata-se de um controle “container” que contém um ou mais controles incorporados
 - Não é o mesmo que um componente referenciado
- Baseado na mesma técnica utilizada para criar controles dinamicamente em tempo de execução

Benefícios de Controles Compostos

- Reusabilidade
- Consistência
- Performance (RAD)

Criando Controles Compostos

- Definir o container
- Criar os controles embutidos
- Disponibilizar propriedades internas
- Tratar eventos internos
- Disponibilizar eventos internos

Definindo o Container

- Deve ter um “window handle”
- Normalmente devem descender de
 - TWinControl
 - TCustomControl
 - TCustomPanel
- O Container é responsável por criar e destruir os controles embutidos

Criando Componentes Embutidos

- Declare um “field” para cada controle
- Crie os componentes no “constructor”
 - Especifique o container (Self) como “owner”
 - `lblFirstName := TLabel.Create(Self);`
- Defina a propriedade “parent”
- Defina valores iniciais para as propriedades do componente embutido
 - Diretamente no “constructor”
 - Pode ser deferido para o método “CreateWnd”

Disponibilizando Propriedades Internas

- Propriedades Empacotadas (Wrappers)
 - Utiliza o componente embutido para manter as propriedades
 - Deve prover métodos para leitura e escrita
- Propriedades Agregadas (Aggregate)
 - Propriedades disponibilizadas via *Object Inspector* de maneira agrupada

Exemplo #1 – Controle Composto

Sub-Componentes

- Método alternativo para disponibilizar propriedades internas
- Um sub-componente pode ser qualquer componente com persistência
- Todas as propriedades “published” do sub-componente aparecem no Object Inspector
- Read Only
 - TLabelledEdit.EditLabel property
- Read/Write
 - THTTPIO.Converter property

Sub-Componentes - Read Only

- O componente composto deve ser o Owner do sub-componente
- O componente composto deve definir uma propriedade para o sub-componente
- Deve chamar o método SetSubComponent
 - Chamado pelo componente composto
 - Ou, chamado pelo construtor do sub-componente

Sub-Componentes – Read/Write

- Método “write” da propriedade do sub-componente deve ser responsável pelo “free” quando outro componente for atribuído
- Por convenção, o sub-componente interno deve ser sempre recriado
- Deve chamar “FreeNotification” após o “SetupSubComponent”

Exemplo #2 – Sub-Componentes

Criação Visual de Controles Compostos?

- A maior dificuldade com controles compostos é sua criação manual
- A criação de controles compostos normalmente exige algum código que os desenvolvedores tentam evitar ao máximo
 - Criar e inicializar controles
 - Posicionar os controles via código
 - Configura e tratar eventos e propriedades
- Entretanto, o Delphi tem o Frame Designer!

Frames: Prós e Contras

- Vantagens
 - Frames são objetos de verdade
 - Frames podem utilizar herança
 - Alterações feitas nos frames são automaticamente propagadas
- Desvantagens
 - Os componentes embutidos são selecionáveis
 - Usuários podem modificar os componentes embutidos
 - Propriedades customizadas não aparecem no *Object Inspector*
 - Um tanto difícil de distribuir em uma equipe

Frame/Componente

- Conversão de um Frame em um Controle Composto
- Elimina as desvantagens da utilização tradicional de frames
 - Componentes embutidos não são selecionáveis
 - Desenvolvedor controla qual componente é passível de modificações
 - Propriedades customizadas aparecem no *Object Inspector*

Outros Benefícios

- Permite visualmente criar o controle composto
- Ligação de propriedades e eventos simplificada
- Permite visualmente editar o controle composto
- *Deployment* como qualquer outro componente

Criando um Frame/Componente

- Crie um novo *frame* normalmente
- Adicione a unit do *frame* a uma package de runtime
- Registre o componente com a IDE
- Carregue a package de design no Delphi
- Utilize o controle composto como qualquer outro componente visual

Exemplo #3 – Frame/Componente

Obrigado!

- Contatos

- Fernando Rizzato
- Fernando.Rizzato@embarcadero.com
- @FernandoRizzato

Embarcadero

www.embarcadero.com

@EmbarcaderoTech

- Perguntas e Respostas