O Visual FoxPro fornece uma verdadeira operação não-modal para que você possa coordenar com facilidade vários formulários automaticamente, assim como executar várias instâncias de um formulário ao mesmo tempo. O Visual FoxPro também gerencia o processamento de eventos para que você possa oferecer aos usuários um ambiente interativo muito mais rico.

Este capítulo aborda os seguintes tópicos:

- Eventos no Visual FoxPro
- Controlando seqüências de eventos
- Atribuindo código a eventos

Eventos no Visual FoxPro

Um código de evento é disparado automaticamente pelo sistema em resposta a alguma ação do usuário. Por exemplo, o código criado para o evento Click é processado automaticamente pelo sistema quando o usuário clica em um controle. O código de evento pode, também, ser disparado por eventos do sistema, como no caso do evento Timer em um controle de cronômetro.

Os eventos principais

GotFocus	O objeto recebe o foco, seja através de uma ação do usuário como tabulação ou clique do mouse, seja pela alteração do foco no código utilizando-se o método SetFocus.
LostFocus	O objeto perde o foco, seja através de uma ação do usuário como tabulação ou clique do mouse em outro objeto, seja pela alteração do foco no código utilizando-se o método SetFocus.
KeyPress	O usuário pressiona e libera uma tecla.
MouseDown	O usuário pressionaum botão do mouse enquanto o ponteiro do mouse permanece sobre o objeto.
MouseMove	O usuário move o mouse sobre o objeto.
MouseUp	O usuário liberaum botão do mouse enquanto o ponteiro do mouse está sobre o objeto.

Recipientes e eventos de objeto

Existem duas regras básicas que você deve ter em mente ao criar um código de evento para os controles:

- Os recipientes não processam eventos associados aos controles neles contidos.
- Se não houver um código de evento associado a um controle, o Visual FoxPro verificará se há um código associado ao evento superior na hierarquia de classes do controle.

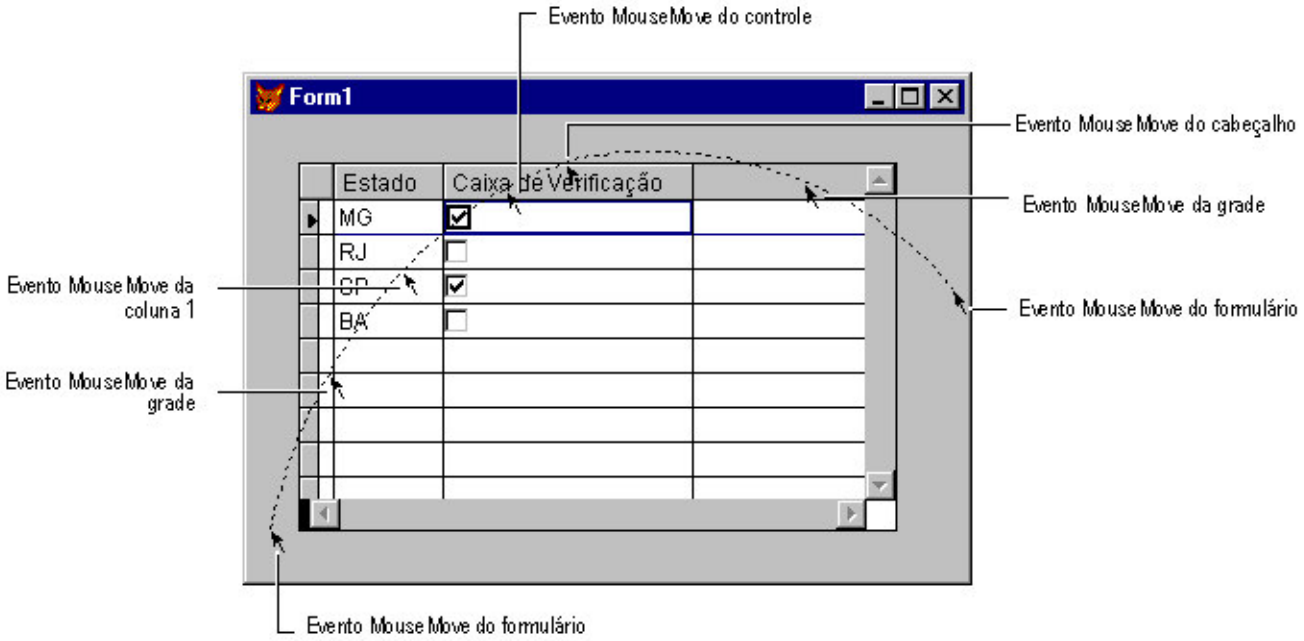
Os eventos de objeto ocorrem quando um usuário interage com um objeto de alguma fora - usando a tecla de tabulação, clicando o mouse, movendo com o ponteiro do mouse etc. Cada objeto recebe seus eventos independentemente. Por exemplo, mesmo que um botão de comando esteja em um formulário, o evento Click do formulário não é disparado quando o usuário clica no botão de comando. Nesse caso, apenas o evento Click do botão de comando é disparado.

O código de evento de recipiente é separado do código de evento de controle

Se não houver um código de evento Click associado ao botão de comando, nada acontecerá quando o usuário clicar no botão, mesmo que haja um código de evento Click associado ao formulário.

Esta regra também é verdadeira para os controles de grade. O Grid contém colunas que, por sua vez, contêm cabeçalhos e controles. Quando os eventos ocorrem, apenas o objeto mais interno envolvido no evento reconhece o evento. Os recipientes de nível superior não reconhecem o evento. A ilustração abaixo mostra quais os objetos que processam os eventos MouseMove gerados quando um usuário move o ponteiro do mouse através do Grid. Para uma versão animada, clique aqui.

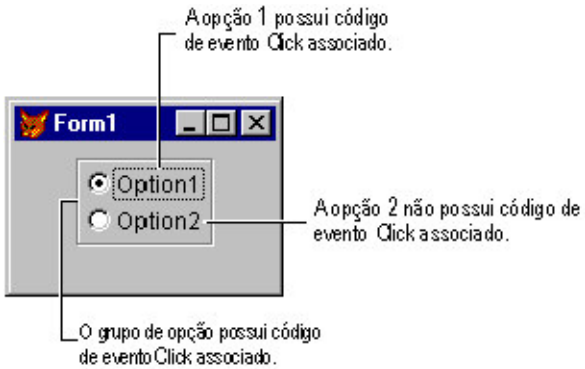
Eventos MouseMove de um Grid

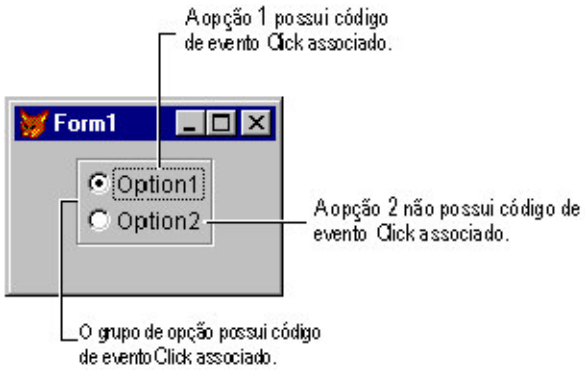


Entretanto, existe uma exceção a essa regra. Se você tiver escrito um código de evento para um grupo de botões de opção ou de botões de comando, mas não existir um código para o evento em um botão específico no grupo, o código de evento de grupo *será* executado quando o evento do botão ocorrer.

Por exemplo, você pode ter um grupo de botões de opção com um código de evento Click associado, mas apenas um dos dois botões de opção no grupo tem código de evento Click associado:

O código de evento para grupos de botões pode ser utilizado como padrão





Se um usuário clicar em Option1, o código de evento Click associado a Option1 será executado. O código de evento Click associado ao grupo de botões de opção não será executado.

Como não existe um código de evento Click associado ao botão **Option2**, se o usuário clicar no botão **Option2**, o código de evento Click do grupo de botões de opção será executado.

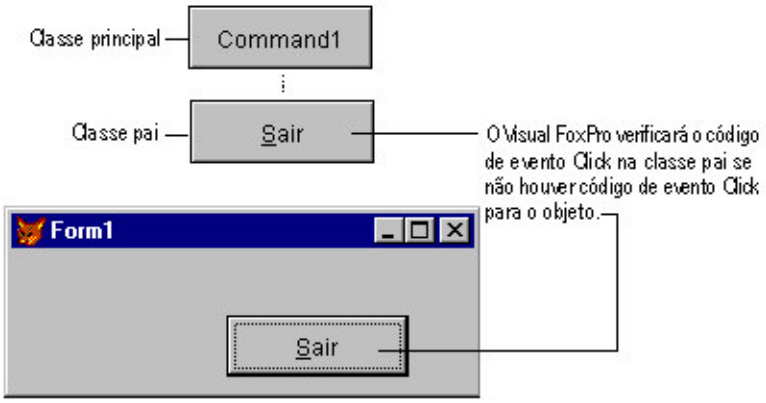
Observação Quando uma seqüência de eventos, como MouseDown seguido de MouseUp, for iniciada para um controle, a seqüência completa de eventos pertencerá ao controle.

Por exemplo, se você clicar o botão esquerdo do mouse em um botão de comando e arrastar o ponteiro do mouse para longe do botão desse comando, os eventos MouseMove do botão de comando continuarão ocorrendo, mesmo que o ponteiro do mouse esteja se movendo sobre o formulário. Se você liberar o botão esquerdo do mouse sobre o formulário e não sobre o botão de comando, o evento MouseUp que ocorrer será associado ao botão de comando em vez de ao formulário.

Classes e eventos de controle

Se um controle em um formulário for baseado em uma classe definida pelo usuário (que, por sua vez, poderia ser baseada em outra classe definida pelo usuário), o Visual FoxPro procurará no controle atual um código de evento quando este ocorrer. Se houver um código neste procedimento de evento, o Visual FoxPro o executará. Caso contrário, o Visual FoxPro verificará o nível superior na hierarquia de classes. Se, em qualquer ponto da hierarquia de classes, o Visual FoxPro encontrar um código para o evento, esse código será executado. Qualquer código que se encontre mais acima na hierarquia não será executado.

Se não houver um código associado a um objeto, o Visual FoxPro verificará a classe pai.



No entanto, você pode incluir um código em um procedimento de evento e chamar o código explicitamente nas classes nas quais o controle se baseia, utilizando a função DODEFAULT( ).

Controlando seqüências de eventos

O modelo de evento do Visual FoxPro é extensivo, permitindo que você tenha um grande controle sobre os componentes do aplicativo em resposta a uma ampla variedade de ações do usuário. Algumas das seqüências de eventos são fixas como, por exemplo, quando um formulário é criado ou destruído. Alguns eventos ocorrem de forma independente, mas a maioria ocorre em conjunção com outros vários eventos baseados na interação com o usuário.

Ativando o Controle de evento

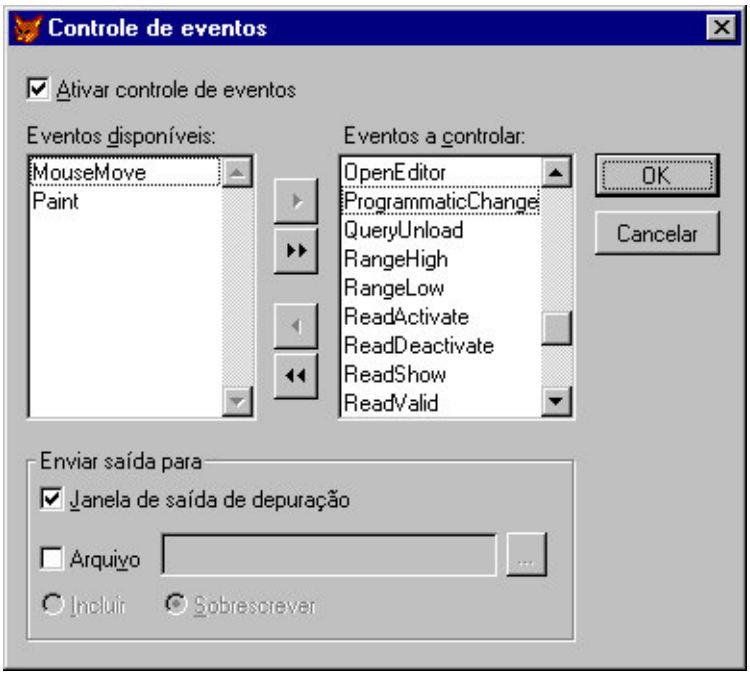
A melhor maneira de ver as seqüências de evento do Visual FoxPro é ativando o controle de evento no depurador. O controle de evento permite que você veja quando cada evento associado aos seus próprios formulários e controles ocorre em relação a outros eventos, para que possa determinar o local mais eficiente para a incluir o seu código.

Para ativar o Controle de eventos

1. No menu **Ferramentas**, na janela **Depurador**, escolha **Controle de eventos**.
2. Na caixa de diálogo **Controle de eventos**, selecione **Ativar controle de eventos**.

Os eventos nos Eventos para controlar lista são gravados na janela **Saída de depuração** ou em um arquivo, à medida que ocorrem.

A caixa de diálogo Controle de eventos



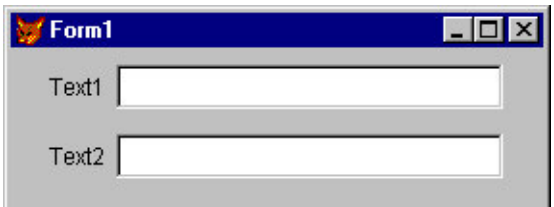
Observação Neste exemplo, os eventos MouseMove e Paint foram removidos dos Eventos para controlar lista porque ocorrem com tanta freqüência que acabam dificultando a visualização das seqüências de outros eventos.

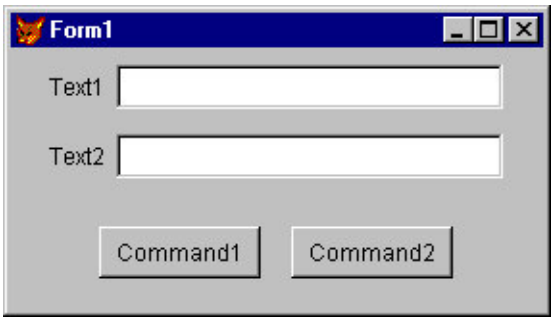
Assistindo a ocorrência de eventos

Algumas vezes, um evento único é disparado por uma ação do usuário, como a movimentação do ponteiro do mouse sobre um controle. Muitas vezes, no entanto, uma ação do usuário dispara vários eventos.

Esta seção descreve a ordem em que os eventos ocorrem em resposta à interação com o usuário, utilizando o formulário abaixo como exemplo.

Um formulário de exemplo para ilustrar seqüências de eventos





Neste exemplo, o usuário executa as seguintes ações no formulário:

1 Executa o formulário.

2 Digita o texto em Text1.

3 Seleciona o campo e o copia na área de transferência.

4 Move-se para Text2.

5 Cola o texto em Text2.

6 Fecha o formulário clicando no Command2.

Essas ações disparam um ou mais eventos de sistema para cada objeto. As tabelas a seguir detalham os eventos disparados em resposta a cada ação do usuário.

Ação 1

O usuário executa o formulário, digitando o comando abaixo na janela **Comando**:

```
DO FORM form1 NAME frmObject
```

O Visual FoxPro carrega o formulário, inicializa cada objeto e, em seguida, inicializa o formulário. O formulário é ativado e, em seguida, o primeiro campo recebe o foco de entrada.

Objeto	Evento
DataEnvironment	BeforeOpenTables
Form1	Load
DataEnvironment	Init
Text1	Init
Text2	Init
Command1	Init
Command2	Init
Form1	Init
Form1	Activate
Form1	GotFocus
Text1	When
Text1	GotFocus

Ação 2

O usuário digita **Teste** em Text1. A cada vez que uma tecla é pressionada, dois eventos são gerados. O evento KeyPress recebe dois parâmetros: a tecla pressionada e o estado das teclas SHIFT, ALT e CTRL.

Objeto	Evento
Text1	KeyPress (84, 1) “T”
Text1	InteractiveChange

Text1	KeyPress (101, 0) “e”
Text1	InteractiveChange
Text1	KeyPress (115,0) “s”
Text1	InteractiveChange
Text1	KeyPress (116,0) “t”
Text1	InteractiveChange

Ação 3

O usuário clica duas vezes em Text1 para selecionar o texto e, em seguida, pressiona CTRL+C para copiar o texto na área de transferência. Os eventos do mouse e um evento Click acompanham o evento DblClick. Os eventos MouseMove e MouseDown recebem quatro parâmetros: um número indicando que botão foi pressionado, o estado Shift e as localizações de X e Y. As localizações de X e Y são relativas ao formulário e refletem o modelo de escala (por exemplo, pixels) do formulário. Apenas um evento MouseMove é listado para cada controle. Na verdade, este evento provavelmente disparará meia dúzia de vezes ou mais.

Objeto	Evento
Form1	MouseMove(0, 0, 100, 35)
Text1	MouseMove(0,0,44,22)
Text1	MouseDown(1, 0, 44, 22)
Text1	MouseUp(1, 0, 44, 22)
Text1	Click
Text1	MouseDown(1, 0, 44, 22)
Text1	MouseUp(1, 0, 44, 22)
Text1	DblClick

Ação 4

O usuário pressiona a tecla TAB para mover-se para Text2.

Objeto	Evento
Text1	KeyPress(9, 0)
Text1	Valid
Text1	LostFocus
Text2	When
Text2	GotFocus

Ação 5

O usuário cola o texto copiado em Text2 pressionando a combinação de teclas CTRL+V.

Objeto	Evento
Text2	InteractiveChange

Ação 6

O usuário clica em Command2, que fecha o formulário.

Objeto	Evento
Form1	MouseMove
Command2	MouseMove
Text2	Valid
Command2	When

Text2	LostFocus
Command2	GotFocus
Command2	MouseDown(1, 0, 143, 128)
Command2	MouseUp(1, 0, 143, 128)
Command2	Click
Command2	Valid
Command2	When

Enquanto o formulário é fechado e o objeto é liberado, estes eventos adicionais ocorrem na ordem inversa em que são apresentados na Ação 1.

Objeto	Evento
Form1	Destroy
Command2	Destroy
Command1	Destroy
Text2	Destroy
Text1	Destroy
Form1	Unload
DataEnvironment	AfterCloseTables
DataEnvironment	Destroy

A seqüência de eventos do Visual FoxPro

A tabela a seguir mostra a seqüência de disparo geral dos eventos do Visual FoxPro. Presume-se que a propriedade AutoOpenTables do ambiente de dados seja definida com verdadeiro (.T.). Os outros eventos podem ocorrer com base em uma interação entre o usuário e resposta do sistema.

Objeto	Eventos
Data environment	BeforeOpenTables
Form set	Load
Form	Load
Data environment cursor(s)	Init
Data environment	Init
Objects [1]	Init
Form	Init
Form set	Init
Form set	Activate
Form	Activate
Object1 [2]	When
Form	GotFocus
Object1	GotFocus
Object1	Message
Object1	Valid [3]
Object1	LostFocus
Object2 [3]	When
Object2	GotFocus
Object2	Message
Object2	Valid [4]

Object2	LostFocus
Form	QueryUnload
Form	Destroy
Object [5]	Destroy
Form	Unload
Form set	Unload
Data environment	AfterCloseTables
Data environment	Destroy
Data environment cursor(s)	Destroy

1 Para cada objeto, do objeto mais interno ao recipiente mais externo

2 Primeiro objeto na ordem de tabulação

3 Próximo objeto a obter foco

4 Quando o objeto perde o foco

5 Para cada objeto, do recipiente mais externo ao objeto mais interno

Atribuindo códigos a eventos

A menos que você associe um código a um evento, nada acontecerá quando o evento ocorrer. Raras vezes, você criará código para todos os eventos associados a qualquer objeto do Visual FoxPro, mas convém incorporar funcionalidade em resposta a determinados eventos importantes em seus aplicativos. Para adicionar um código a ser executado quando ocorrer um evento, use a janela **Propriedades** no **Criador de formulários**.

A seqüência de eventos influi no local onde que você deve colocar o código. Siga as dicas abaixo:

- O evento Init de todos os controles em um formulário é executado antes do evento Init do formulário. Portanto, você pode incluir um código no evento Init do formulário para manipular qualquer um dos controles no formulário antes do formulário ser exibido.
- Se você quiser que alguns códigos sejam processados sempre que o valor de uma caixa de listagem, caixa de combinação ou caixa de verificação for alterado, associe-o ao evento InteractiveChange. O evento Click pode não ocorrer ou pode ser chamado, mesmo que o valor não tenha sido alterado.
- Quando você estiver arrastando um controle, os outros eventos do mouse serão suspensos. Por exemplo, os eventos MouseUp e MouseMove não ocorrem durante uma operação de arrastar e soltar.
- Como padrão, os eventos Valid e When retornam um valor verdadeiro (.T.). Se você retornar falso (.F.) ou 0 de um evento When, o controle não poderá obter o foco. Se você retornar falso (.F.) ou 0 do evento Valid, o foco não poderá deixar o controle.

Para obter maiores informações sobre a utilização do **Criador de formulários**, consulte o capítulo 9, Criando formulários. Para obter informações sobre a codificação de classes e a adição de códigos de evento, consulte o capítulo 3, Programação orientada a objetos.