

Você pode ampliar a eficácia dos aplicativos do Visual FoxPro empregando os pontos fortes de outros aplicativos ativados por OLE ou [controles ActiveX](#). Em formulários ou campos do tipo Geral de seus aplicativos, você pode incluir funcionalidades específicas ou dados como texto, som, figuras e vídeo de outros aplicativos. Pode visualizar ou manipular esses dados de modo visível utilizando o aplicativo que os criou. Ou, então, manipular os dados de modo invisível e automático controlando o aplicativo através da linguagem de programação com a automação.

Outros aplicativos também podem cooperar para a maior eficácia do Visual FoxPro através da automação. É possível até mesmo criar servidores OLE (componentes ActiveX) no Visual FoxPro que podem ser acessados por seus aplicativos ou por outros aplicativos local ou remotamente.

Este capítulo aborda os seguintes tópicos:

- Criando um aplicativo de OLE
- Adicionando objetos OLE a aplicativos
- Utilizando controles ActiveX
- Manipulando objetos com a automação
- Dividindo objetos em subclasses
- Controlando o Visual FoxPro a partir de outros aplicativos
- Criando servidores OLE
- Utilizando a automação remota

Criando um aplicativo de OLE

Os aplicativos ativados por OLE e os componentes ActiveX podem atuar como [servidores](#), clientes, ou ambos. Os componentes que atuam como servidores podem fornecer objetos a outros aplicativos, os componentes que atuam como clientes podem criar objetos.

Você pode incorporar facilmente a potência e a flexibilidade de aplicativos como o Microsoft Excel e o Word nos aplicativos do Visual FoxPro. Como o Visual FoxPro também atua como um servidor, é possível fornecer funcionalidade que pode ser integrada em pacotes de soluções baseados no Microsoft Office ou outros componentes ActiveX.

Os objetos OLE inseríveis são obtidos de aplicativos com o recurso de OLE como o Excel e o Word. Tais objetos incluem documentos do Word e planilhas do Excel. Em formulários, você pode vincular ou incorporar estes objetos utilizando o controle OLE Container, e armazená-los em campos do tipo Geral de uma tabela, exibindo-os nos formulários com o controle OLE Bound.

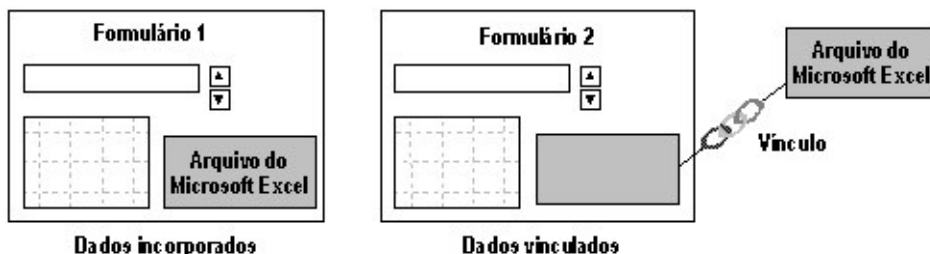
Em um aplicativo do Visual FoxPro, você pode utilizar OLE e a tecnologia ActiveX de várias maneiras. Antes de criar um aplicativo, considere as maneiras como poderia utilizá-lo.

Vinculando ou incorporando objetos OLE

Você pode incorporar ou vincular arquivos de outros aplicativos do Windows a tabelas e formulários. Por exemplo, incorporar ou vincular um documento do Word a um campo do tipo Geral de uma tabela e incorporar ou vincular uma planilha do Excel a um formulário.

A diferença entre incorporar e vincular está no local onde os dados são armazenados. A incorporação armazena os dados na tabela ou no formulário, enquanto que isso não acontece na vinculação. Por exemplo, quando você incorpora uma planilha do Excel a um formulário, ele passa a conter uma cópia da planilha. Mas quando você vincula, o formulário contém apenas uma referência à planilha—e não a planilha propriamente dita.

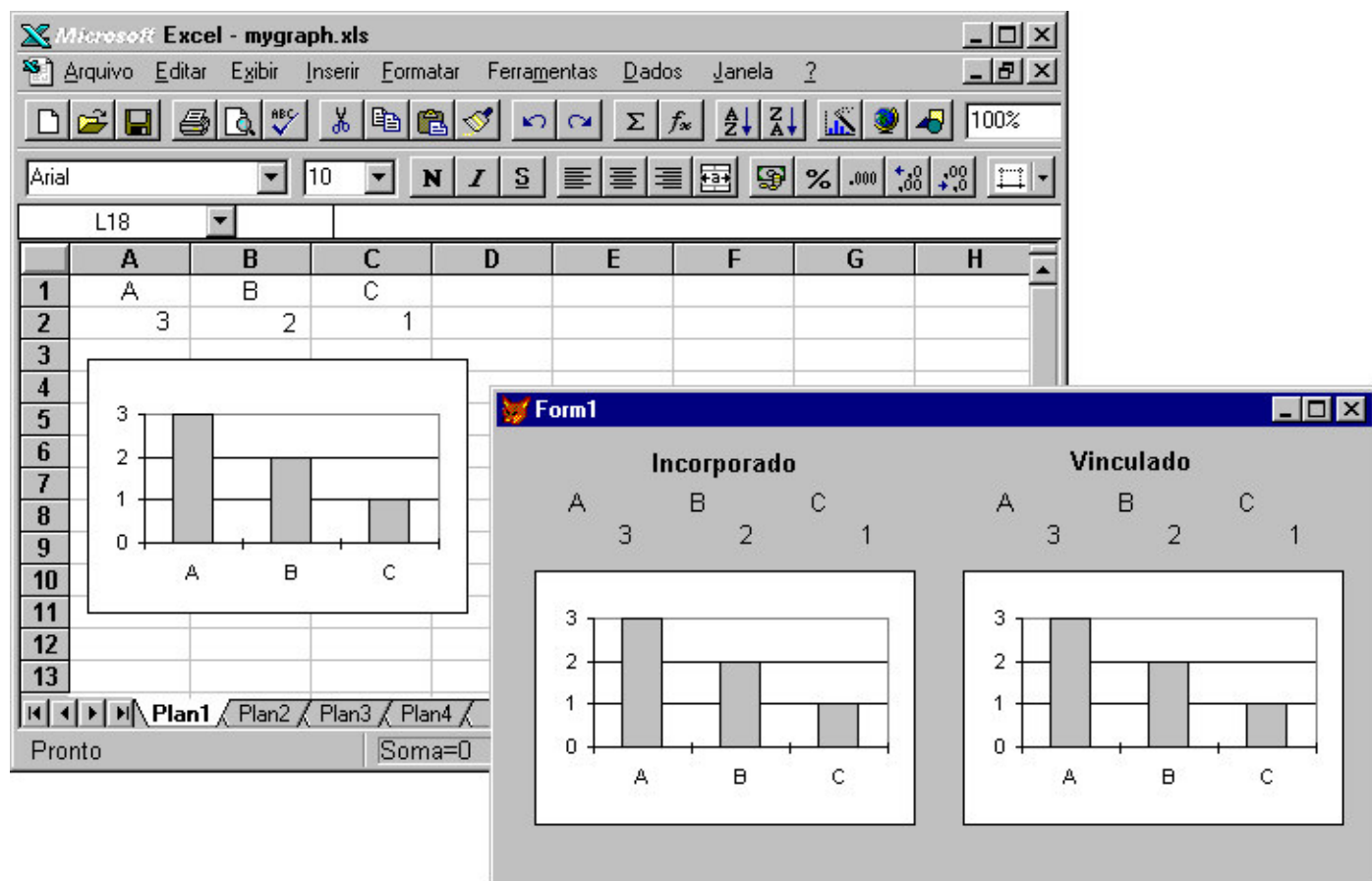
Incorporando e vinculando dados





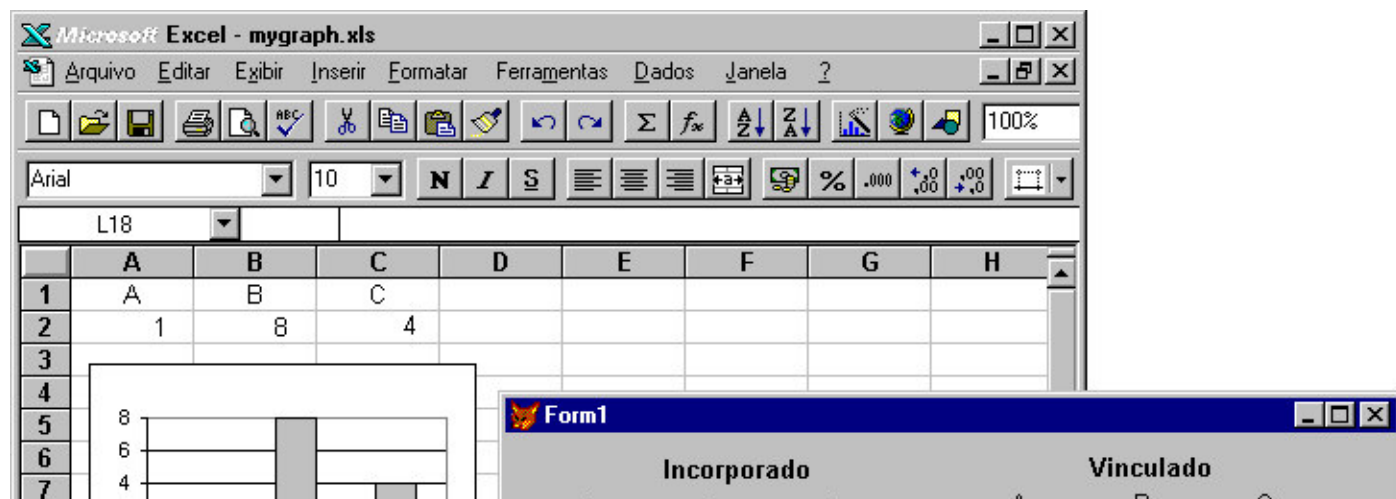
Tanto os dados incorporados como os vinculados são iniciados com o conteúdo original do arquivo do servidor, como mostra a ilustração:

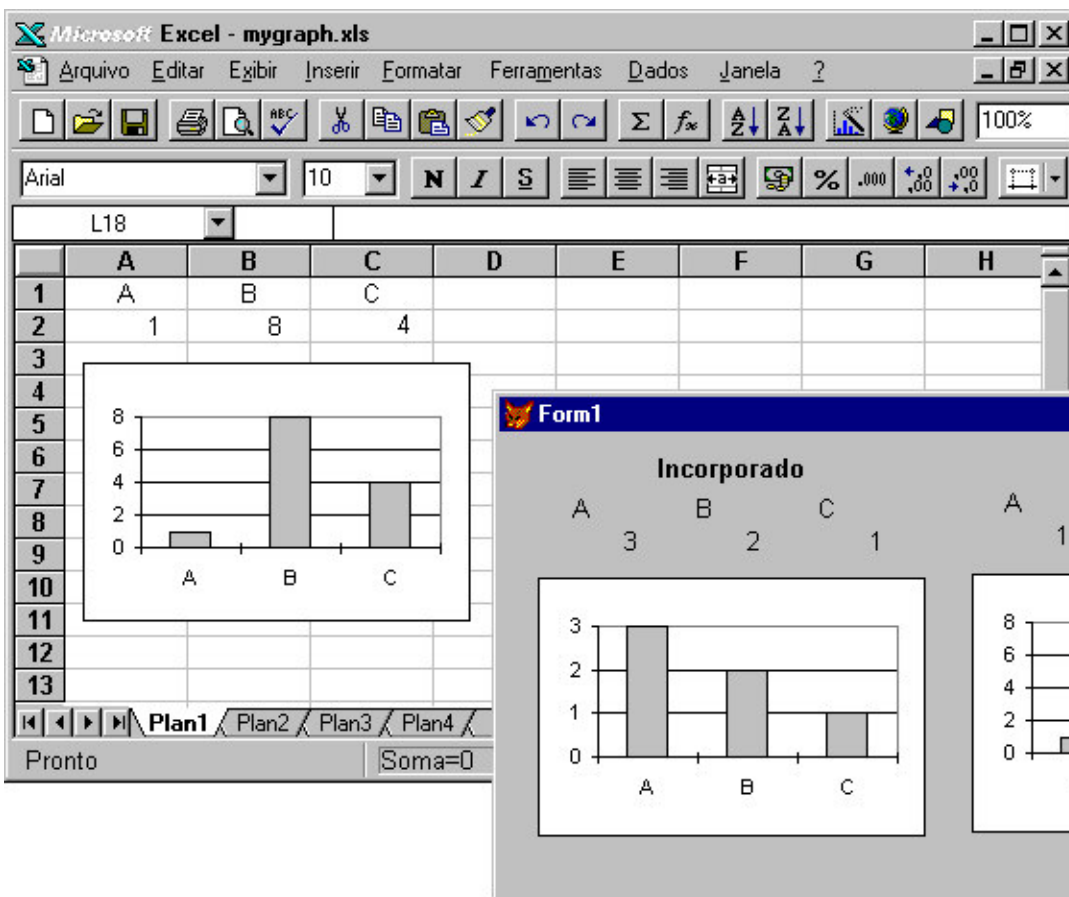
Uma planilha incorporada e vinculada em um formulário



Mas quando o arquivo original é alterado, os dados vinculados são automaticamente atualizados para que reflitam a alteração, enquanto os dados incorporados não são:

Dados vinculados atualizados em um formulário





Para obter uma versão animada, clique aqui.

Os dados incorporados não são necessariamente estáticos, no entanto. Tanto os dados incorporados quanto os vinculados podem ser exibidos, alterados e manipulados de modo interativo e através da linguagem de programação no Visual FoxPro.

Adicionando objetos OLE com ou sem ligação



Em um formulário ou relatório, você pode criar objetos que sejam ligados a campos do tipo Geral de tabelas. Tais objetos são chamados de *objetos OLE com ligação* e você pode utilizá-los para exibir o conteúdo de objetos OLE em campos do tipo Geral. Para criar objetos de OLE com ligação, utilize o controle **OLE Bound** na **Barra de ferramentas controles de formulário**. Ou, você pode criar *objetos OLE sem ligação* utilizando o controle **OLE Container**. Um objeto OLE sem ligação não está conectado a um campo do tipo Geral em uma tabela.

Adicionando objetos OLE a aplicativos

É possível adicionar objetos OLE a tabelas e formulários de modo interativo ou através da linguagem de programação.

Adicionando objetos OLE a tabelas

Quando estiver criando tabelas para o seu aplicativo, pense se você precisa de objetos OLE nessas tabelas. Por exemplo, suponha que você possua uma tabela de produtos e queira incluir documentos do Word que contenham descrições bem formatadas dos produtos para serem enviadas a

consumidores em potencial. Para incluir os documentos do Word, você deve definir um campo do tipo Geral na tabela. Em seguida, adicione os documentos à tabela vinculando ou incorporando-os ao campo do tipo Geral.

► **Para adicionar um objeto OLE a uma tabela**

- 1 Utilize o **Criador de tabelas** para criar uma tabela com um campo do tipo Geral.
- 2 Abra a janela do campo do tipo Geral pesquisando a tabela e clicando duas vezes no campo tipo Geral ou utilizando o comando **MODIFY GENERAL**.
- 3 No menu **Editar**, selecione **Inserir objeto**.
–Ou –
 - Utilize o comando **APPEND GENERAL**.

Para obter maiores informações sobre como adicionar objetos OLE com o **Criador de tabelas**, consulte o capítulo 10, **Compartilhando informações com outros aplicativos**, no *Guia do Usuário*.

Incluindo objetos OLE em tabelas

É possível adicionar objetos OLE a tabelas através da linguagem de programação utilizando o comando **APPEND GENERAL**. Com este comando, você pode importar um objeto OLE a partir de um arquivo e colocá-lo em um campo do tipo Geral. Se o campo já contiver um objeto, este será substituído pelo novo objeto.

Observação Diferentemente de APPEND e APPEND BLANK, APPEND GENERAL não adiciona um novo registro à tabela.

Você pode utilizar APPEND GENERAL para incorporar ou vincular a OLE objetos criados por aplicativos como o Excel e o Word. Estes dois aplicativos suportam a vinculação e a incorporação. No entanto, alguns aplicativos como o Microsoft Graph suportam somente a incorporação.

Suponha que você queira armazenar arquivos do Microsoft Word em uma tabela do Visual FoxPro. Se na tabela existir um campo do tipo Geral denominado `Documento do Word`, você poderá incorporar os documentos utilizando o código abaixo:

```
CREATE TABLE oletable (name c(24), worddoc g)
CD GETDIR()

nFiles = ADIR(aWordFiles, "*.doc")
IF nFiles > 0
    FOR i = 1 to nFiles
        APPEND BLANK
        REPLACE oletable.Name WITH aWordFiles(i,1)
        APPEND GENERAL WordDoc FROM aWordFiles(i,1)
    ENDFOR
ELSE
    MESSAGEBOX("Nenhum arquivo do Word encontrado.")
ENDIF
```

Observação O exemplo anterior procura apenas arquivos com a extensão .DOC, a extensão padrão utilizada por arquivos do Word. Visto que o Microsoft Word e a OLE estão cientes disto, os arquivos são automaticamente associados ao servidor do Word quando você utiliza APPEND GENERAL.

Se você utilizar uma extensão diferente daquela esperada pelo servidor, deverá declarar a classe do servidor, utilizando a cláusula CLASS. Por exemplo, se adicionar a classe para Word ao exemplo anterior, o código ficará assim:

```
APPEND GENERAL WordDoc FROM wordfiles(i,1) CLASS "Documento do Word. 6"
```

Se você tiver arquivos com extensões comuns (por exemplo, .BMP) que possam ser utilizados por outros servidores, poderá utilizar a cláusula CLASS para especificar o servidor específico que deseja utilizar para esses arquivos. Ou, se preferir vincular em vez de incorporar objetos, utilize a

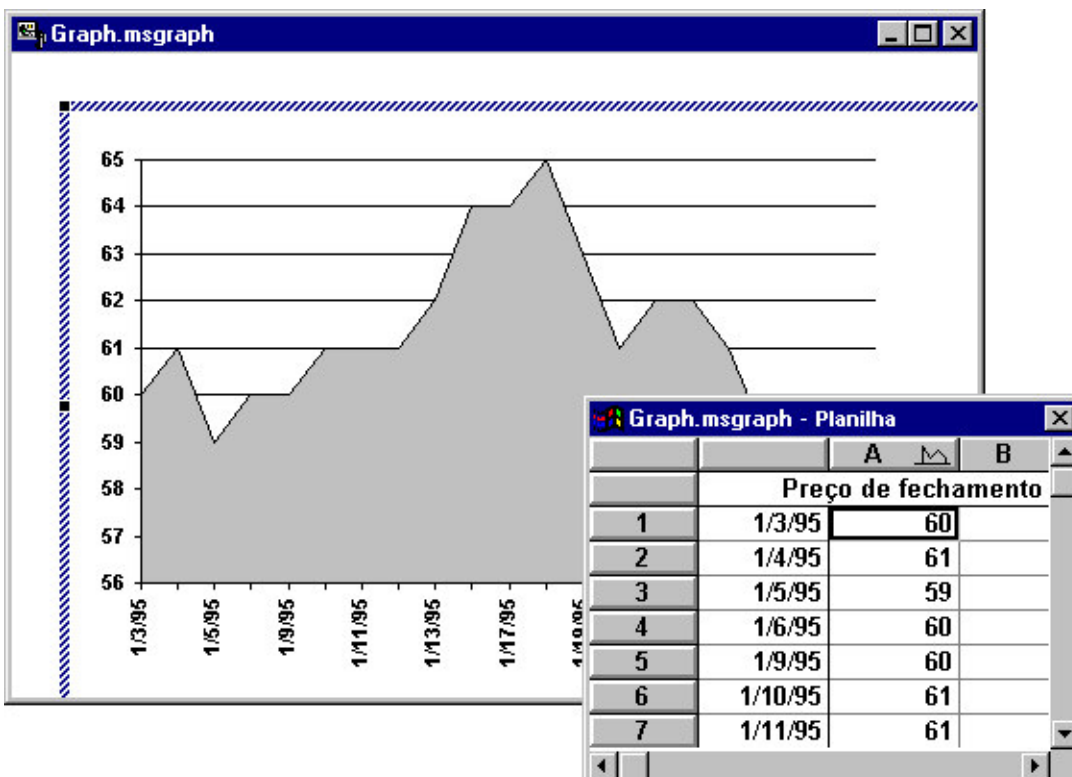
palavra-chave LINK.

APPEND GENERAL WordDoc FROM wordfiles(i,1) LINK CLASS "Documento do Word.6"
Além disso, você pode substituir dados em um objeto utilizando a palavra-chave DATA de [APPEND GENERAL](#), como ilustra o exemplo do Microsoft Graph a seguir.

Atualizando o Microsoft Graph

O Microsoft Graph é um aplicativo incorporável. Os valores em um gráfico do Microsoft Graph baseiam-se nos valores da planilha do Microsoft Graph.

Objeto do Microsoft Graph em um campo do tipo Geral



Para alterar os dados em um gráfico do Microsoft Graph utilizando a linguagem de programação, é necessário construir uma sequência que contenha os novos dados, inclusive tabulações, retornos de carro e alimentações de linha e transferir essa sequência a um objeto do Microsoft Graph com a cláusula DATA do comando [APPEND GENERAL](#).

O exemplo a seguir supõe que você possua uma tabela denominada `estoque` com, entre outros campos, `data` e `fechamento` para a data e o preço de fechamento do estoque. O objeto do Microsoft Graph está armazenado no campo do tipo geral `gráficos` de uma tabela denominada `gráfico`. O exemplo atualiza um gráfico com preços de fechamento de estoque dos últimos 30 dias.

Código

```
#DEFINE CRLF CHR(13)+CHR(10)
#DEFINE TAB CHR(9)
LOCAL lcData

SELECT date, close;
  FROM Stock WHERE BETWEEN(date, ;
    DATE(),DATE() - 30) ;
  ORDER BY date INTO CURSOR wtemp
```

Comentários

Define os caracteres de avanço de linha e de tabulação.

Seleciona os valores com os quais você deseja atualizar o gráfico; neste caso, os valores de data e fechamento de estoque dos últimos 30 dias.

```

SELECT wtemp
lcData = " " + ;
    TAB + "Preço de Fechamento" + CRLF
SCAN
    lcData = lcData + DTOC(date)
    lcData = lcData + TAB
    lcData = lcData + ;
    ALLTRIM(STR(close)) + CRLF
ENDSCAN

```

Constrói uma sequência de caracteres (lcData) proveniente dados do cursor para atualizar o gráfico.

“Preço de Fechamento”, como o cabeçalho da coluna, é o texto que será exibido como padrão na legenda do gráfico.

```

SELECT graph
APPEND GENERAL msgraph DATA lcData

```

Envia os novos valores para o gráfico na cláusula DATA do comando APPEND GENERAL.

```
USE IN wtemp
```

Fecha o cursor.

Observação Também é possível exibir objetos OLE de campos do tipo Geral em relatórios. Para obter detalhes sobre como exibir objetos OLE em relatórios, consulte a seção “Adicionando um campo geral” no capítulo 7, [Criando relatórios e etiquetas](#), no *Guia do Usuário*.

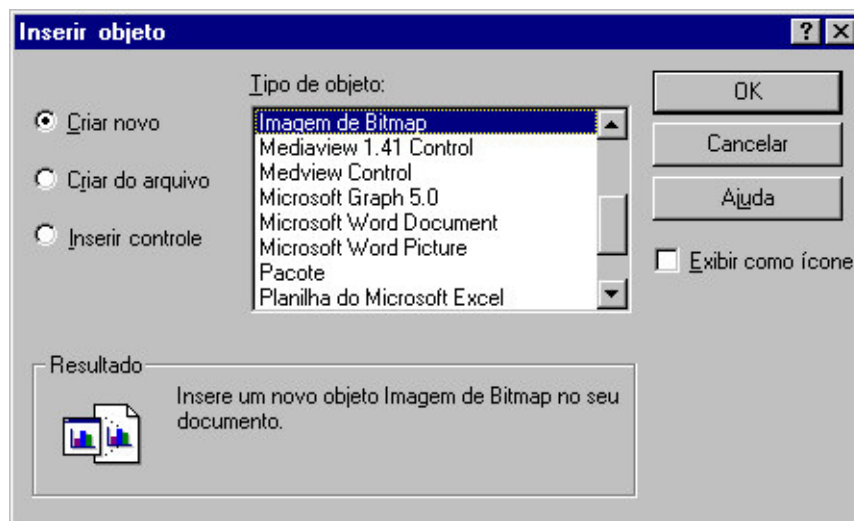
Adicionando objetos OLE a formulários

Utilizando o **Criador de formulários**, você poderá adicionar objetos OLE inseríveis a formulários com o controle [OLE Container](#). Além disso, você poderá exibir objetos OLE de campos do tipo Geral utilizando o controle [OLE Bound](#).

► Para adicionar um controle de OLE a um formulário

- 1 No **Criador de formulários**, adicione um controle [OLE Container](#) ao formulário. A caixa de diálogo **Inserir objeto** será aberta.
- 2 Na caixa de diálogo **Inserir objeto**, selecione **Criar novo** ou **Criar do arquivo**.

Caixa de diálogo Inserir objeto



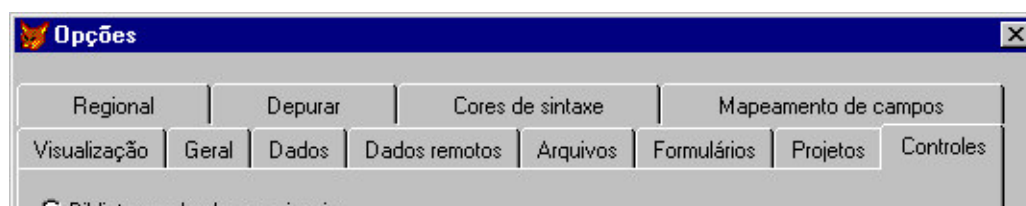
- 3 Selecione o controle de OLE apropriado na lista **Tipo de objeto**.

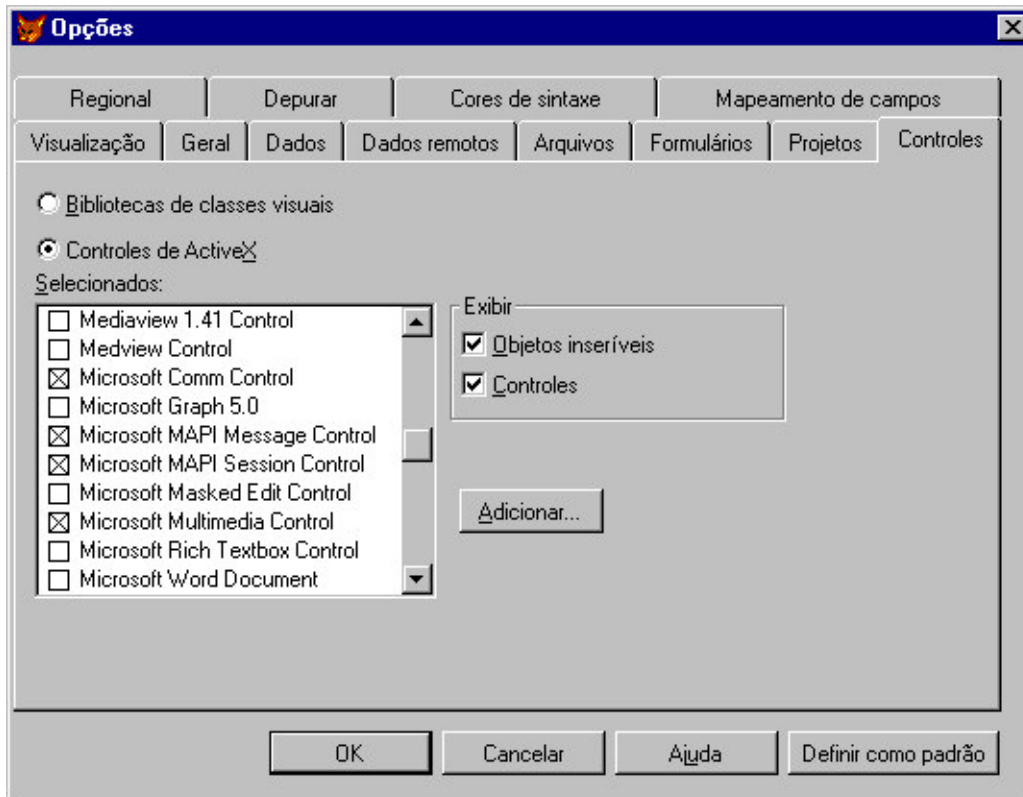
Você pode, também, personalizar a **Barra de ferramentas controles de formulário** para poder adicionar diretamente objetos OLE específicos.

► Para adicionar objetos OLE à Barra de ferramentas controles de formulário

- 1 No menu **Ferramentas**, selecione **Opções**.
- 2 Na guia **Controles** da caixa de diálogo **Opções**, selecione **Controles ActiveX**.

Guia Controles da caixa de diálogo Opções





- 3 Na lista **Selecionados**, selecione o objeto OLE e os controles ActiveX que deseja que estejam disponíveis na **Barra de ferramentas controles de formulário**.
- 4 Selecione **Definir como padrão** e, em seguida, selecione **OK**.
- 5 Na **Barra de ferramentas controles de formulários**, selecione **Exibir classes** e, em seguida, selecione **Controles ActiveX**.

► **Para exibir um objeto OLE de um campo do tipo Geral**

- 1 No **Criador de formulários**, adicione um controle **OLE Bound** ao formulário.
- 2 Especifique o campo do tipo Geral que contém os dados, definindo a propriedade **ControlSource** do objeto.

Por exemplo, se o nome da tabela for Estoque e o nome do campo do tipo Geral for Atual, defina a propriedade ControlSource como Estoque.Atual.

Também é possível exibir um objeto OLE de um campo do tipo Geral utilizando a linguagem de programação:

Código	Comentários
<code>frm1 = CREATEOBJECT("formulário")</code>	Cria formulário
<code>frm1.ADDOBJECT("olb1", "controleboundole")</code>	Adiciona controle
<code>frm1.olb1.ControlSource = "Inventário.Atual"</code>	Liga os dados ao controle
<code>frm1.olb1.Visible = .T. frm1.Visible = .T.</code>	Tornam visíveis o controle e o formulário.

Interagindo com objetos OLE

Se você adicionar um objeto OLE a um formulário ou campo do tipo Geral, será possível editar os dados e exibir características do objeto em tempo de execução ou de criação.

Observação Não é possível editar os dados de um objeto OLE em um controle OLEBound no tempo de criação.

Alguns objetos OLE suportam a edição local de modo que você possa editar o objeto na janela utilizada pelo aplicativo. Por exemplo, se o usuário clicar duas vezes em um objeto planilha do Excel em um campo do tipo Geral em vez de iniciar uma cópia do Excel em uma outra janela, os títulos de menu se alterarão para refletir a estrutura de menu do Microsoft Excel e as barras de ferramentas padrão do Excel. Você ou o usuário do seu aplicativo poderão, então, editar o objeto Excel sem sair do aplicativo.

Observação Só é possível editar os objetos incorporados, não os objetos vinculados.

Você pode, também, abrir o servidor OLE em outra janela, editar os dados ou exibir características ali e ter os novos valores refletidos no aplicativo quando retornar a ele.

► **Para editar um objeto OLE local em uma janela de campo do tipo Geral**

- No menu **Editar**, selecione o tipo de objeto específico e no submenu, selecione **Editar**.
Por exemplo, se o objeto for um documento do Word, selecione o item de menu **Objeto documento**; se o objeto for um gráfico do Microsoft Graph, selecione o item de menu **Objeto gráfico**.
–Ou –
- Clique duas vezes sobre o objeto.

► **Para abrir o aplicativo para um objeto OLE em uma janela de campo do tipo Geral**

- No menu **Editar**, selecione o tipo de objeto específico e, a partir do submenu, selecione **Abrir**.

Ao adicionar um objeto OLE a um formulário seja no controle OLE [Container](#), seja no controle de OLE [Bound](#), você terá maior controle sobre a abertura e a edição do objeto.

Você pode determinar se o objeto OLE será aberto ou editado quando o controle obtiver o foco ou quando o usuário clicar duas vezes no controle definindo a propriedade `AutoActivate` de um controle OLE de recipiente ou de ligação. A propriedade `AutoVerbMenu` indica que o menu de atalho do controle `ActiveX` permite que um usuário abra ou edite o objeto OLE.

Para controlar o acesso de forma que o objeto OLE só possa ser aberto ou editado através da linguagem de programação com o método [DoVerb](#), defina [AutoActivate](#) como 0 - Manual e [AutoVerbMenu](#) como falso (.F.) .

Controlando menus

Quando um usuário estiver editando um objeto OLE no local, a barra de menus exibe os menus do objeto OLE, não os menus do aplicativo. Se você criar um título de menu e quiser que ele seja exibido mesmo quando o usuário editar um objeto OLE, selecione `Negotiate` na caixa de diálogo **Opções de aviso** do criador de menus. Para obter maiores informações, consulte o capítulo 11, [Criando menus e barras de ferramentas](#), ou a cláusula `NEGOTIATE` no tópico da Ajuda [DEFINE PAD](#).

Usando controles ActiveX

Os controles `ActiveX` são objetos com funcionalidade encapsulada e propriedades, eventos e métodos expostos. Os controles `ActiveX` fornecem uma ampla gama de funcionalidades nas quais você pode entrar com facilidade. Os controles `ActiveX` fornecidos juntamente com o Visual FoxPro incluem:

- Controles do Windows 95, como os controles `RichText` e `TreeView`.
- Controles de sistema, como os controles `Communications` e `MAPI`.

Os controles `ActiveX` são versáteis porque você pode dividi-los em subclasses para criar outros controles e pode controlá-los usando os eventos, métodos e propriedades associadas com os controles. Não é possível criar controles `ActiveX` com o Visual FoxPro; no entanto, é possível criá-los

usando o Microsoft OLE Custom Control Developer's Kit fornecido juntamente com o Microsoft Visual C++™ 4.0.

Para obter maiores informações sobre como acessar controles ActiveX, consulte o capítulo 27, [Estendendo o Visual FoxPro com bibliotecas externas](#). Para obter maiores informações sobre a criação de controles ActiveX específicos ao Visual FoxPro, consulte o capítulo 28, [Acessando a API do Visual FoxPro](#).

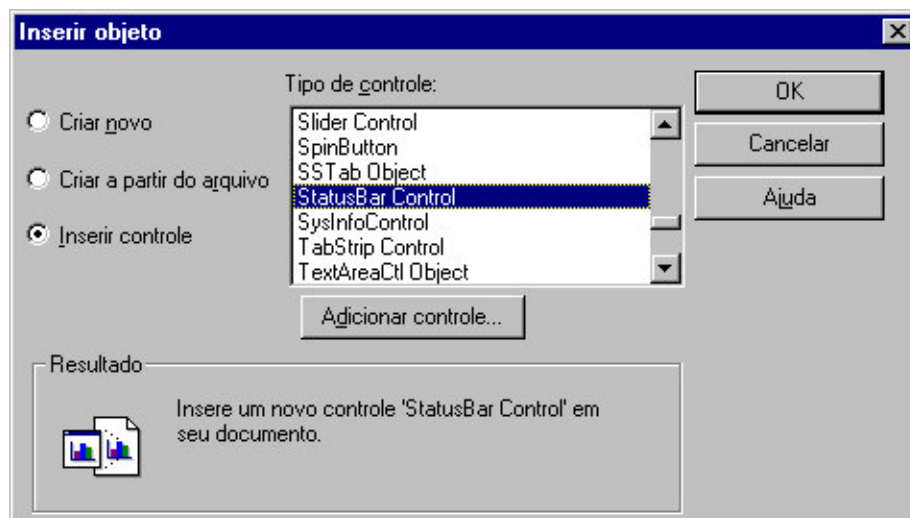
Adicionando controles ActiveX a um formulário

Os controles ActiveX no Visual FoxPro precisam estar contidos em um controle **OLE Container** (a classe base é OLEControl). Quando você adiciona um controle OLE Container a um formulário, pode escolher o controle ActiveX que deseja adicionar ao formulário.

► Para adicionar um controle ActiveX a um formulário

- 1 Na **Barra de ferramentas controles de formulário**, selecione **Controle OLE Container** e arraste-o para dimensionar no formulário.
- 2 Na caixa de diálogo **Inserir objeto**, selecione **Inserir controle**.

Caixa de diálogo Inserir objeto



- 3 Na lista **Tipo de controle**, selecione o controle ActiveX desejado.
- 4 Selecione **OK**.

Gerenciando controles ActiveX de ligação

Se um controle ActiveX suportar ligação de dados simples, o Visual FoxPro exporá uma propriedade **ControlSource** para o controle. Tudo o que você tem a fazer é definir a propriedade **ControlSource** como um campo de tabela e o valor exibido no controle ActiveX refletirá o valor no campo subjacente. As alterações no valor do controle são salvas no campo.

Para obter exemplos da utilização de controles ActiveX, execute SOLUTION.APP no diretório SAMPLES\SOLUTION.

Observação Para garantir que todos os eventos de controles ActiveX sejam processados, defina a propriedade **AutoYield** do objeto do Application do Visual FoxPro como falso (.F.).

Manipulando objetos com a automação

Os objetos OLE em seus formulários ou programas ou os controles ActiveX dentro de controles OLE Bound, podem ser manipulados pelo código da mesma forma que os objetos nativos do Visual

FoxPro podem ser programados.

Manipulando propriedades de objetos extrínsecos

Em código, você pode manipular um objeto utilizando as suas propriedades. A maneira como você faz referência a uma propriedade depende se o objeto é independente ou faz parte de um recipiente, como o controle [OLE Container](#) ou o controle [OLE Bound](#).

Observação Os controles ActiveX sempre fazem parte de um controle OLE Container.

Um objeto em um recipiente tem duas partes: o objeto propriamente dito e um recipiente ao redor do objeto. O objeto e o recipiente possuem propriedades e, às vezes, têm os mesmos nomes de propriedades. Para certificar-se de que está fazendo referência às propriedades do objeto, inclua sempre a propriedade [Object](#) do recipiente no nome do objeto. Por exemplo, o código abaixo refere-se à propriedade [Left](#) do objeto.

```
frm1.olecontroll1.Object.Left = 25  && Propriedade Left do objeto
```

Se omitir a propriedade [Object](#), você fará referência à propriedade [Left](#) do recipiente.

```
frm1.olecontroll1.Left= 25  && Propriedade Left do recipiente
```

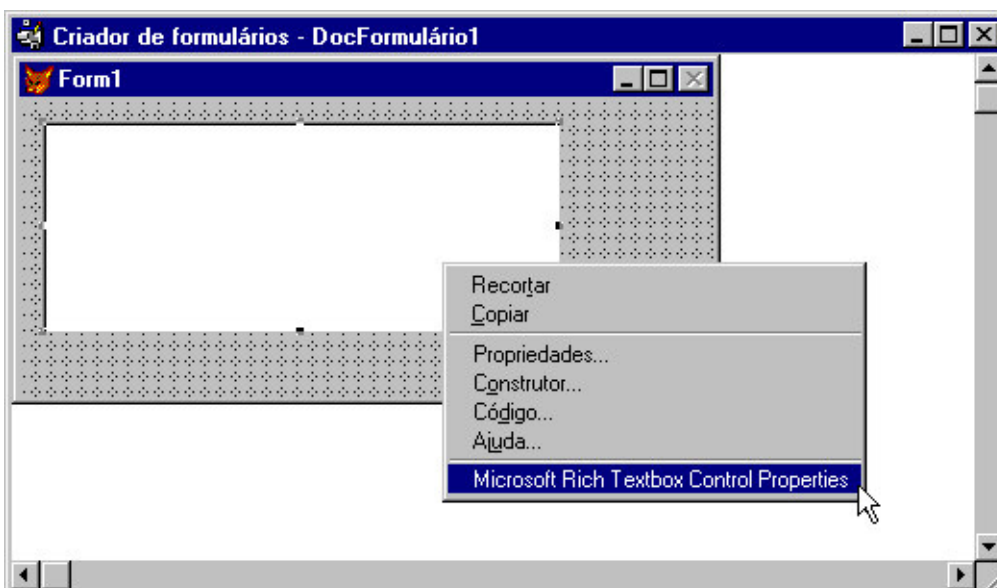
Por exemplo, suponha que você possua um aplicativo que envie correspondência eletrônica quando o usuário clica em um botão de comando de composição. Se você adicionou um controle de mensagem do Microsoft MAPI a um formulário como `olecontrole1`, o código associado ao evento [Click](#) do botão de comando poderia ser:

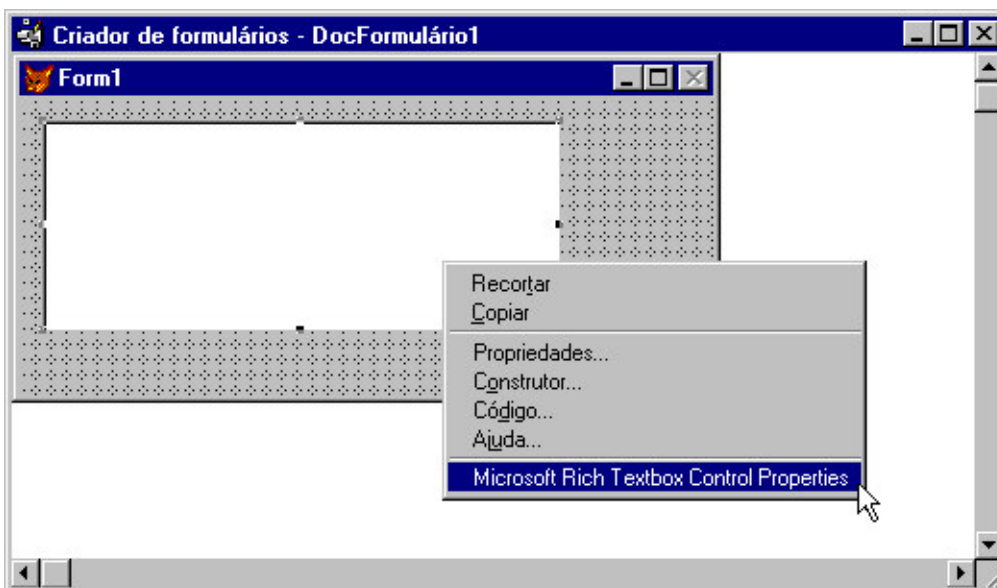
```
THISFORM.controlole1.Object.Compose  
THISFORM.controlole1.Object.Send(.T.)
```

Além de utilizar a propriedade [Object](#) para fazer referência às propriedades do objeto recipiente, você poderá utilizar outras propriedades do controle de recipiente. Por exemplo, utilizar a propriedade [OLEClass](#) somente-leitura para identificar o tipo de objeto no recipiente e a propriedade [Sizable](#) para evitar que os usuários alterem o tamanho de um objeto. Para obter detalhes sobre as propriedades dos controles de recipientes, consulte [Controle OLE Bound](#).

Nos **Criadores de formulário e de classe**, as propriedades dos controles ActiveX são exibidas na janela [Propriedades](#) do Visual FoxPro, mas a maioria dos controles ActiveX têm, também, sua própria interface para definir propriedades comuns. Você pode ver essa interface de propriedades selecionando a opção [Propriedades](#) específica do objeto do menu de atalho do controle ActiveX. Por exemplo, para abrir a caixa de diálogo **Propriedades** de um controle Rich Text, escolha **Propriedades de controle do RichText Microsoft** no menu de atalho.

Abrindo a caixa de diálogo de propriedades de controle Rich Text





Utilizando métodos de objetos extrínsecos

Além de definir e recuperar propriedades de objetos, você pode manipular um objeto utilizando métodos que ele suporta. Por exemplo, você pode utilizar o método Add de um objeto do conjunto Microsoft Excel para criar uma pasta de trabalho do Microsoft Excel.

O exemplo de automação abaixo utiliza o método Add para criar uma pasta de trabalho do Excel, o método Save para salvar a pasta de trabalho e o método Quit para encerrar o Excel:

Código

```
oleApp = CREATEOBJECT("Aplicativo.Excel")
oleApp.Visible=.T.
oleApp.Workbooks.Add

oleApp.Cells(1,1).Value=7

oleApp.ActiveWorkbook.SaveAs("C:\TEMP.XLS")
oleApp.Quit
```

Comentários

Inicia o Excel.
Exibe o Excel.
Cria uma pasta de trabalho.
Define o valor de uma célula.
Salva a pasta de trabalho.
Sai do Excel.

Se você criar um objeto utilizando o controle OLE Container ou o controle OLE Bound, você poderá utilizar o método [DoVerb](#) do controle para executar um verbo no objeto. Por exemplo, utilize [DoVerb\(0\)](#) para executar o verbo padrão, [DoVerb\(-1\)](#) para ativar o objeto para edição visual e [DoVerb\(-2\)](#) para abrir o objeto em uma janela separada.

Observação Consulte a documentação do aplicativo para determinar os comandos de automação por ele suportados. Por exemplo, os componentes suplementares do Microsoft Excel não estão disponíveis para a automação.

Definindo tempos limite

Quando você passa um requerimento a um objeto OLE, o servidor OLE o processa. Você não tem muito controle sobre o processamento do servidor, mas pode especificar o tempo de espera para que um processo seja concluído definindo as propriedades [OLERequestPendingTimeout](#) e [OLEServerBusyTimeout](#). Você pode determinar o que acontece quando o tempo expira definindo a propriedade [OLEServerBusyRaiseError](#).

Acessando conjuntos de objetos

Um tipo de objeto pode representar um único objeto ou um conjunto de objetos relacionados. Por exemplo, o objeto `Workbook` do Excel representa uma única pasta de trabalho enquanto que o objeto `Workbooks` representa todas as pastas de trabalho atualmente carregadas. Como o objeto `Workbooks` representa um conjunto de objetos, ele é chamado de objeto `Collection`.

Em código, um conjunto é uma lista não-ordenada na qual a posição de um objeto pode variar sempre que forem adicionados ou removidos objetos do conjunto. Para acessar um objeto de um conjunto, percorra o conjunto, utilizando a propriedade **Count** do conjunto. Essa propriedade retorna o número de itens no conjunto. Além disso, você pode utilizar o método `Item` para retornar um item em um conjunto.

Por exemplo, para exibir os nomes de planilhas em uma pasta de trabalho do Excel, utilize o código abaixo:

```
oleApp = CREATEOBJECT("Aplicativo.Excel")
oleApp.Workbooks.Add
FOR EACH x IN oleApp.Workbooks
    ? x.Name
ENDFOR
```

Também é possível acessar um conjunto dentro de um conjunto. Por exemplo, você pode acessar um conjunto de células dentro de um intervalo utilizando o código abaixo:

```
oleApp = CREATEOBJECT("Folha Excel ")
oleApp.Workbooks.Add
oleApp.Range(oleApp.Cells(1,1),oleApp.Cells(10,10)).Value=100
oleApp.Visible=.T.
```

Utilizando matrizes de objetos

Você pode passar matrizes para métodos e pode receber matrizes novamente. Entretanto, deve passar matrizes por referência, precedendo o nome da matriz com um sinal `@`.

Por exemplo, para enviar uma matriz do Visual FoxPro para o Excel, considere o código abaixo. Ele cria uma matriz no Visual FoxPro, atribui à matriz alguns valores, inicia o Excel, cria uma pasta de trabalho, define um valor para a primeira célula de uma planilha e, em seguida, copia este valor para as outras folhas da matriz:

```
DIMENSION aV(3)
aV(1) = "Folha1"
aV(2) = "Folha2"
aV(3) = "Folha3"
oleApp=CREATEOBJECT("Aplicativo.Excel ")
oleApp.Workbooks.Add
oleI=oleApp.Workbooks.Item(1)
oleI.Sheets.Item(1).Cells(1,1).Value = 83
oleI.Sheets(@aV).;
    FillAcrossSheets(oleI.Worksheets("Folha1").Cells(1,1))
oleApp.Visible = .T.
```

Ou, o exemplo abaixo retorna uma matriz para o Visual FoxPro e, em seguida, exibe o conteúdo da matriz:

```
oleApp = CREATEOBJECT("Aplicativo.Excel ")
aOleArray = oleApp.GetCustomListContents(3)
FOR nIndex = 1 to ALLEN(aOleArray)
    ? aOleArray(nIndex)
ENDFOR
```

Observação Com o Visual FoxPro, não é possível passar matrizes com mais de duas dimensões para objetos OLE. Para obter maiores informações sobre como trabalhar com matrizes no Visual FoxPro, consulte o capítulo 3, [Programação orientada a objetos](#) e [Visão geral da linguagem](#).

Liberando objetos extrínsecos

Um servidor OLE é automaticamente liberado se não for visível e nenhuma variável no escopo fizer referência ao objeto. Você pode utilizar o comando **RELEASE** para liberar a variável associada ao objeto. Se o servidor estiver visível, utilize o método **Quit** para liberá-lo.

Dividindo objetos em subclasses

Você pode criar objetos personalizados, dividindo em subclasses as classes principais fornecidas com o Visual FoxPro. Por exemplo, o código abaixo divide em subclasses o controle **Outline** fornecido com o Visual FoxPro:

Dividindo o controle Outline em subclasses

Código	Comentários
<pre>PUBLIC frmMeuForm, cNomeArq SET SAFETY OFF FrmMeuForm = CREATEOBJECT("formulário") frmMeuForm.Width = 100 frmMeuForm.ADDOBJECT("oleOutl", "meuoutline") DIMENSION aSection(3) aSection(1) = "Tabela" aSection(2) = "Campo" aSection(3) = "Índice" cFilename = GETFILE("dbc", "Selecione um DBC") USE (cNomeArq) INDEX ON objecttype FOR (objecttype = "Tabela" ; OR objecttype = "Campo" ; OR objecttype = "Índice") ; TAG fname FOR nÍndice = 1 TO 3 STEP 1 frmMeuForm.oleOutl.AddItem(aSection(nÍndice)) frmMeuForm.oleOutl.Indent; ((frmMeuForm.oleOutl.ListCount-1)) = 1 SCAN IF objecttype = aSection(nIndex) frmMeuForm.oleOutl.Additem(objectname) frmMeuForm.oleOutl.Indent; ((frmMeuForm.oleOutl.ListCount-1)) = 2 ENDIF ENDSCAN GO TOP ENDFOR FrmMeuForm.oleOutl.Visible = .T. frmMeuForm.Show DEFINE CLASS meuoutline AS olecontrol OleClass = "msoutl.outline" Top = 5 Left = 5 Height = 10 Width = 60 ENDDEFINE</pre>	<p>Declara variáveis e inicializa.</p> <p>Cria um formulário, adiciona o controle Outline personalizado ao formulário e cria uma matriz para os itens listados pelo controle.</p> <p>Solicita um banco de dados que contém as informações que você quer que o controle liste.</p> <p>Reúne informações do banco de dados e as adiciona ao controle.</p> <p>Torna o controle visível e exibe o formulário.</p> <p>Define uma subclasse do controle OLE Container e adiciona o controle Outline de contorno definindo a propriedade OLEClass do recipiente e, em seguida, definindo outras definições personalizadas.</p>

Se você quiser distribuir seus aplicativos, será preciso levar em conta alguns aspectos. Para obter maiores informações, consulte o capítulo 25, [Construindo um aplicativo para distribuição](#).

Controlando o Visual FoxPro a partir de outros aplicativos

Como o Visual FoxPro atua como servidor (com compatibilidade de nível 2) e também como cliente, os aplicativos que suportam a automação podem criar instâncias do Visual FoxPro, executar os comandos do Visual FoxPro e acessar objetos do Visual FoxPro.

Você pode até mesmo manipular o Visual FoxPro a partir de aplicativos que não suportam automação utilizando [FPOLE.DLL](#).

É possível controlar o Visual FoxPro a partir de outros aplicativos utilizando o objeto **Application** do Visual FoxPro. Um objeto Application é automaticamente criado sempre que o Visual FoxPro é iniciado diretamente, por DDE, ou pela automação.

Por exemplo, as linhas de código a seguir no Visual Basic® ou um módulo do Microsoft Excel criam uma referência a um objeto de aplicativo do Visual FoxPro:

```
Dim oFox as Object
Set oFox = CreateObject("Aplicativo do VisualFoxPro ")
```

Uma vez que você tenha uma referência ao objeto **Application** do Visual FoxPro, poderá chamar métodos associados com o objeto de aplicativo e acessar outros objetos através das propriedades de conjunto do objeto **Application**.

Métodos do objeto Application

DataToClip	Help
DoCmd	Quit
Eval	RequestData

O exemplo a seguir usa código do Visual Basic para Aplicativos em um módulo de Excel para criar um objeto **Application** do Visual FoxPro, abrir uma tabela do Visual FoxPro e adicionar resultados de uma consulta à planilha ativa.

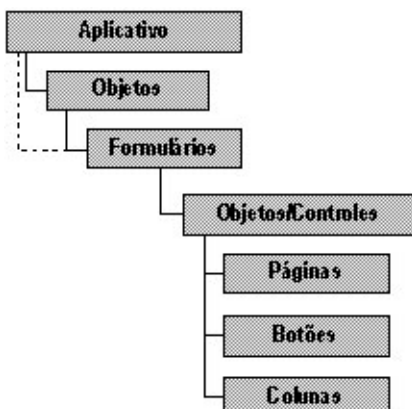
```
Sub FoxTest()
Dim oFox as Object
Set oFox = CreateObject("Aplicativo do VisualFoxPro ")

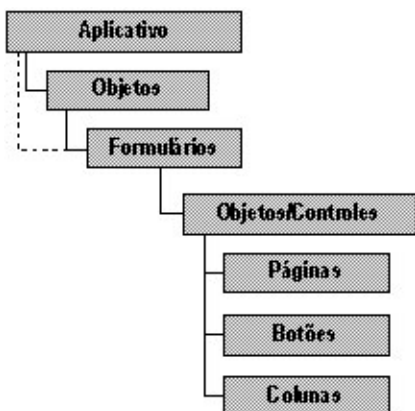
oFox.DoCmd "USE cliente"
oFox.DoCmd "SELECT o contato, telefone FROM cliente
    WHERE país = " + Chr$(39) + USA+ Chr$(39) + " INTO CURSOR cust"
oFox.DataToClip "cust",,3
Range("A1:B1").Select
ActiveSheet.Paste
End Sub
```

O modelo de objeto Application do Visual FoxPro

Um objeto de aplicativo é automaticamente criado sempre que o Visual FoxPro é iniciado, seja diretamente, seja através de automação ou DDE. Esse objeto de aplicativo fornece acesso a todos os objetos criados em uma sessão do Visual FoxPro através das propriedades **Collection**.

O modelo de objeto de aplicativo do Visual FoxPro





Acessando objetos através de propriedades Collection

O objeto de aplicativo do Visual FoxPro e todos os objetos de recipiente no Visual FoxPro têm uma propriedade **Count** e uma propriedade **Collection** associadas a eles. A propriedade **Collection** é uma matriz que faz referência a cada objeto recipiente. A propriedade **Count** é uma propriedade numérica que indica o número de objetos contidos.

A tabela a seguir lista objetos e o conjunto correspondente e propriedades Count.

Objeto	Propriedade Collection	Propriedade Count
Application	Objects	Count
	Forms	FormCount
FormSet	Forms	FormCount
Form	Objects	Count
	Controls	ControlCount
PageFrame	Pages	PageCount
Page	Controls	ControlCount
Grid	Columns	ColumnCount
CommandGroup	Buttons	ButtonCount
OptionGroup	Buttons	ButtonCount
Column	Controls	ControlCount
ToolBar	Controls	ControlCount
Container	Controls	ControlCount
Control	Controls	ControlCount

Essas propriedades permitem que você utilize um loop para manipular programaticamente todos os objetos contidos ou alguns específicos. Por exemplo, as linhas de código a seguir definem a propriedade **Visible** de todos os formulários como Verdadeira (.T.):

```

FOR EACH Form IN Application.Forms
    Form.Visible = .T.
ENDFOR
  
```

Criando servidores OLE

Com o Visual FoxPro, é possível criar servidores OLE (componentes ActiveX) que compactam códigos para realizar tarefas comuns a muitos aplicativos ou que implementam regras comerciais complexas. Essas tarefas e regras estarão, então, disponíveis para outros programadores da sua empresa e a usuários de ferramentas que suportam automação.

Por exemplo, você poderia criar uma ou mais classes para manipular regras comerciais em nível

empresarial. Um aplicativo cliente que utiliza os objetos de regras comerciais passariam parâmetros de entrada em uma chamada de método e o servidor OLE poderia então realizar uma grande quantidade de trabalho, recuperando dados de várias fontes e realizando cálculos complexos, antes de retornar a resposta.

Exemplos de servidores OLE estão instalados em VFP\SAMPLES\SERVERS.

Criando o servidor

Tudo o que você precisa para criar um Servidor OLE no Visual FoxPro é um projeto que contenha classes definidas como OLEPUBLIC. Você pode ter quantas classes OLEPUBLIC desejar no projeto e podem ser definidas em arquivos de programa (.PRG) ou bibliotecas de classe (.VCX).

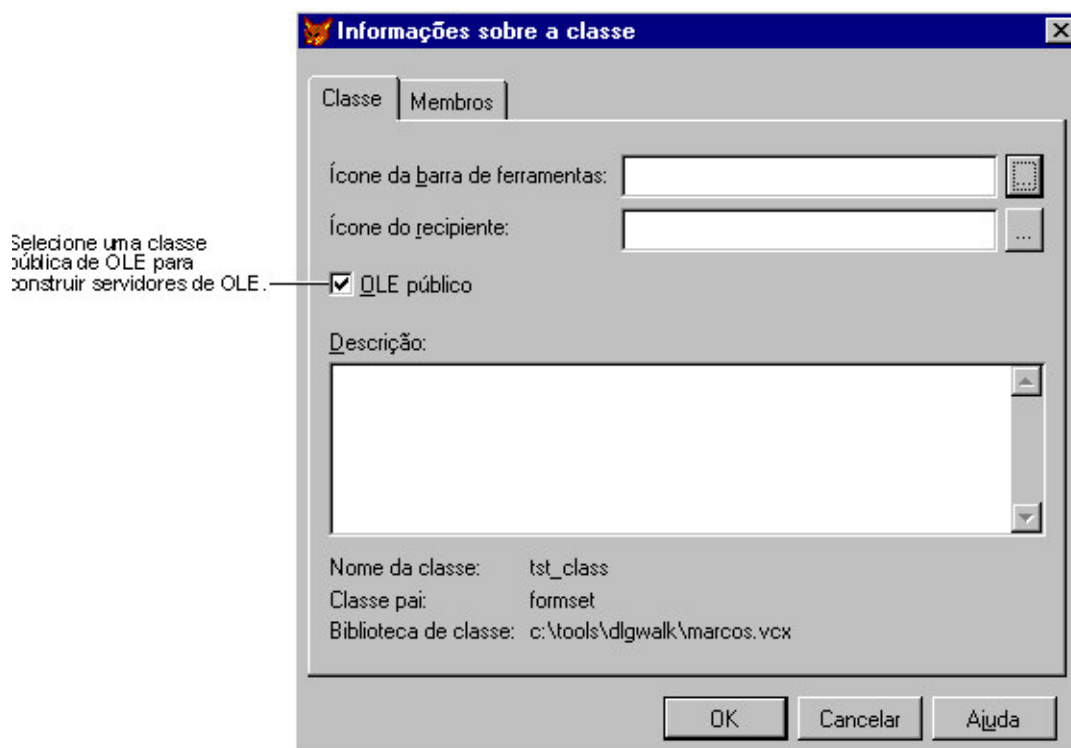
Por exemplo, a definição de classe a seguir em um arquivo de programa cria uma classe pública de OLE personalizada:

```
DEFINE class person AS CUSTOM OLEPUBLIC
  FirstName = SPACE(30)
  LastName = SPACE(45)

  PROCEDURE GetName
    RETURN THIS.FirstName + " " + This.LastName
  ENDPROC
ENDDEFINE
```

Ao criar uma classe no **Criador de classes**, selecione **OLE Pública** na caixa de diálogo **Informações sobre a classe** para designar a classe como OLEPUBLIC.

Caixa de diálogo Informações sobre a classe



Compilando o servidor

No Visual FoxPro, você pode criar um servidor OLE [fora-do-processo](#) ou um servidor OLE [dentro-do-processo](#). Um componente *fora-do-processo* é uma executável (arquivo .EXE) que executa seu próprio processo. A comunicação entre um aplicativo cliente e um servidor

fora-do-processo é, portanto, chamada de comunicação de *processos-cruzados*. Um componente *dentro-do-processo* é uma biblioteca de vínculo dinâmico (DLL) que é executada no mesmo espaço de endereço de processo que o cliente que a chama.

Cada um deles apresenta as suas vantagens. Um servidor dentro-do-processo é mais rápido porque não há comunicação inter-processo fixa. Por outro lado, um servidor fora-do-processo pode ser localizado remotamente e um servidor dentro-do-processo não. Além disso, como o servidor dentro-do-processo e o cliente compartilham um espaço de endereço de processo, qualquer erro grave na .DLL encerrará o cliente enquanto um erro em um .EXE fora-do-processo encerraria apenas o servidor.

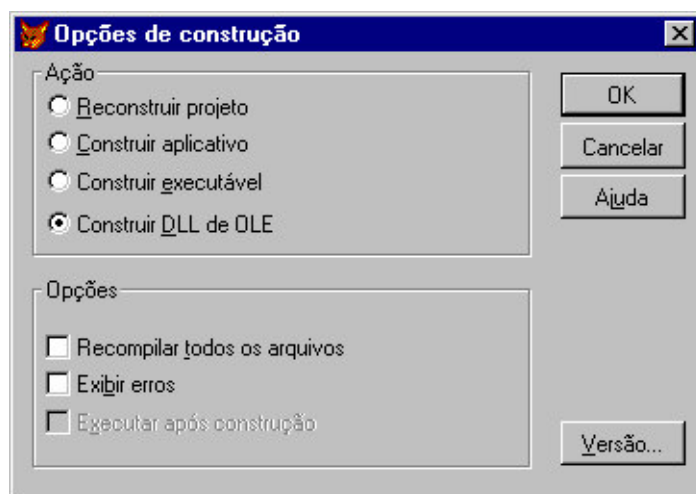
Ao criar uma executável com classes Públicas de OLE, você não perde nenhuma de suas capacidades .EXE normais. Você pode, ainda, executar a executável, fornecer uma interface com o usuário e todas as funcionalidades normais que seriam incluídas em um aplicativo. Aumente a capacidade de ampliação do aplicativo, contudo, permitindo que outros aplicativos penetrem na funcionalidade específica que deseja expor.

Observação Se houver mais de um usuário acessando o servidor OLE, pode haver conflitos. Se você forneceu acesso a automação e uma interface com o usuário para a sua funcionalidade, forneça à interface uma camada extra de verificação de consistência para certificar-se de que o ambiente não foi alterado.

► Para compilar um servidor OLE

- 1 No **Gerenciador de projetos**, selecione **Construir**.
- 2 Na caixa de diálogo **Opções de construção**, selecione **Construir executável** ou **Construir DLL de OLE**.

Caixa de diálogo Opções de construção



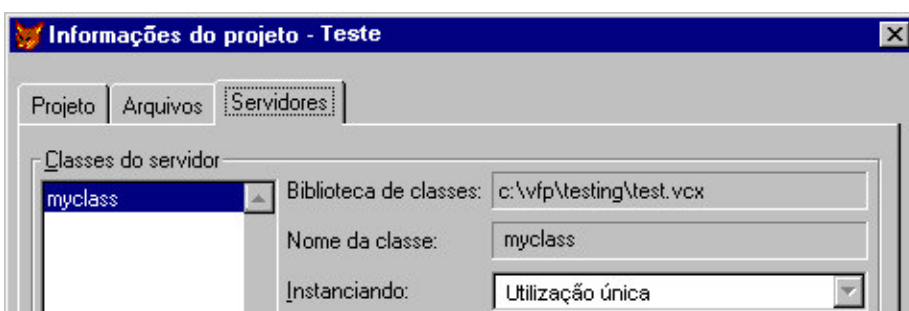
- 3 Selecione **OK**.

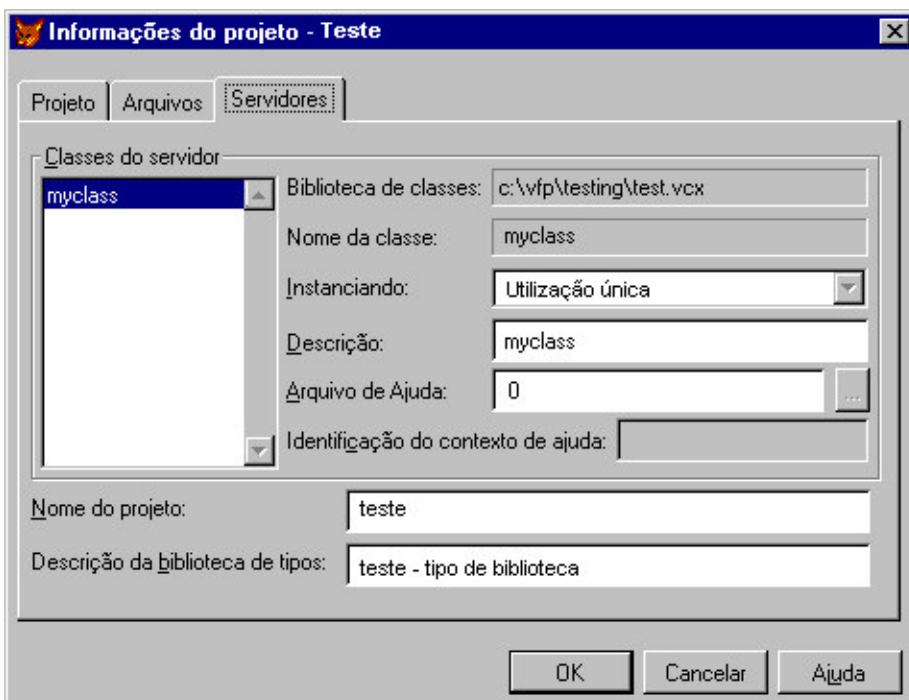
–Ou –

- Use os comandos **BUILD DLL** ou **BUILD EXE**.

Uma vez construído o projeto, você pode ver as classes de servidor exibidas na caixa de diálogo **Informações sobre o projeto**. Aqui você pode, também, especificar um arquivo de ajuda e uma Identificação do contexto da **Ajuda** para cada classe. Esse arquivo de ajuda pode ser aberto a partir da maioria dos pesquisadores de objetos genéricos.

Caixa de diálogo Informações sobre o projeto





Você pode selecionar valores de instância específicos de classe na caixa de diálogo **Informações sobre o projeto**. As opções de instância são:

- **Impossível criar** Mesmo que a classe esteja marcada como pública de OLE, não estará disponível para outros aplicativos. Por exemplo, você poderia ter uma biblioteca padrão de classes públicas de OLE utilizadas em vários aplicativos e desativar a automação de uma ou mais classes para um único aplicativo.
- **Utilização única** Cada aplicativo cliente que utiliza o seu servidor cria uma instância separada de classe do servidor. Cada instância tem uma única linha de execução. Embora instâncias individuais exijam mais memória, a seleção de **Utilização única** permite que o sistema operacional aplique multitarefas preemptivas.
- **Múltiplas utilizações** Criado o servidor, outros aplicativos podem utilizar a mesma instância.

Observação Se você fizer alterações na guia **Servidor** da caixa de diálogo **Informações sobre o projeto**, será necessário reconstruir a .DLL ou .EXE para que as novas definições sejam ativadas.

Quando você constrói um projeto com as classes públicas de OLE, são criados três arquivos:

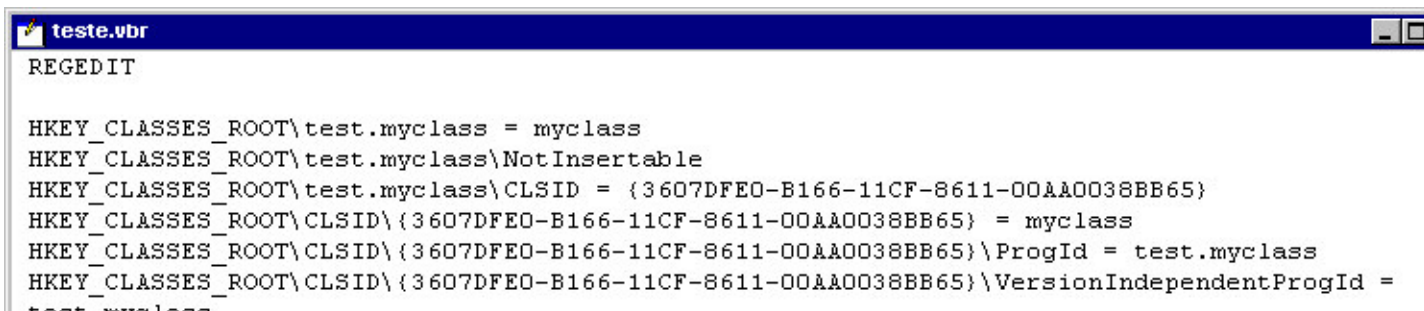
- A .DLL ou .EXE
- Um arquivo de biblioteca de tipos (.TLB)
- Um arquivo de registro (.VBR)

O arquivo de biblioteca de tipos é um arquivo binário que lista todas as classes publicadas em seu servidor OLE, junto com suas propriedades, métodos e eventos. Os pesquisadores de objetos OLE lêem estas informações e a apresentam em uma interface legível.

O arquivo de registro lista as [Identificações únicas globais](#) (GUID) para as classes em seu servidor.

Observação Um arquivo de registro .VBR é igual a um arquivo de registro .REG a não ser pelo fato de que o arquivo .VBR não inclui caminhos codificados.

Um arquivo .VBR com GUIDs para cada classe pública de OLE em um projeto





```
teste.vbr
REGEDIT

HKEY_CLASSES_ROOT\test.myclass = myclass
HKEY_CLASSES_ROOT\test.myclass\NotInsertable
HKEY_CLASSES_ROOT\test.myclass\CLSID = {3607DFE0-B166-11CF-8611-00AA0038BB65}
HKEY_CLASSES_ROOT\CLSID\{3607DFE0-B166-11CF-8611-00AA0038BB65} = myclass
HKEY_CLASSES_ROOT\CLSID\{3607DFE0-B166-11CF-8611-00AA0038BB65}\ProgId = test.myclass
HKEY_CLASSES_ROOT\CLSID\{3607DFE0-B166-11CF-8611-00AA0038BB65}\VersionIndependentProgId =
test.myclass
HKEY_CLASSES_ROOT\CLSID\{3607DFE0-B166-11CF-8611-00AA0038BB65}\InProcServer32 =
c:\vfp4\testing\test.dll
HKEY_CLASSES_ROOT\CLSID\{3607DFE0-B166-11CF-8611-00AA0038BB65}\TypeLib =
{3607DFE1-B166-11CF-8611-00AA0038BB65}

; TypeLibrary registration
HKEY_CLASSES_ROOT\TypeLib\{3607DFE1-B166-11CF-8611-00AA0038BB65}
HKEY_CLASSES_ROOT\TypeLib\{3607DFE1-B166-11CF-8611-00AA0038BB65}\1.0 = test Type Library
HKEY_CLASSES_ROOT\TypeLib\{3607DFE1-B166-11CF-8611-00AA0038BB65}\1.0\0\win32 =
c:\vfp4\testing\test.tlb
```

Registrando um servidor OLE

Os seus servidores OLE estão disponíveis para outros aplicativos uma vez que os servidores tenham sido adicionados ao Registro do Windows. Quando você constrói um servidor OLE, ele é automaticamente registrado na máquina em que foi construído. Você pode, também, registrar seus servidores em outras máquinas.

Quando você utiliza o **Assistente de instalação** do Visual FoxPro para criar discos de instalação, o programa de instalação registra os seus servidores nas máquinas dos clientes. Também é possível registrar servidores manualmente.

► Para registrar um componente .EXE

- Execute o arquivo .EXE com a chave **/regserver**.

Por exemplo, para registrar MYSERVER.EXE, execute o comando a seguir:

```
myserver /regserver
```

► para remover uma entrada de registro de um componente .EXE

- Execute o arquivo .EXE com a chave **/unregserver**.

Por exemplo, para anular o registro de Myserver.exe execute o comando a seguir:

```
myserver /unregserver
```

► Para registrar um componente .DLL

- Execute REGSVR32 com o nome do servidor.

Por exemplo, para registrar Myserver.dll execute o comando a seguir:

```
REGSVR32 myserver.dll
```

► Para remover uma entrada de registro de um componente .DLL

- Execute REGSVR32 com o nome do servidor e a chave **/u**.

Por exemplo, para registrar Myserver.dll execute o comando a seguir:

```
REGSVR32 /u myserver.dll
```

Observação O registro contém o caminho completo para o arquivo, assim, se você mover o

arquivo, será necessário registrá-lo novamente.

Usando o servidor OLE

Qualquer aplicativo que possa criar objetos de automação pode criar objetos baseados no servidor OLE, definir propriedades que não são HIDDEN ou PROTECTED e chamar métodos. Por exemplo, pressupondo que o servidor se chame `foxole` e contenha uma classe denominada `person` com um método `GetName`, o código a seguir poderia ser executado no Visual FoxPro 3.0:

```
oTest = CREATEOBJECT("olefox.pessoa")
cName = oTest.GetName()
```

Um código semelhante poderia ser executado no Microsoft Excel ou no Visual Basic:

```
Set oTest = CreateObject("olefox.pessoa")
cName$ = oTest.GetName()
```

Criando ou retornando erros de servidores OLE

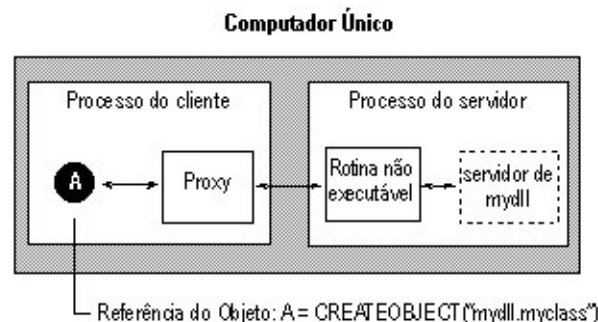
A única interação com os objetos fornecidos por um servidor OLE (componente ActiveX) é através dos métodos e propriedades das classes expostas. Quando um aplicativo cliente chama um método de um objeto e ocorre um erro no servidor OLE, o método retorna um valor de erro ou cria um erro no aplicativo cliente.

O aplicativo cliente decide se deve alertar o usuário ou continuar com outro caminho de execução. O próprio servidor OLE nunca interage com o usuário. Isso permite que a localização do servidor OLE seja transparente para o aplicativo cliente. O servidor OLE pode ser local, sendo executado no computador do usuário ou pode utilizar o recurso automação Remota do Visual FoxPro para executá-lo em um servidor de rede.

Usando a automação remota

Em cenários de automação normais, tanto o cliente quanto o servidor estão em um único computador e compartilham os mesmos recursos, como memória e processador.

Automação em um único computador

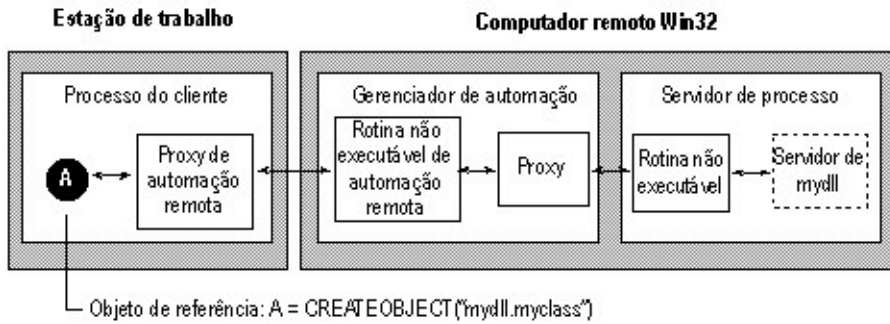


Ao criar servidores locais para automação, você pode distribuí-los remotamente. A automação remota permite a mesma flexibilidade, capacidade de extensão e eficácia que a automação local, mas em uma rede. A automação remota permite que:

- Os servidores utilizem recursos individuais.
- Muitos usuários diferentes acessem o mesmo servidor.

É possível configurar um servidor e um computador local para automação remota com o **Remote Automation Connection Manager (Gerenciador de conexão de automação remota)**, que salva as definições no registro. O **Gerenciador de automação** sendo executado no computador servidor gerencia a automação de modo que o mesmo código que manipula um objeto local possa manipular automaticamente um objeto remoto.

Automação remota

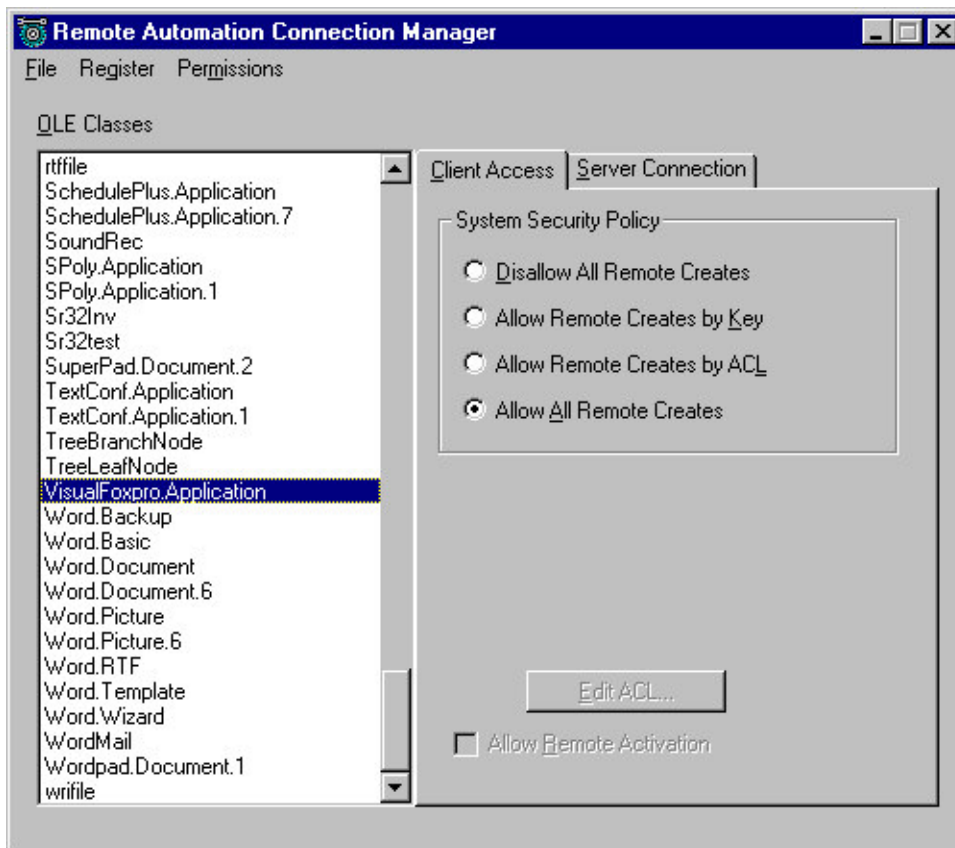


Configurando o servidor

A primeira etapa para ativar a automação remota é configurar o computador servidor para acesso de cliente no **Gerenciador de conexão de automação remota**.

► Para configurar o servidor de automação remota

- 1 Copie o arquivo executável do servidor OLE (.EXE) para o servidor e execute-o uma vez para registrá-lo no Registro do Windows como descrito em [Registrando um servidor OLE](#).
- 2 No computador servidor, execute RACMGR32.EXE, o **Gerenciador de conexão de automação remota**.



- 3 Selecione a sua classe na lista **OLE Classes (Classes OLE)**.
- 4 Na guia **Client Access (Acesso de clientes)**, certifique-se de que **Allow Remote Activation (Permitir ativação remota)** esteja selecionado.

Após ter ativado o acesso de cliente no **Gerenciador de conexão de automação remota**, execute o

Gerenciador de automação, AUTMGR32.EXE no computador servidor. AUTMGR32.EXE está instalado na pasta SYSTEM no Windows 95 ou na pasta SYSTEM32 no Windows NT. Isso ativará as conexões de automação remota de outros computadores.

Configurando o cliente

Quando o computador servidor é definido, você pode configurar o computador cliente local.

► para configurar o computador local para automação remota

- 1 Copie o arquivo .VBR que foi criado durante a criação do servidor OLE para a máquina cliente.
- 2 Execute CLIREG32 com o nome do arquivo .VBR. Por exemplo, se o arquivo for MyServer.VBR, execute o seguinte comando no aviso Comando:

```
CLIREG32 MYSERVER.VBR
```

- 3 Na caixa de diálogo que é aberta, digite o endereço de rede da máquina servidora e selecione o protocolo da rede (geralmente TCP/IP).

Opções da política de segurança do sistema

A tabela a seguir descreve as opções de área da Política de segurança do sistema na guia **Client Access** do **Gerenciador de conexão de automação remota**.

Nome	Valor ¹	Descrição
Proibir Todas as Criações Remotas	0	Não permite que quaisquer objetos sejam criados.
Permitir Criações Remotas por Chave	2	Um objeto pode ser criado somente se a caixa de seleção Allow Remote Activation estiver selecionada. Isso altera a sua CLSID no Registro do Windows para incluir a seguinte definição de subchave: <code>AllowRemoteActivation = Y</code>
Permitir Criações Remotas por ACL	3	Um usuário pode criar um objeto apenas se a Lista de Controle de Acesso da CLSID no Registro do Windows incluir o usuário. Somente no Windows NT.
Permitir Todas as Criações Remotas	1	Permite que qualquer objeto seja criado. Não é recomendável fora do ambiente de desenvolvimento.

¹ A coluna Valor lista a definição preferencial da RemoteActivationPolicy do **Gerenciador de automação** no Registro do Windows.

Usando autenticação na automação remota

Para Servidores OLE remotos executados em qualquer sistema operacional do Windows, o [procedimento de chamada remota](#) (RPC) fornece os seguintes níveis de autenticação.

Nome	Valor	Descrição
Padrão	0	Usa o padrão da rede.
Nenhum	1	Sem autenticação.
Conectar	2	Conexão do servidor é autenticada.
Chamar	3	Autentica apenas no início de cada chamada de procedimento remota quando o servidor recebe a solicitação. Não se aplica a seqüências de protocolo baseadas

		em conexão (as que começam com o prefixo "ncacn").
Pacote	4	Verifica se todos os dados recebidos são do cliente esperado.
Pacote Integridade	5	Verifica se nenhum dado transferido entre o cliente e o servidor foi modificado.
Pacote Privacidade	6	Verifica todos os níveis anteriores e criptografa os valores de argumento de cada chamada de procedimento remota.

A necessidade de autenticação RPC deve ser avaliada com cuidado, pois à medida que o nível de autenticação RPC aumenta, o desempenho diminui. É possível especificar um nível de autenticação para cada classe em seu Servidor OLE de modo que os níveis demorados como criptografia, não precisem ser aplicados ao componente inteiro.

Por exemplo, um serviço de dados implementado como um Servidor OLE remoto pode ter uma classe Logon usada para transmitir informações de usuário e senha e essa classe pode exigir autenticação Packet Privacy. Outras classes expostas pelo servidor podem usar um nível de autenticação muito mais baixo.

Resolvendo problemas de automação remota

Aqui estão algumas sugestões para resolver possíveis dificuldades.

Problema	Ação
Código de erro OLE 0x800706d9: Não há mais pontos finais disponíveis no gerenciador de pontos finais.	Certifique-se de que o Gerenciador de automação está sendo executado no computador servidor e que o nome do computador servidor esteja digitado corretamente na caixa Network Address (Endereço da rede) do Gerenciador de conexão de automação remota .
O Visual FoxPro.Application não aparece na lista Classes OLE do Gerenciador de automação remota .	<ol style="list-style-type: none"> 1. Execute REGEDIT.EXE para abrir o registro. 2. Exclua todas as referências ao Visual FoxPro 5. 3. Execute o Visual FoxPro com a marca de linha de comando -r: vfp.exe -r