

Capítulo 5

Utilizando Forms

Ao final deste capítulo, você estará apto a :

- Adicionar um form do template ao projeto
- Gravar um form como um template
- Definir a ordem de tab
- Adicionar um menu ao form

Utilizando Forms

Overview

No Delphi forms são pontos focais da aplicação que você está desenvolvendo. Um form em branco é bem parecido com uma tela de pintura onde você pode adicionar componente, desenvolvendo a interface de usuário. Este capítulo cobre tópicos como a utilização de templates de forms, quadros de dialogo e menus.

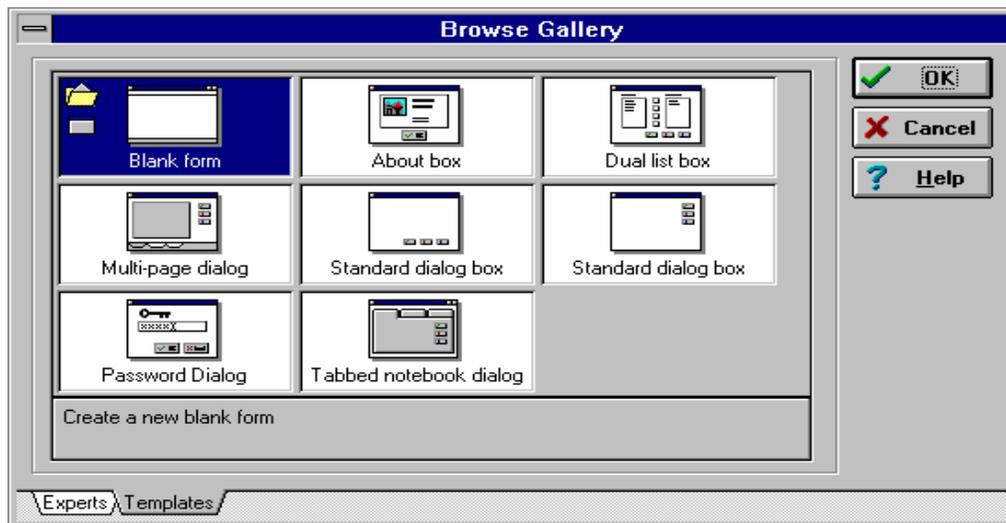
Utilizando um Template de Form

Introdução

Quando estiver construindo a interface de usuário de sua aplicação, você pode acelerar o desenvolvimento através da seleção de diversos templates de forms fornecidos com o Delphi. Um template é um form predefinido. Os templates incluídos com o Delphi fornecem um modelo para forms utilizados com maior frequência, por exemplo, um quadro de dialogo padrão o um quadro Sobre que fornece informações sobre uma aplicação. Os templates de form disponíveis são exibidos na Forms Gallery.

Adicionando um Template de Form a um Projeto

Para utilizar os templates na Forms Gallery, habilite a opção Gallery no quadro de dialogo Environment Options. Para acessar este quadro, no menu **Options**, selecione **Environment** e depois a página Preferência. Quando a opção **Use on New Project** estiver habilitada, o quadro de dialogo Forms Gallery aparece quando você selecionar **New Form** no menu **File**. O quadro de dialogo Forms Gallery aparece, como segue:



Passos para Adicionar um Template a um Projeto

Para adicionar um template de form ao projeto, primeiro você deve ter um projeto aberto. Uma vez aberto o projeto, execute os passos a seguir para adicionar um template ao projeto:

Passo	Ação
1	No menu File, selecione New Form . O quadro de dialogo Browse Gallery aparece.
2	Na página Template, selecione o template apropriado.
3	De um clique em OK. Uma cópia do form e arquivos unit associados são adicionados ao seu project.

Tutorial: Adicionando um Quadro Sobre...

Introdução

Neste Tutorial, você utilizará a Forms Gallery para adicionar um template de form ao seu projeto. A Forms Gallery contém diversos templates de form para os tipos mais comuns. Um template utilizado na maioria das aplicações é o quadro Sobre... O quadro Sobre... é uma característica padrão que os desenvolvedores adicionam a maioria dos programas Windows.

Estágios do Tutorial

Este tutorial envolve os seguintes estágios:

Estágio	Processo
1	Adicione um form Sobre...utilizando o template About.
2	Modificar o design do form.
3	Exibir o form Sobre...

Os templates de form efetuam o primeiro estágio e parte do segundo por você.

Passos para o Estágio 1

Execute os passos a seguir para adicionar um quadro Sobre utilizando o template de form AboutBox.

Passo	Ação
1	Abra um novo projeto e grave o arquivo de unit como UABOUT.PAS e o arquivo de projeto como PABOUT.DPR.
2	No menu File , selecione New Form . O quadro de dialogo Browse Gallery aparece. Se a Browse Gallery não aparecer, você seleciona no menu Options, Environment . Localize a página Preference, e no grupo Gallery, habilite Use on New Form .
3	Selecione o template About Box na Gallery, e de clique em OK . Uma nova unit e um novo form são adicionados ao projeto.

Passos para o Estágio 2

Execute os passos a seguir para modificar o design do form AboutBox:

Passo	Ação
1	Utilize os valores a seguir para a propriedade Caption dos componentes Label no Object Inspector.

Componente	Propriedade	Valor
------------	-------------	-------

ProductName	Caption	Exemplo de Quadro Sobre a Aplicação
Version	Caption	Versão 1.0
CopyRight	Caption	Copyright 1996
Comments	Caption	Tutorial de Quadro Sobre Aplicação

Passos para o Estágio 3

Para exibir o form, o primeiro passo é adicionar o identificador da unit do quadro Sobre... à cláusula uses da unit que irá exibir o quadro. Depois, utilize o método ShowModal ou Show em um event handler para exibir o form.

Passo	Ação
1	Adicione o identificador da Unit, Unit2, a cláusula uses da unit about, como mostrado na figura a seguir:

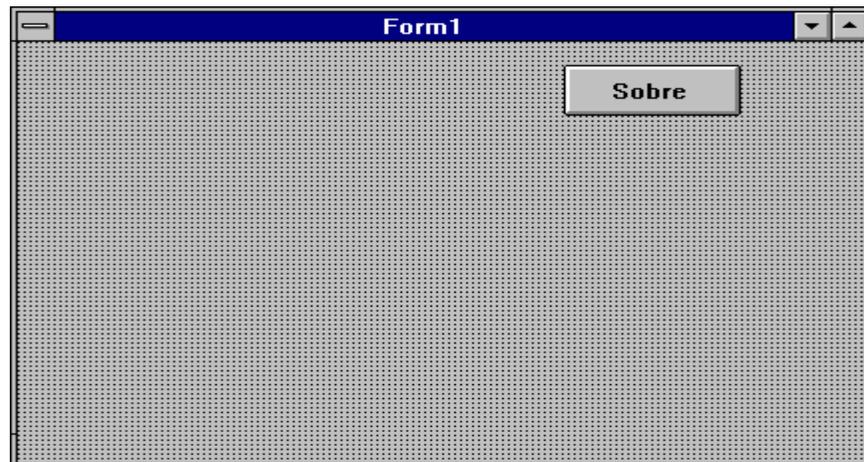
```

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Unit2;

type
  TForm1 = class (TForm)
    Button1: TButton;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
    
```

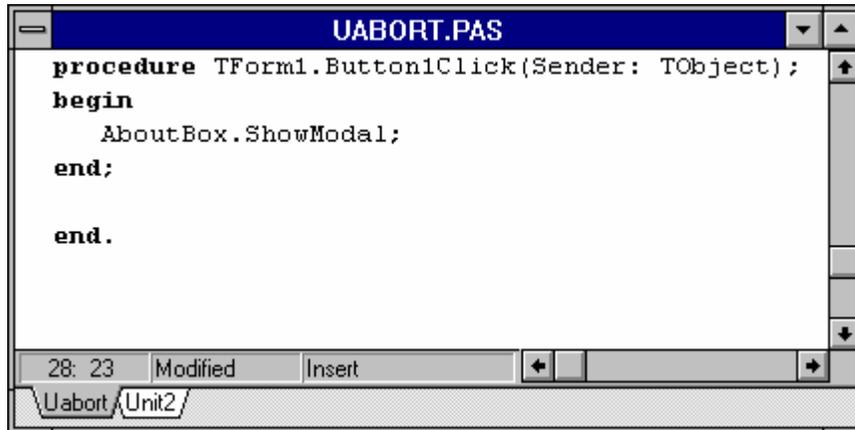
Passo	Ação
2	Adicione um botão ao form principal (Form1). Rotule este botão, como Sobre como na figura a seguir.



Passo	Ação
-------	------

Introdução ao Delphi

3	Crie um event handler para o evento OnClick do botão Sobre.
4	No event handler, digite a seguinte instrução: <code>AboutBox.ShowModal</code> ; O event handler parecerá com o seguinte:



```
procedure TForm1.Button1Click(Sender: TObject);
begin
    AboutBox.ShowModal;
end;

end.
```

The screenshot shows a code editor window with a blue title bar containing the text 'UABORT.PAS'. The code is written in Pascal and defines a procedure for the OnClick event of a button. The procedure body contains a single line: `AboutBox.ShowModal;`. The code is enclosed in `begin` and `end;` blocks, with a final `end.` for the unit. The status bar at the bottom of the window shows '28: 23 Modified Insert' and a tab labeled 'Uabort / Unit2'.

Passo	Ação
3	Compile e grave sua aplicação
4	Execute e teste a aplicação

Gravando um Form como Template

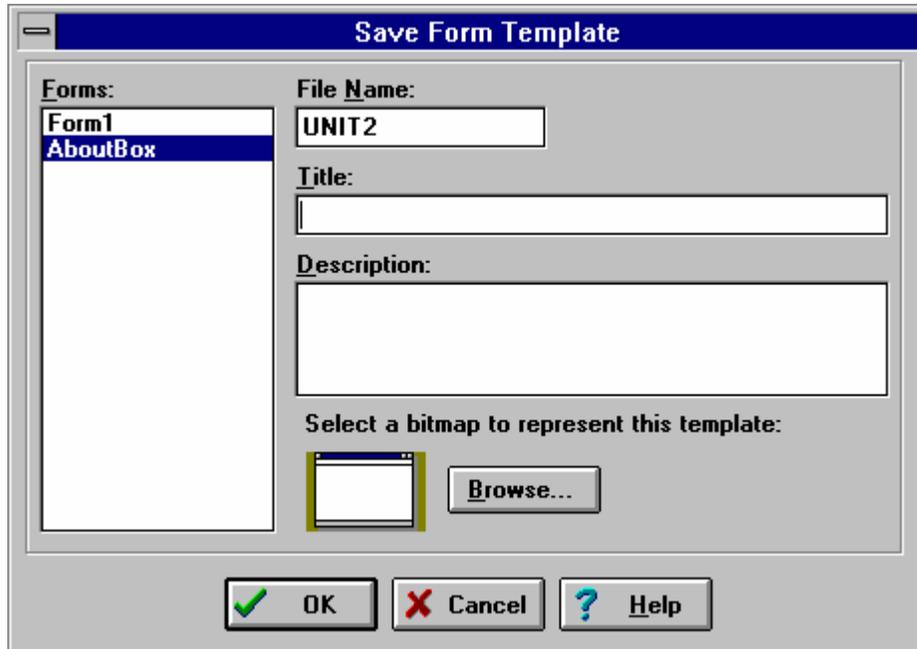
Introdução

Você pode gravar um form feito por você como um template. Estes templates estão disponíveis para serem utilizados em outros projetos. Gravar um form como um template insere o form no quadro de diálogo Browse Gallery. Para gravar um form como template, utilize o SpeedMenu do form e selecione Save As template. De um nome, descrição e icone ao novo template. Se você quiser especificar um icone para o template, você pode selecionar um bitmap existente ou criar um próprio.

Passos para Gravar um Form como um Template

Execute os passos a seguir para gravar um form como um template:

Passo	Ação
1	Dê um clique com o botão direito do mouse sobre o form para acessar o SpeedMenu.
2	Selecione Save As Template . O quadro de diálogo Save Form Template aparece, como segue:
3	No campo Title , digite um nome para o novo template.
4	No campo Description , digite uma descrição significativa do template.
5	Para associar um icone ao template, de um clique no botão Browse .
6	Localize e selecione o bitmap para associa-lo ao template, e de um clique no OK .
7	No quadro de diálogo Save As Form Template, de um clique em OK para gravar a informação e saia.



Criando Quadros de Dialogo

Introdução

O Delphi permite criar um número ilimitado de quadros de dialogo, do mais simples ao mais complexo. Este tópico cobre a criação de quadros de dialogo simples, tais como quadros de mensagem e de entrada de dados, bem como criar um quadro de dialogo com um form em branco.

Quadro de Mensagem

Um quadro de mensagem é uma janela que exibe text. Você pode utilizá-la para:

- Exibir informação
- Exibir uma mensagem de erro ou aviso
- Solicitar a confirmação de uma ação

Por exemplo, se o usuário tentar a aplicação antes de gravá-la, um quadro de mensagem deveria aparecer solicitando informando o usuário que os dados devem ser gravados antes de fechar. Para que a aplicação continue, o usuário deve remover o quadro de mensagem da tela manualmente, geralmente com o clique de um botão.

Para exibir um quadro de mensagem simples, utilize a função `MessageDlg`. O exemplo a seguir mostra a sintaxe da função:

```
Function MessageDlg (const Msg: String; AType: TMsgDlgType;
  AButtons: TMsgDlgButtons; HelpCtx: LongInt ): Word;
```

Explicação dos Parâmetros de MessageDlg

A tabela a seguir explica os parâmetros utilizados na sintaxe da função `MessageDlg`:

Parâmetro	Explicação
-----------	------------

Msg	A mensagem que você quer exibir (uma constante string)
AType	Determina o tipo de quadro de mensagem que ira aparecer. Os valores possíveis são: <ul style="list-style-type: none">• <i>mtWarning</i> Exibe um quadro de mensagem com um ponto de exclamação em amarelo(!)• <i>mtError</i> Exibe um quadro de mensagem com uma não vermelha• <i>mtInformation</i> Exibe um quadro de mensagem com um símbolo de informação em azul (i)• <i>mtConfirmation</i> Exibe um quadro de dialogo com um ponto de interrogação em verde (?)
AButtons	Determina quais botões aparecerão no quadro de mensagem. AButtons é do tipo TMsgDlgBtns que é um tipo set. Você pode incluir diversos botões dentro do set. O set pode incluir os seguintes valores: <ul style="list-style-type: none">• <i>mbYes</i> Exibe um botão com um check mark em verde e o caption Yes• <i>mbNo</i> Exibe um botão com um circulo e uma barra por dentro, em vermelho e o caption No• <i>mbOK</i> Exibe um botão com um check mark em verde e o caption OK• <i>mbCancel</i> Exibe um botão com um X em vermelho e o caption Cancel• <i>mbHelp</i> Exibe um botão com um ponto de interrogação e o caption Help• <i>mbYesNoCancel</i> Exibe um conjunto que inclui os botões Yes, No, Cancel• <i>mbOKCancel</i> Exibe um conjunto que inclui os botões OK e Cancel
HelpCtx	Determina qual tela de Help está disponível para o quadro de mensagem. Um número de contexto para um Help sensível ao contexto.

Valores de Retorno de MessageDlg

Os valores de retorno para a função MessageDlg indicam o tipo de botões clicados pelo usuário. O valor de retorno pode ser um dos seguintes:

- *mrNone*
- *mrOK*
- *mrCancel*
- *mrAbort*
- *mrRetry*
- *mrIgnore*
- *mrYes*
- *mrNo*
- *mrAll*

Exemplos de Código Utilizando MessageDlg

O código de exemplo a seguir ilustra o uso da função MessageDlg:

```
Res := MessageDlg( ' Sair da Aplicação ??' , mtConfirmation, [ mbOk, mbCancel ], 0 );  
If Res = mbOk Then  
  Close;
```

Quadros Simples de Entrada de Dados

Um quadro simples de entrada de dados requer uma linha de informação do usuário. Um exemplo de um destes quadros é um que solicite uma senha. Para exibir um quadro de entrada de dados, utilize as funções InputBox ou InputQuery. A sintaxe para cada função aparece, como segue:

```
function InputBox(const ACaption, APrompt, ADefault: string): string;  
function InputQuery(const ACaption, APrompt: string; var Value: string): Boolean;
```

Explicação dos Parâmetros de InputBox e InputQuery

A tabela a seguir explica os parâmetros das funções InputBox e InputQuery:

Parâmetros	Explicação
ACaption	Determina o caption do quadro de diálogo
APrompt	Determina o texto de solicitação ao usuário
ADefault	Determina a string exibida quando o quadro aparecer
Value	Determina a string exibida quando o quadro aparecer, contém a string digitada pelo usuário quando OK for clicado. Utilizado por InputQuery.

Valores de Retorno de InputBox

A função InputBox exibe um quadro de entrada e aguarda que o usuário digite um valor. O valor de retorno de InputBox é uma string. Utilize a função InputBox quando não importa se o usuário escolher entre o botão OK ou Cancelar para deixar o quadro de diálogo. A função InputQuery exibe um quadro de entrada e aguarda que o usuário digite um valor. InputQuery retorna True se o usuário escolher OK e False se escolher **Cancelar**.

Forms MDI e SDI

Uma aplicação Multiple Document Interface (MDI) é uma aplicação com uma interface de janela principal, ou janela pai, que permite a abertura de diversos documentos, ou janelas filhas.

Uma aplicação Single Document Interface (SDI) pode processar mais de um documento dentro de uma janela pai.

Você pode implementar qualquer um de seus forms como MDI ou SDI. Embora você possa criar um form MDI ou SDI alterando a propriedade FormStyle de um form existente, é mais provável que você crie um form MDI ou SDI utilizando a Gallery.

Propriedades dos Quadros de Diálogo

Quando você cria um form no Delphi, ele possui as seguintes características:

- Botões de Maximizar e Minimizar
- Menu de Controle
- Borda de Reajuste

Estas características são úteis para quadros de diálogo não-modais, mas desnecessárias para quadros modais. Um quadro de diálogo não-modal é um quadro de onde o usuário pode remover o foco sem fecha-lo. Um quadro de diálogo modal requer que o usuário feche o quadro antes de continuar o trabalho na aplicação. Você pode definir a propriedade BorderStyle para bsDialog. Esta definição possui as seguintes características:

- Remover os botões de Maximizar e Minimizar
- Fornecer um menu Controle com as opções **Mover** e **Fechar**
- Tornar a borda do form não-reajustavel com aparência tridimensional

Adicionando Botões de Comando

Se seu quadro de diálogo será utilizado em estado modal, você deve oferecer botões de comando dentro do quadro. Por exemplo, você deve oferecer um botão Sair que permite que o usuário saia do quadro de diálogo. Os botões de comando mais comuns incluem os seguintes:

Introdução ao Delphi

- Um botão Cancelar ou Não que saia do quadro de diálogo sem gravar as alterações
- Um botão OK ou Sim que grave as alterações e depois saia do quadro de diálogo
- Um botão que tenha o foco por default para que o usuário possa pressionar a tecla Enter para aceitar o botão default

Você pode utilizar o componente BitBtn da página Additional para criar vários destes botões de comando. O componente BitBtn permite que você utilize os bitmaps padrões da Borland, tais como um check mark em verde () para um botão OK. Você pode selecionar o bitmap e caption padrão que aparecem no botão definindo a propriedade Kind.

Definindo a Ordem de Tab

Em uma aplicação sendo executada, a ordem de Tab é a seqüência no qual o usuário pode mover-se entre os componentes pressionando a tecla Tab. Para permitir que os usuários utilizem a tecla Tab para mover-se pelos componentes, você deve definir a propriedade TabStop de cada componente para True. Por default, a ordem de Tab é definida de acordo com a ordem na qual você inseriu os componentes no form. Você pode alterar a ordem de Tab default alterando a propriedade TabOrder do componente no Object Inspector ou utilizar o quadro de diálogo Edit Tab Order.

Passos para Definir a Ordem de Tab

Execute os passos a seguir para definir a ordem de Tab utilizando o quadro de diálogo Tab Order:

Passo	Ação
1	Selecione o form contendo os componentes cuja ordem de tab você queira definir.
2	No menu Edit , selecione Tab Order. O quadro de dialogo Edit Tab Order aparece, listando os componentes na ordem atual, como no exemplo a seguir:
3	No campo Controls listed in tab order , selecione o controle (componente).
4	Utilize os botões com as setas para cima e para baixo para reordenar a ordem dos componentes. Você também pode arrastar os componentes.
5	Quando tiver terminado, de um clique em OK .



Definindo o Foco em Quadro de Diálogo Durante o Design

Em um quadro de diálogo, somente um componente por vez pode ter o foco. O foco pode ser definido durante o design ou durante a execução. Durante o design, você pode definir a propriedade ActiveControl do quadro de diálogo abrir. Se você não especificar a propriedade ActiveControl, o primeiro componente da ordem de Tab receberá o foco. Entretanto, isto não se aplica nas seguintes situações:

- O componente está desabilitado

- O componente não é visível durante a execução
- A propriedade TabStop do componente está definida para **False**.

Definindo o Foco em um Quadro de Diálogo Durante a Execução

Durante a execução, o usuário pode alterar o foco dos componentes automaticamente, utilizando a tecla Tab. Entretanto, você pode querer especificar que o foco seja alterado de campo para campo utilizando as teclas de seta. Você pode utilizar o método `SetFocus` para especificar quais componentes recebem o foco. Para alterar o componente ativo durante a execução, digite o seguinte código no event handler apropriado:

```
<componente>.SetFocus;
```

Por exemplo, a instrução a seguir especifica que `Button2` é o componente ativo durante a execução:

```
Button2.SetFocus;
```

Criando Menus

Introdução

Em uma aplicação, os menus oferecem uma maneira dos usuários executarem comandos. O Delphi fornece dois tipos de componentes menu: `MainMenu` e `PopupMenu`. Durante o processo de especificação de um menu, você utiliza um recurso dos componentes `TMainMenu` e `TPopupMenu` chamado `Menu Designer`. O `Menu Designer` permite criar um menu ou adicionar um menu já pronto ao seu form. Você também pode utilizar o `Menu Designer` para deletar, editar ou reordenar itens do menu.

Quando utilizar o `Menu Designer` para criar um menu, você pode visualizar seu menu como ele apareceria na execução sem ter que realmente executá-la.

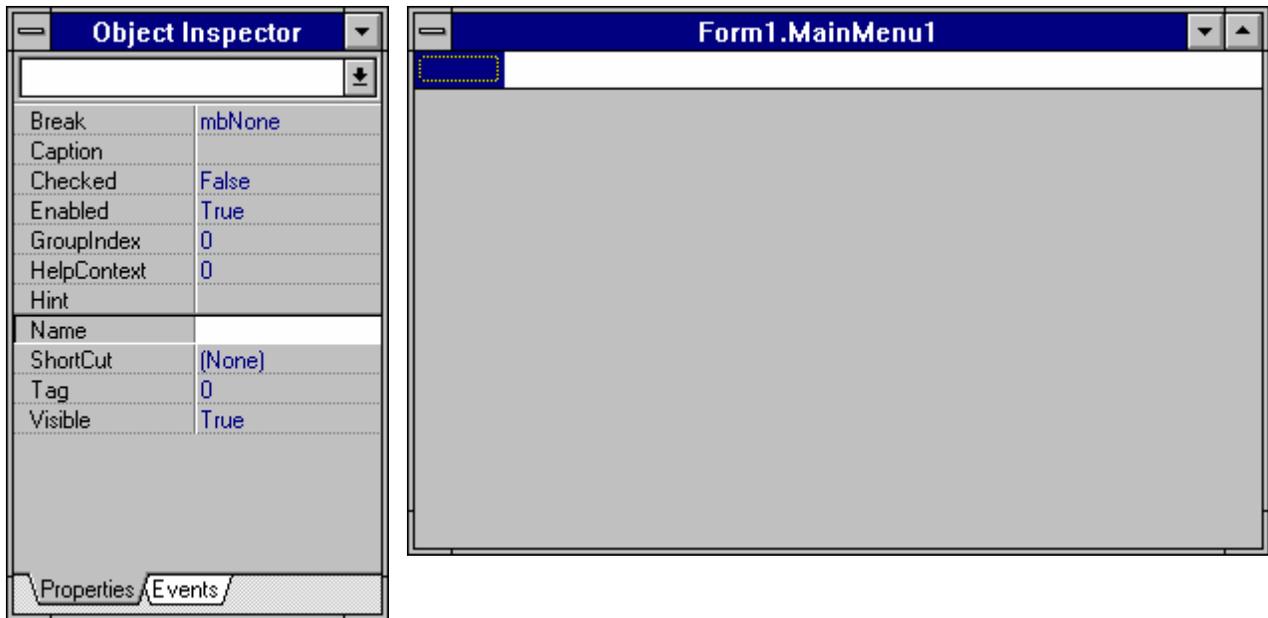
Adicionando um Menu

Para adicionar um menu a um form, utilize o `Menu Designer`. O `Menu Designer` permite definir um menu ou adicionar um menu personalizado ao seu form.

Passos para Adicionar um Menu

Execute os passos a seguir para adicionar um menu ao seu form durante o design:

Passo	Ação
1	Na página Standard da Component Palette, dê um duplo-clique no componente <code>MainMenu</code> ou <code>PopupMenu</code> . O componente é adicionado ao seu form.
2	Dê um duplo-clique no componente menu para abrir o <code>Menu Designer</code> . O <code>Menu Designer</code> e o <code>Object Inspector</code> aparecem, como segue. A propriedade <code>Name</code> no <code>Object Inspector</code> é selecionada.



Passo	Ação
3	Digite o texto para o primeiro item da barra de menus na propriedade Caption do Object Inspector.
4	Pressione Enter . O primeiro item da barra de menu aparece na barra do form.
5	Para adicionar itens ao menu, digite o primeiro item na propriedade Caption.
6	Pressione Enter . O nome do item de menu aparece endentado abaixo do item da barra de menus.
7	Repita os passos 5 e 6 ate que você tenha adicionado todos os itens de menu que você queira para seu primeiro menu.
8	Para inserir o próximo item da barra de menus, de um clique na barra de menus e repita os passos 3 e 4.
9	Adicione qualquer item ao segundo menu repetindo os passos 5 e 6.

Adicionando Barras Separadoras

Adicionar uma barra separadora insere uma linha entre itens de menu ou grupos de itens. Você pode utilizar barras separadoras para agrupar os itens logicamente ou criar uma quebra visual na lista de itens. Para criar uma barra separadora, digite um hífen (-) como propriedade Caption para este item. Barras separadoras aparecem como segue:

Run	
Run	F9
Parameters...	
Step Over	F8
Trace Into	F7
Run to Cursor	F4
Show Execution Point	
Program Pause	
Program Reset	Ctrl+F2
Add Watch...	Ctrl+F5
Add Breakpoint...	
Evaluate/Modify...	Ctrl+F7

Habilitando Acesso aos Menus pelo Teclado

Você pode permitir que os usuários acessem menus e comandos de menu através do teclado, como segue:

- Teclas de atalho
- Teclas aceleradora

A figura anterior também mostra um exemplo de um menu com teclas aceleradoras e teclas de atalho.

Adicionando Teclas Aceleradoras

Teclas aceleradoras permitem que o usuário acesse menus e comandos do menu utilizando a tecla **Alt** juntamente com a letra apropriada. Por exemplo, para acessar o menu File, o usuário pressionaria **Alt+F**. Você pode adicionar teclas aceleradoras ao seu menu precedendo a letra selecionada com o símbolo **&**, quando estiver digitando o caption do item. Por exemplo, para adicionar um menu Arquivo com a letra A como tecla aceleradora, digite *&Arquivo* como caption do item de menu.

Adicionando Teclas de Atalho

Teclas de atalho permitem que o usuário acesse comandos do menu utilizando o teclado para ignorar os menus. Por exemplo, você pode designar F1 como tecla para o comando Help. Você pode especificar atalhos digitando um valor na propriedade ShortCut ou selecionando na lista da propriedade ShortCut.



O Delphi não faz a checagem para assegurar que teclas aceleradoras ou atalhos não estão duplicados em sua aplicação. Isto é responsabilidade do desenvolvedor.

Propriedades Adicionais de Menu

Você pode definir as seguintes propriedades de menu para alterar a aparência dos menus em sua aplicação.

- **Checked**
Definindo esta propriedade para **True**, um check aparece próximo ao item.
- **Enabled**
Definindo esta propriedade para **False**, o item torna-se desabilitado, ou acinzentado, e o usuário não poderá acessá-lo ou a qualquer um de seus submenus.

Criando Menus Aninhados

Muitos menus contém menus aninhados, ou submenus, que fornecem ao usuário comandos adicionais para serem selecionados. Menus aninhados são indicados por uma seta próxima ao item na lista de itens. Você pode querer utilizar menus aninhados para economizar espaço vertical em sua aplicação. No Delphi, você pode criar tantos menus aninhados quantos forem necessários.

Utilizando o SpeeMenu do Menu Designer

O SpeedMenu do Menu Designer oferece acesso rápido aos comandos mais utilizados e opções de template de menu. Você pode exibir o SpeedMenu dando um clique com o botão direito do mouse na janela do Menu Designer ou pressionando **Alt+F10** enquanto o cursor estiver na janela do Menu Designer. O SpeedMenu aparece, como segue:

I nsert	I ns
D elete	D el
C reate Submenu	C trl+Right
S elect Menu...	
S ave As Template...	
I nsert From Template...	
D elete Templates...	
I nsert From Resource...	

Descrição das Opções do SpeedMenu

A tabela a seguir descreve as opções no SpeedMenu do Menu Designer:

Opção	Descrição
Insert	Adiciona uma localização de item de menu antes do cursor
Delete	Remove o item de menu selecionado
Create Menu	Insere um item de menu a direita do item selecionado, adiciona uma ponta de seta para indicar um nível de alinhamento de menu
Select Menu	Permite selecionar um menu existente
Save As Template	Permite gravar um menu para utilização posterior
Insert From Template	Permite adicionar um template de menu previamente criado ao menu atual
Delete Template	Permite remover menus criados previamente
Insert From Resource	Permite importar um menu de um arquivo de resource (.RC) do Windows

Associando Eventos de Menu com Código

Introdução

Uma vez criado um menu, você precisa associar ao evento OnClick de cada item.

Manipulando Eventos de itens de Menu

O evento OnClick é o único evento para itens de menu. O código associado com o evento OnClick dos itens dos menus serão executados quando o usuário selecionar o item.

Você pode criar um event handler para qualquer item de menu dando um duplo-clique no item, na janela do Menu Designer e adicionando o código dentro da instrução begin...end.

Associando um Item de Menu a um Manipulador de Menu Existente

Se você quiser reutilizar código, você pode associar um item de menu com um event handler já existente. Para isto, vá para a página Events do Object Inspector e abra a lista dos event handlers para o evento OnClick. Você pode selecionar qualquer um dos event handlers que aparecem na lista, e o código para aquele event handler estar associado com o item de menu selecionado.

Lab: Adicionando um Template de Form a sua Aplicação

Objetivo:

Este lab reforça sua habilidade em:

- Utilizar um Template de Form para adicionar um form
- Digitar código para exibir um form
- atribuir teclas de atalho aos menus

Cenário

Neste lab você utilizará a aplicação desenvolvida no lab Adicionando Componentes Standard a uma Aplicação para adicionar um quadro de dialogo Sobre... utilizando um template de form.

Processo

Utilize o processo a seguir para aplicar o que você aprendeu:

Estágio	Processo
1	Abra o projeto chamado PLAB5.DPR. Grave o arquivo unit como ULAB8.PAS e o projeto como PLAB8.DPR. Importante: Certifique-se de utilizar o comando Save File As , bem como Save Project As no menu File .
2	Abra o Menu Designer para MainMenu1. Adicione um separador e a palavra Sobre no final do menu File .
3	Utilize um template de form para adicionar um quadro Sobre a aplicação. Certifique-se de gravar este form com o nome UNIT2.PAS.
4	Adicione uma tecla de atalho ao menu para que o usuário possa pressionar F2 para exibir o quadro Sobre.
5	Adicione aceleradores para os itens Abrir e Gravar .
6	Adicione um event handler que exiba o form AboutBox.
7	Após compilar sua aplicação, grave-a. Execute e teste sua aplicação.

Resumo do capítulo

Pontos Chave

Após completar este capítulo, você aprendeu que:

- Templates fornecem uma “planta” para os forms mais utilizados, e eles compõem a Form Gallery.
- Você pode gravar uma cópia de um form por você como um template para ser utilizado posteriormente em outro projeto. Gravar um form como um template insere o form na lista de templates que você pode selecionar quando cria um novo form.
- Quando você cria um form, ele possui as seguintes características:
Botões de Maximizar e Minimizar
Menu Controle
Borda Reajustável
- Ordem de Tab é a seqüência na qual o usuário pode se mover de componente a componente pressionando a tecla **Tab**.
- Menus fornecem uma maneira de agrupar comandos. Você pode utilizar o Menu Designer para criar ou adicionar um menu predeterminado ao form.

Termos e Definições

A tabela a seguir é uma referência rápida aos termos explicados neste capítulo:

Termo	Definição
MDI	Multiple Document Interface, uma aplicação feita de uma janela principal, ou janela pai, que permite abrir diversos documentos ou janelas
Ordem de Tab	A seqüência na qual um usuário pode se mover de componente a componente pressionando a tecla Tab
Quadro de Diálogo Modal	Um quadro de dialogo onde o usuário deve fechar antes de continuar a utilizar a aplicação
Quadro de Diálogo Não-Modal	Um quadro de dialogo que o usuário pode manter aberto enquanto continua a trabalhar na aplicação
SDI	Single Document Interface, uma aplicação que pode processar mais que um documento. Mas os documentos estão contidos de uma janela pai.