FoxcodePlus - Novas funcionalidades ao IntelliSense do Microsoft Visual FoxPro 9

Por Rodrigo D. Bruscain – Versão Beta 3.13.2 – Last updated 26/Maio/2013

O FoxcodePlus não substitue o IntelliSense do VFP, ele interage em pontos que o IntelliSense padrão do VFP não ajuda adequadamente ou não faz absolutamente nada. A ideia do FoxcodePlus é trazer um pouco das funcionalidades do IntelliSense do Visual Studio para o VFP, ou seja, dar mais agilidade e evitar erros ao escrever os programas.

Vejo abaixo as novas funcionalidades:

1- Incremental IntelliSense para funções, comandos, etc.

*Como no Visual Studio , tudo que for digitado na tela de codificação (Edit window) o VFP fará uma pesquisa incremental de tudo que é possivel incluir no IntelliSense. A pesquisa incremental acontece sempre que uma tecla é pressionada. No "IntelliSense Manager" você pode escolher na opção "**Incrementally search**" como incremen tar o IntelliSense.*

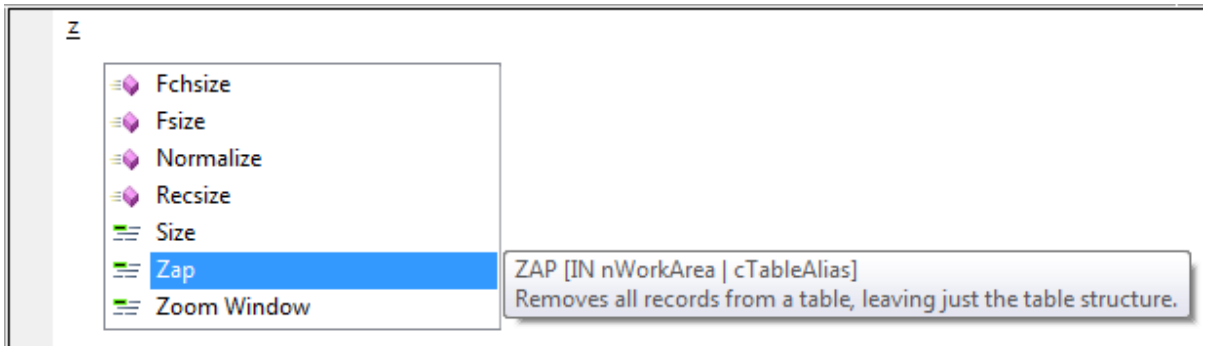


Figura 1.10

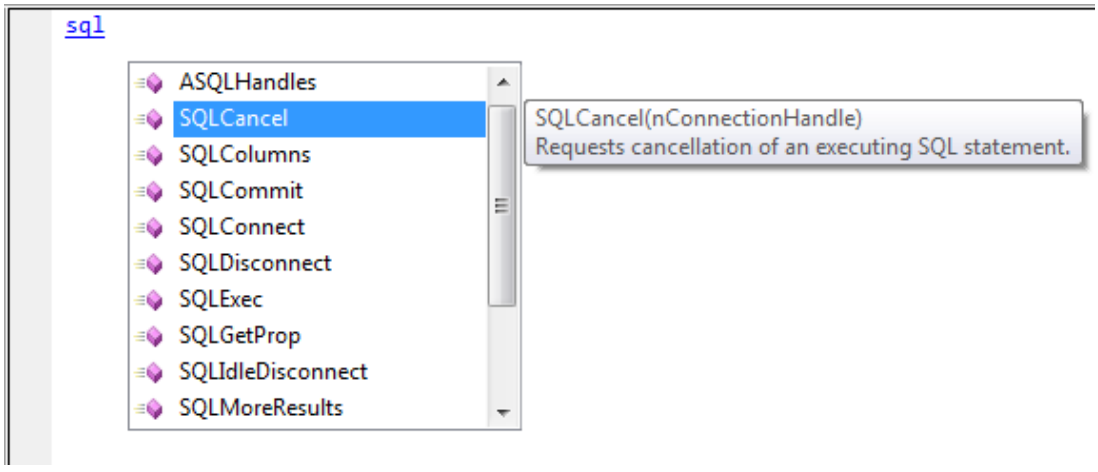


Figura 1.11

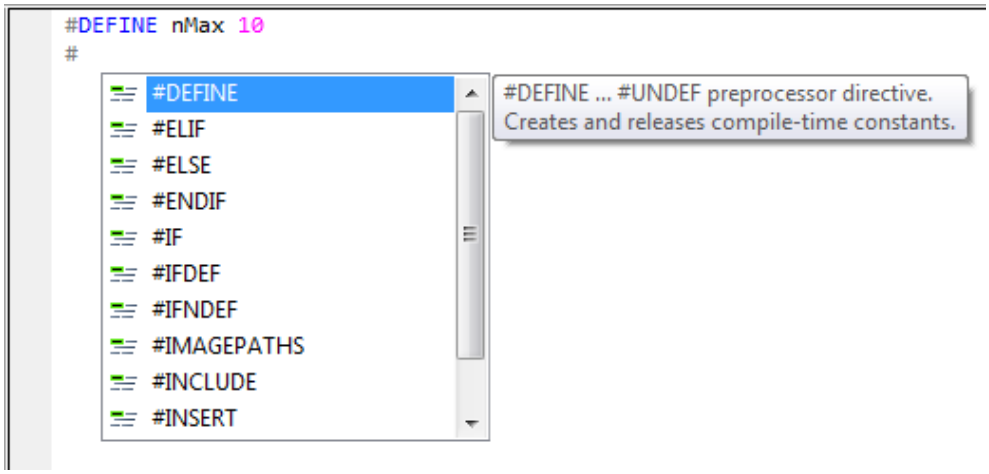


Figura 1.12

2- Variáveis em write-time

As variaveis criadas em write-time tambem podem ser acessada pelo IntelliSense. O VFP possue varias formas para criar uma variavel e todas são suportadas pelo FoxcodePlus.

```
local loRs as "ADODB.RecordSet"
local array laNames[10,2]
local lcFirstNames as String, lcAddress as Character, lnNumber as Integer

text to lcText textmerge
    My Text Here !!!
endtext

count to lnReccount
calculate avg(aa) to lnAvg
sum abc.xvalue to lnSum

lc
```

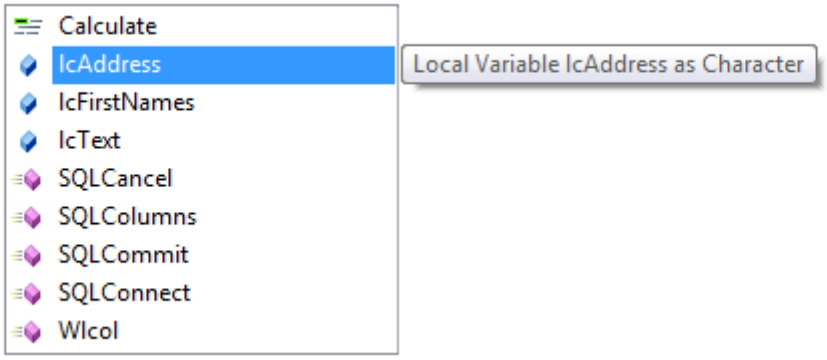


Figura 2.10

```
local loRs as "ADODB.RecordSet"
local array laNames[10,2]
local lcFirstNames as String, lcAddress as Character, lnNumber as Integer

text to lcText textmerge
    My Text Here !!!
endtext

count to lnReccount
calculate avg(aa) to lnAvg
sum abc.xvalue to lnSum

ln
```

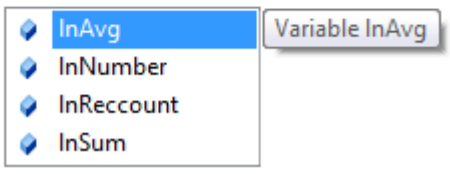


Figura 2.11

3- Acessando a lista de variáveis em write-time.

Caso deseje saber todas as variáveis criadas até onde o cursor esteja posicionado, digite "m." , dessa forma o IntelliSense será aberto de modo "não incremental".

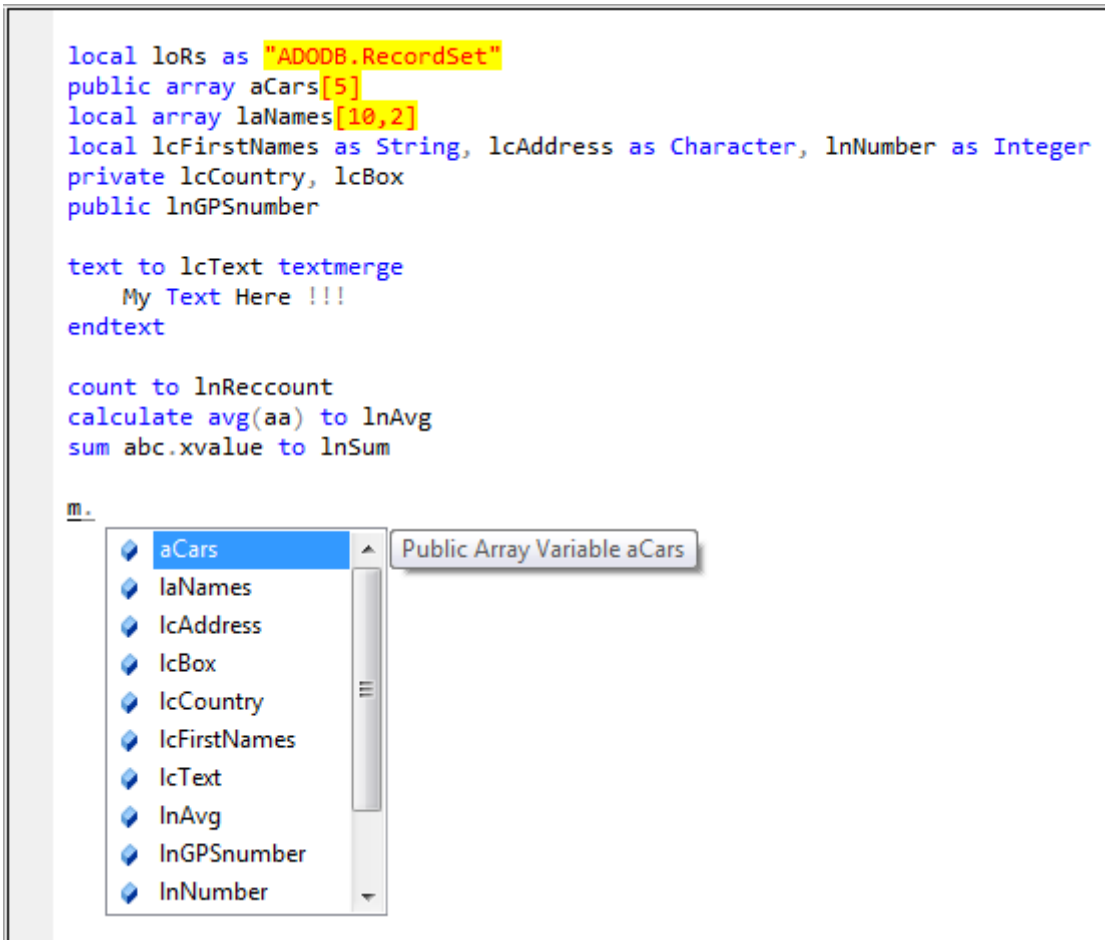


Figura 3.10

4- Constantes em write-time

As constantes criadas também podem ser acessadas pelo IntelliSense. O tooltip apresenta o valor da contante.

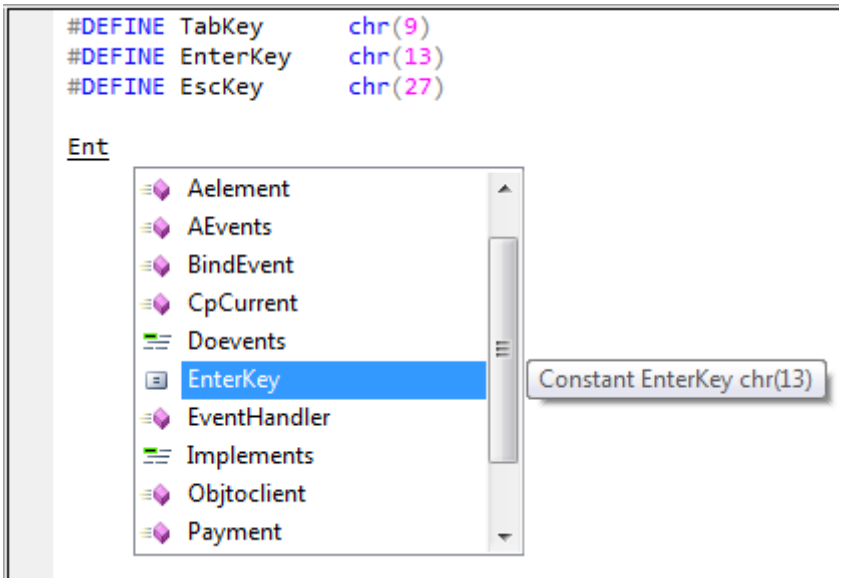


Figura 4.10

As constantes armazendas em arquivos e incluidas atraves do #INCLUDE tambem são reconhecidas pelo IntelliSense. O exemplo na figura abaixo demonstra na pratica essa nova funcionalidade do IntelliSense.

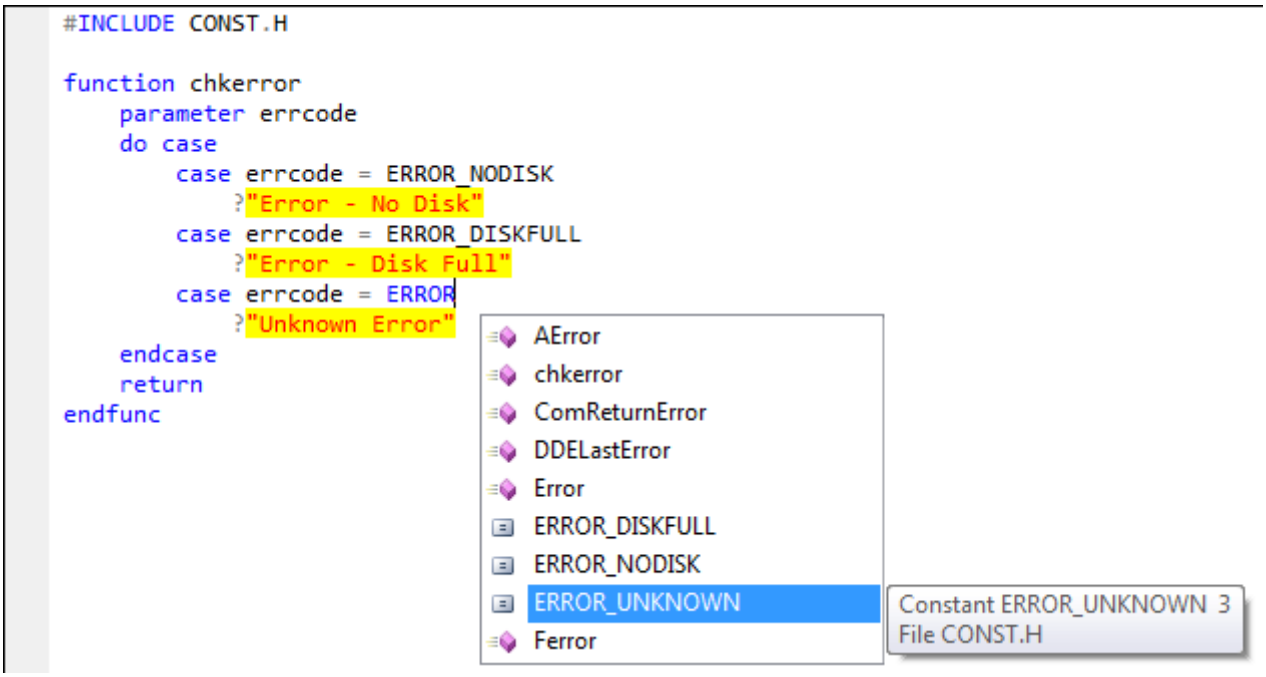


Figura 4.12

5- Tabelas em write-time e run-time

As tabelas que são criadas e/ou abertas conforme imagem abaixo, também são incluídas no IntelliSense. O tooltip indica o modo que a tabela foi aberta.

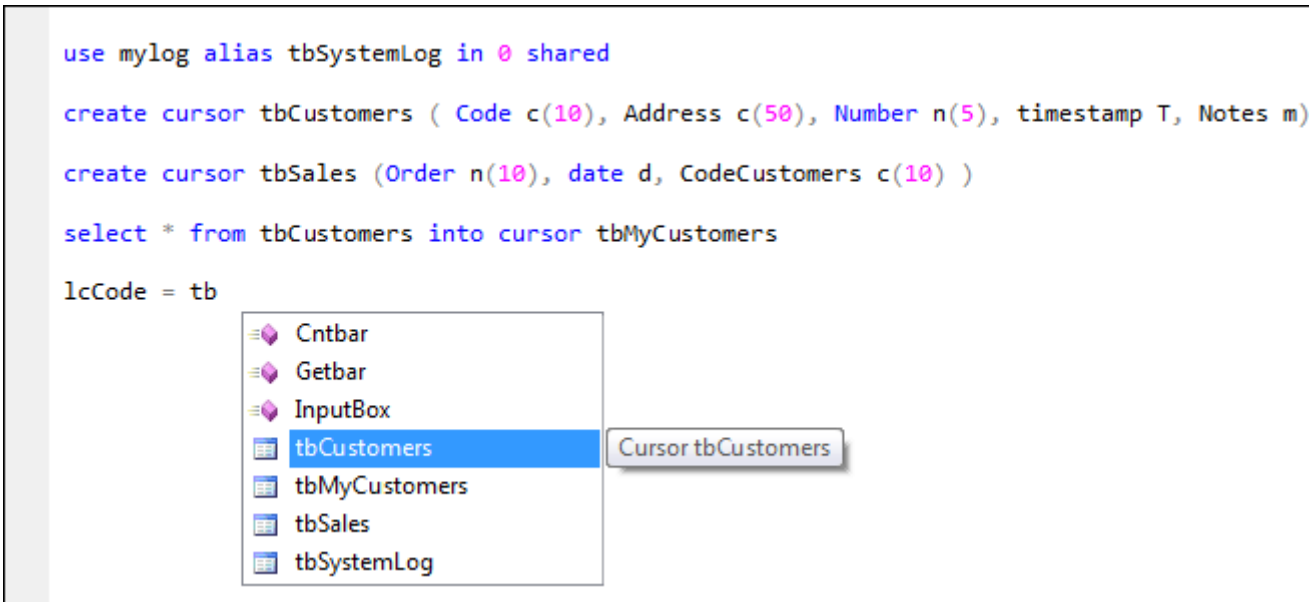


Figura 5.10

IntelliSense para tabelas do DataEnvironment de Forms (VFP tables)

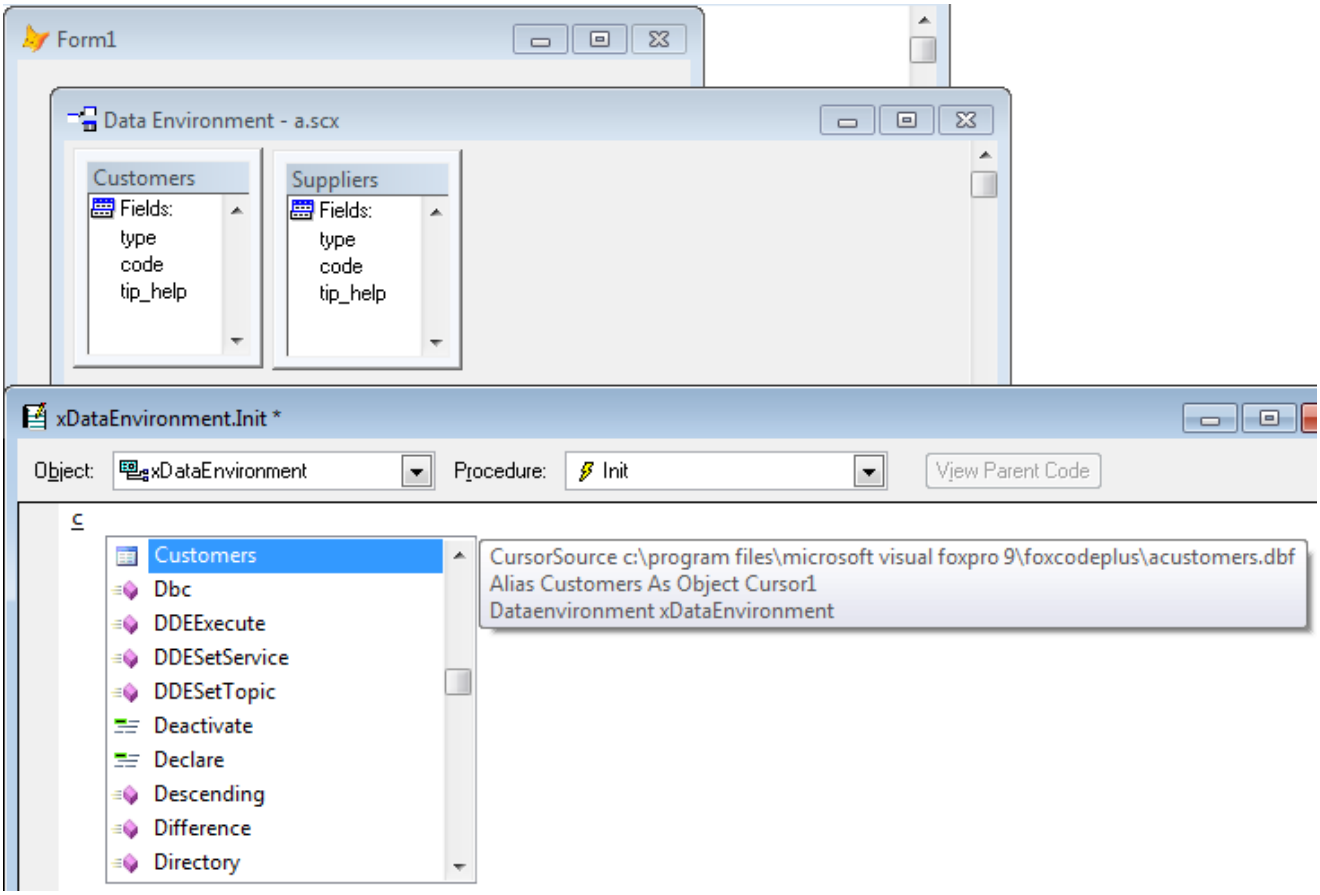


Figura 5.11

IntelliSense para tabelas do DataEnvironment de Reports (VFP tables)

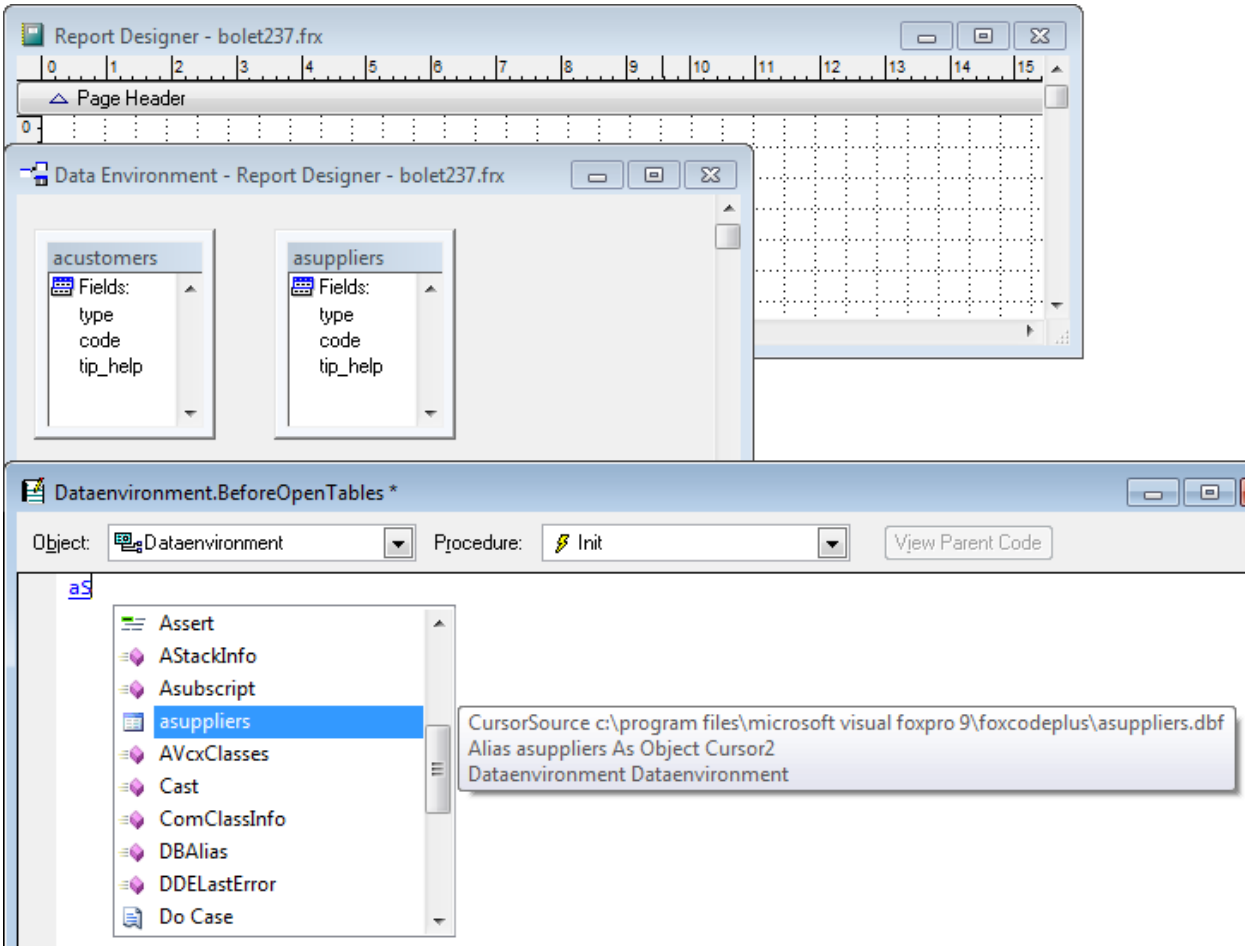


Figura 5.12

6- Campos em write-time e run-time

Os campos das tabelas também são incluídos no IntelliSense. O tipo do campo é apresentado no tooltip.

OBS: Campos de tabelas criadas pelo "select – SQL command" não são suportados.

```
use mylog alias tbSystemLog in 0 shared

create cursor tbCustomers ( Code c(10), Address c(50), Number n(5), timestamp T, Notes m)

create cursor tbSales (Order n(10), date d, CodeCustomers c(10) )

select * from tbCustomers into cursor tbMyCustomers

lcCode = tbCustomers.
```

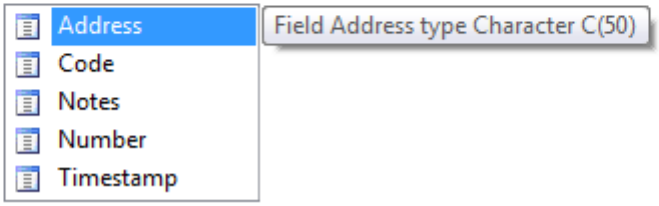


Figura 6.10

7- Selecionando uma tabela com o comando "Select" ou todos os comandos com a clausula "IN"

```
use mylog alias tbSystemLog in 0 shared

create cursor tbCustomers ( Code c(10), Address c(50), Number n(5), timestamp T, Notes m)

create cursor tbSales (Order n(10), date d, CodeCustomers c(10) )

select * from tbCustomers into cursor tbMyCustomers

select
```

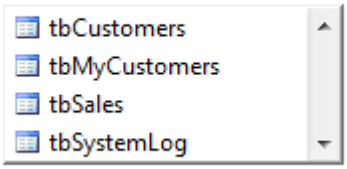


Figura 7.10

Todos os comandos de manipulação de dados em tabelas que contem a clausula "IN" interage diretamente com o IntelliSense, assim todas a tabelas criadas e/ou abertas no código são apresentadas

Segue abaixo a lista de comandos suportados com a clausula "IN"

– Append	– Recall
– Replace	– Seek
– Blank	– Select
– Calculate	– Set Filter
– Delete	– Set Order To
– Display	– Set Relation
– Flush	– Skip
– Go | Goto	– Unlock
– List	– Zap

```
use mylog alias tbSystemLog in 0 shared

create cursor tbCustomers ( Code c(10), Address c(50), Number n(5), timestamp T, Notes m)

create cursor tbSales (Order n(10), date d, CodeCustomers c(10) )

select * from tbCustomers into cursor tbMyCustomers

replace code with "XX100" in
```

tbCustomers
tbMyCustomers
tbSales
tbSystemLog

Figura 7.11

8- APIs em write-time e run-time

```
declare integer GetWindow in Win32API as apiGetWindow integer hwnd, integer nType
declare integer GetWindow in Win32API integer hwnd, integer nType, string

ap
```

apiGetWindow
Append
Application
APrinters
AProcInfo
Capslock
GetCursorAdapter
Zap

apiGetWindow(integer hwnd , integer nType)
DLL Function GetWindow in Win32API as apiGetWindow

Figura 8.10

```
declare integer GetWindow in Win32API as apiGetWindow integer hwnd, integer nType
declare integer GetWindow in Win32API integer hwnd, integer nType, string

GetW
```

apiGetWindow
GetWindow
GetWordCount
GetWordNum

GetWindow(integer hwnd , integer nType , string)
DLL Function GetWindow in Win32API

Figura 8.12

9- Funções e Procedures em write-time

Funcões e Procedures criadas dentro do PRG corrente.

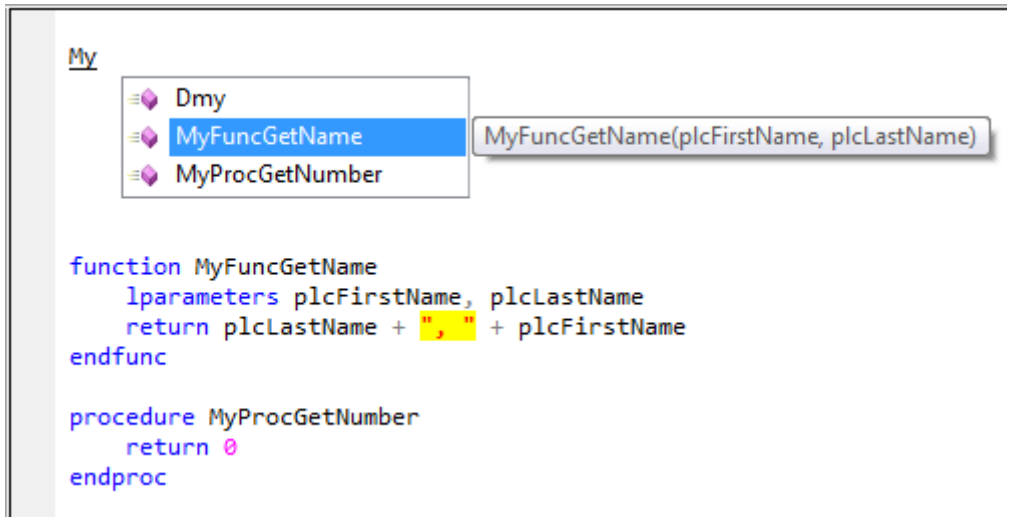


Figura 9.10

Funcões e Procedures criadas em outros PRGs invacodos pelo SET PROCEDURE TO...

SET PROCEDURE TO … em memoria também é considerado.

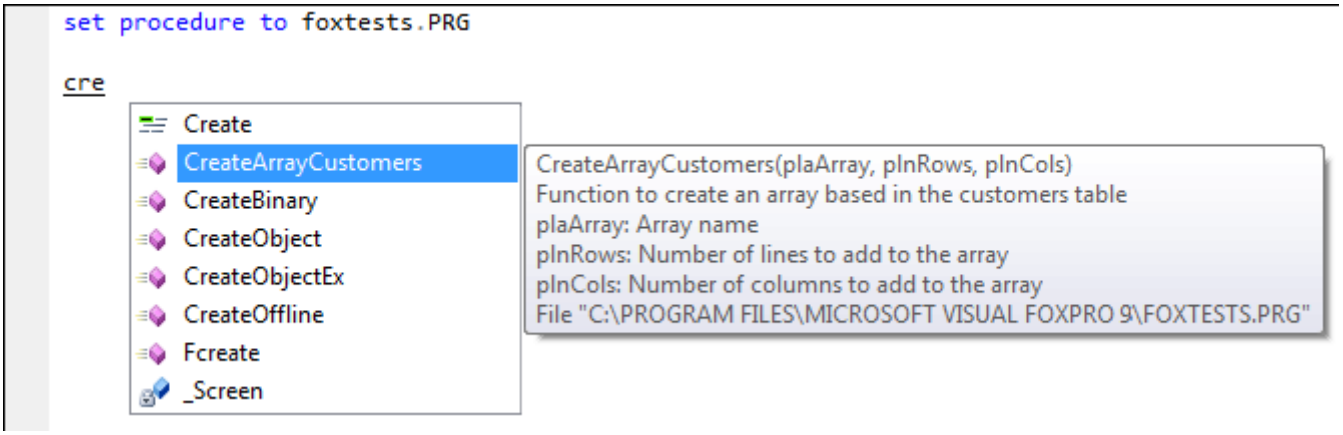


Figura 9.11

10- Classes em write-time

Classes criadas dentro do PRG corrente.

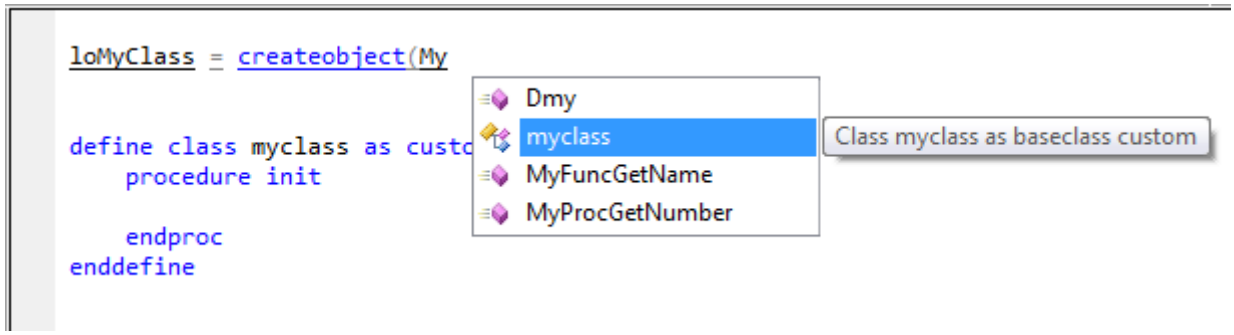


Figura 10.10

Classes criadas em outros PRGs invacodos pelo SET PROCEDURE TO...

SET PROCEDURE TO … em memoria também é considerado.

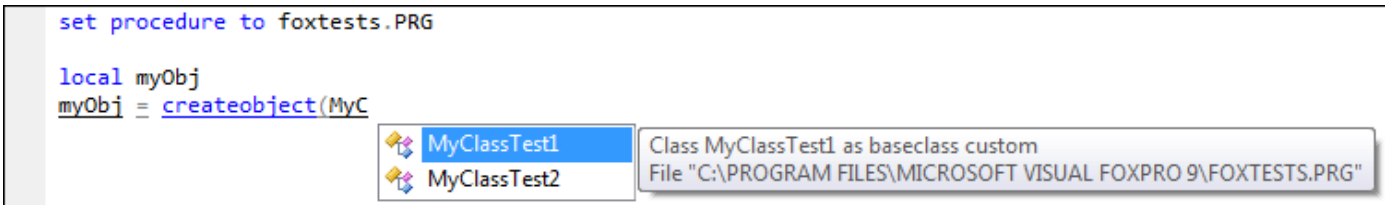


Figura 10.11

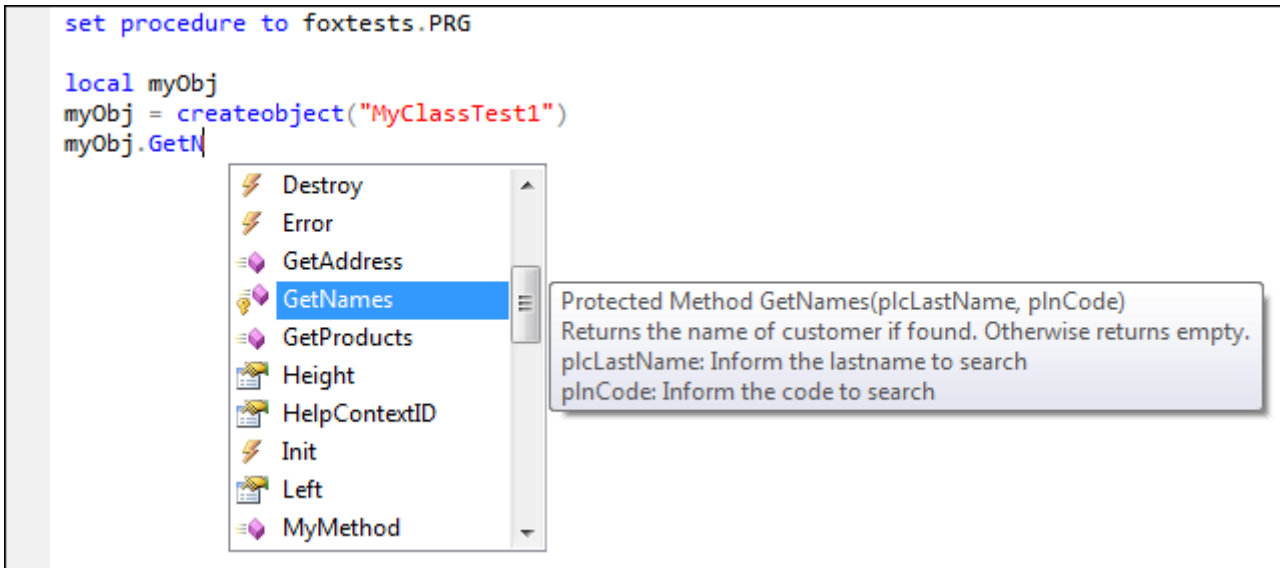


Figura 10.12

Classes criadas em arquivos VCX invacodos pelo SET CLASSLIB TO...

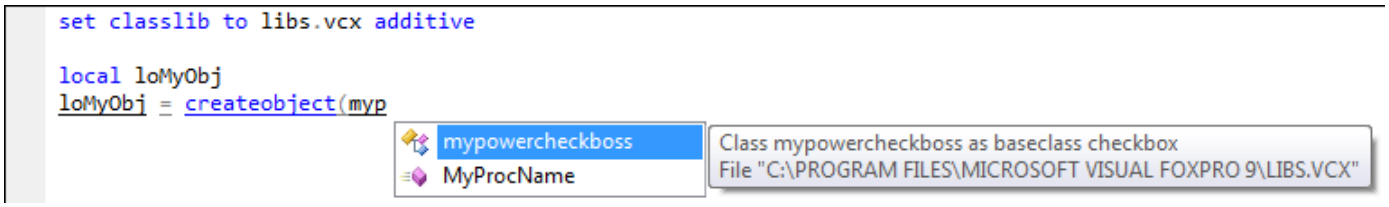


Figura 10.13

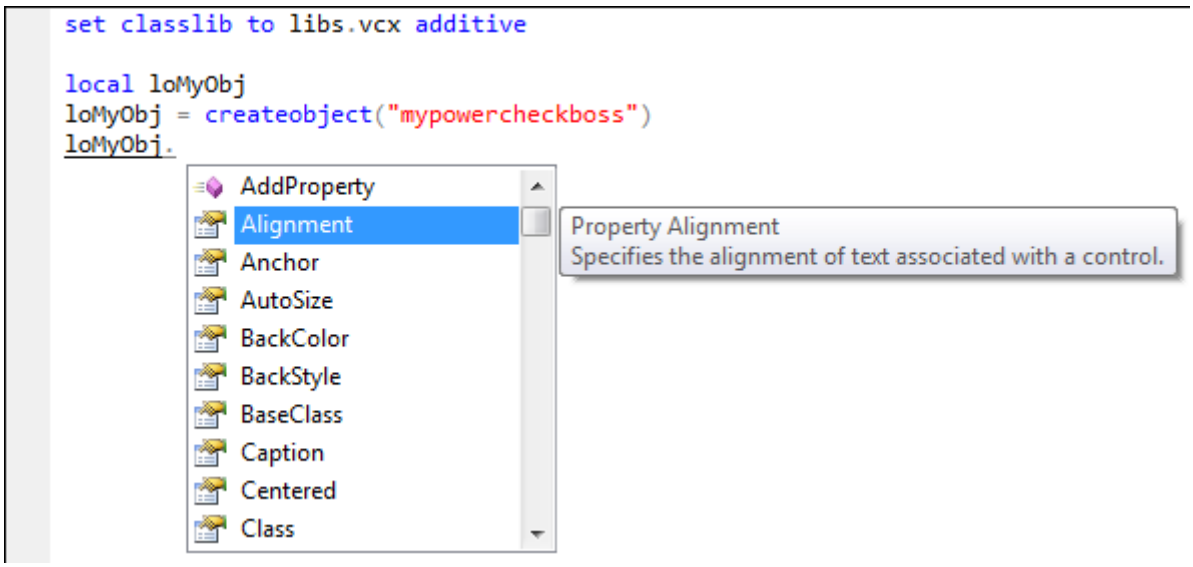


Figura 10.14

11- Propriedades em write-time

Alem das propriedades da classe herdada, as propriedades incluídas a classe também são incluídas no IntelliSense. Para acessar as propriedades utilizamos "this.", neste caso o IntelliSense abre de modo não incremental.

OBS: Caso a classe herdada não seja uma classe padrão do VFP ou uma ActiveX registrada, serão apresentados somente as propriedades e métodos adicionados a classe. Caso a classe herdada esteja em um outro programa as propriedades e métodos herdados não são considerados pelo IntelliSense. (por enquanto).

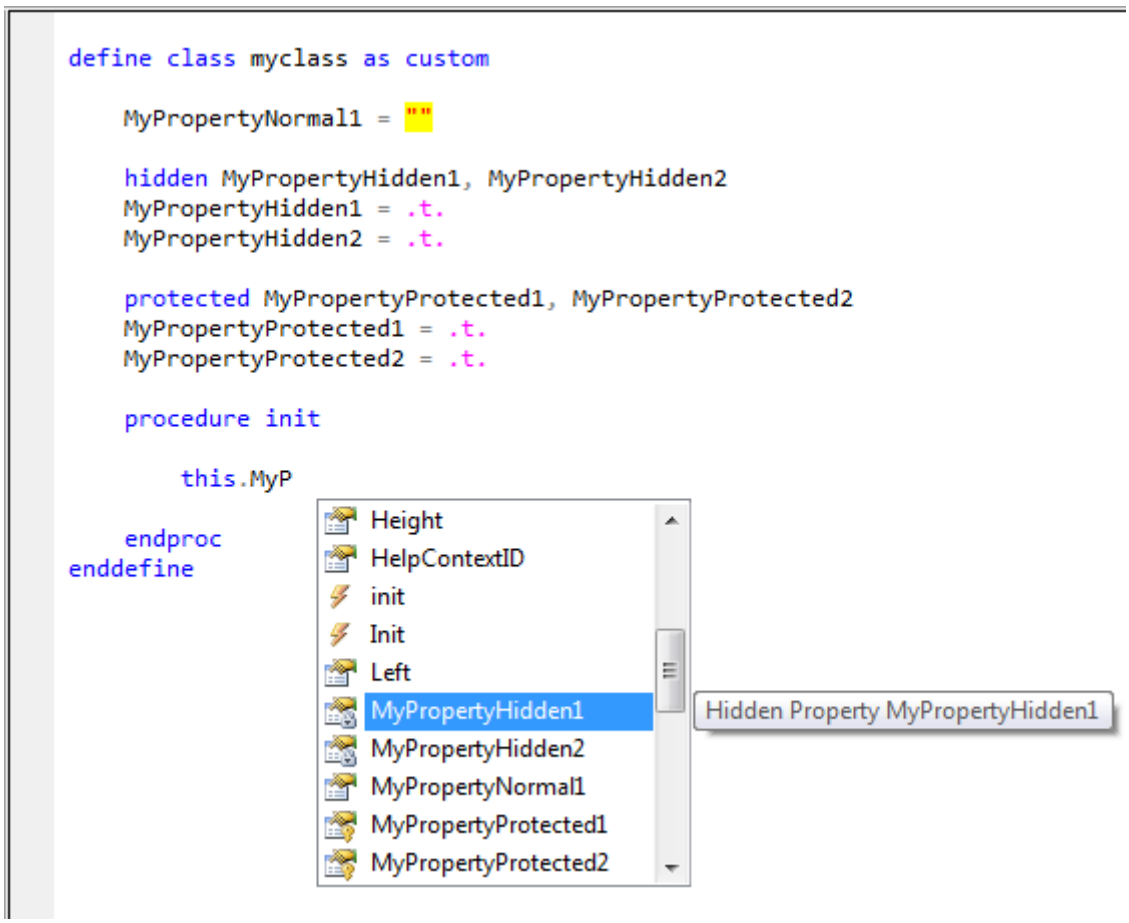


Figura 11.10

12- Métodos e Eventos em write-time

Alem dos métodos e eventos da classe padrão herdada, os novos métodos incluídos a classe também são incluídos no IntelliSense. Para acessar as métodos e eventos utilizamos "this.".

OBS: Caso a classe herdada não seja uma classe padrão do VFP ou uma ActiveX registrada, serão apresentados somente os métodos e eventos e métodos adicionados a classe. Caso a classe herdada esteja em um outro programa as propriedades e métodos herdados não são considerados pelo IntelliSense. (por enquanto).

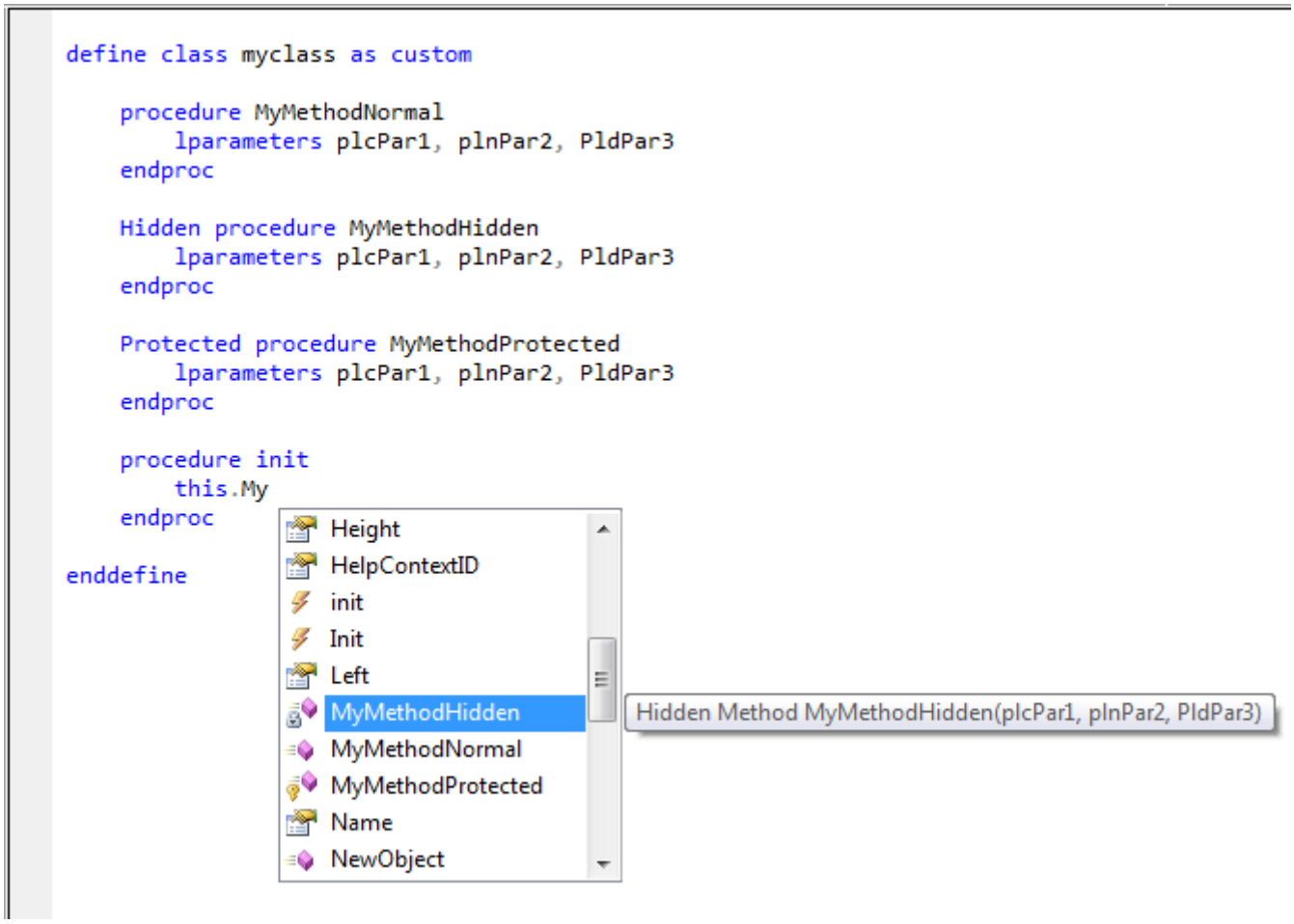


Figura 12.10

13- Summary Tooltip para funções, procedures, métodos e eventos.

*Pressionando três vezes asterisco "***" na linha acima onde a função, procedure, método ou evento é criado, o bloco do summary é inserido automaticamente. O summary é uma forma de documentar o programa de modo padronizado, alem de fornecer um tooltip personalizado.*

```
define class myclass as custom

    *** <summary>
    *** You can type here what you want.
    *** </summary>
    *** <param name="plcPar1">This is my parameter number 1</param>
    *** <param name="plnPar2">This is my parameter number 2</param>
    *** <param name="PldPar3">This is my parameter number 3</param>
    *** <remarks></remarks>
    Protected procedure MyMethodProtected
        lparameters plcPar1, plnPar2, PldPar3
    endproc

    procedure init
        this.My
    endproc

enddefine
```

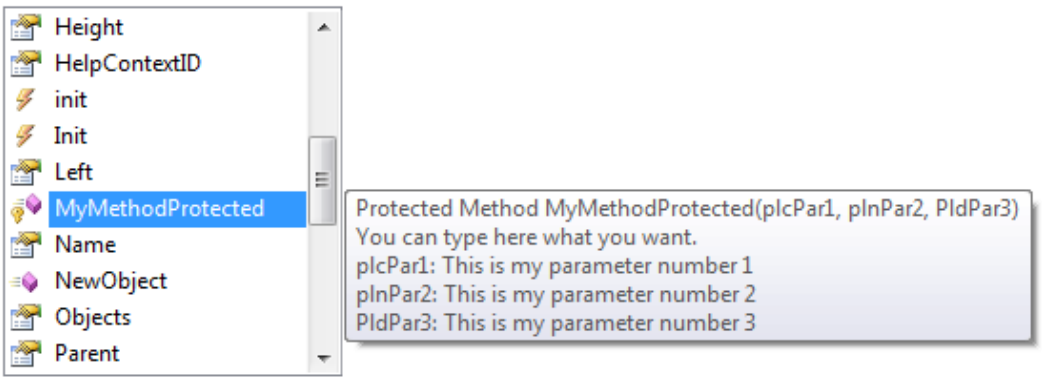


Figura 13.10

14- Objetos de classes em write-time

```
define class myclass as form

    add object xprog1 as custom
    add object xprog2 as custom
    add object xCmd1 as commandbuttom
    add object xCmd2 as commandbuttom
    add object xoleProgressBar as olecontrol with oleclass = "COMCTL.ProgCtrl.1"

    procedure init
        this.xC
    endproc

enddefine
```

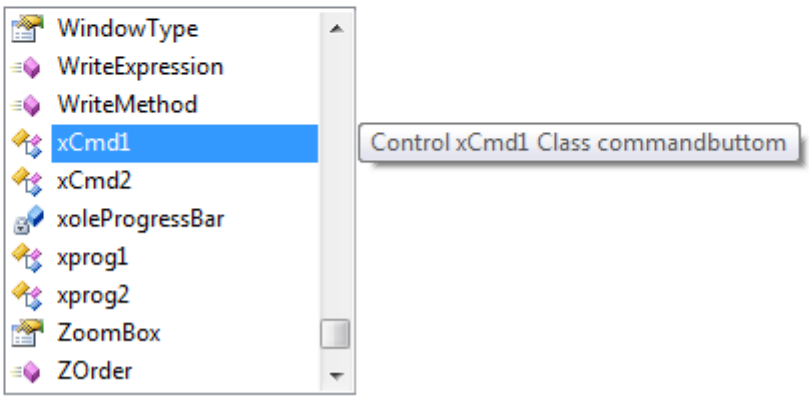


Figura 14.10

15- With...Endwith com aninhamento infinito para qualquer classe ou objeto instanciado em write-time e run-time

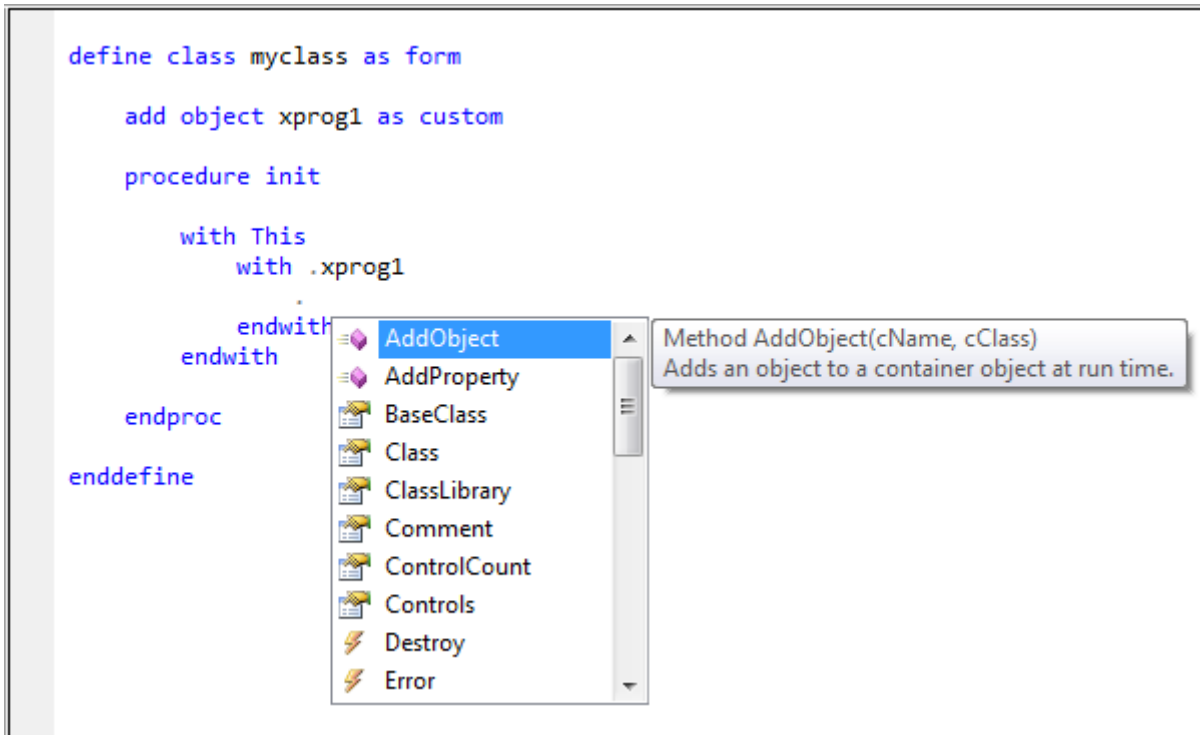


Figura 15.10

16- Objetos instanciados em memória

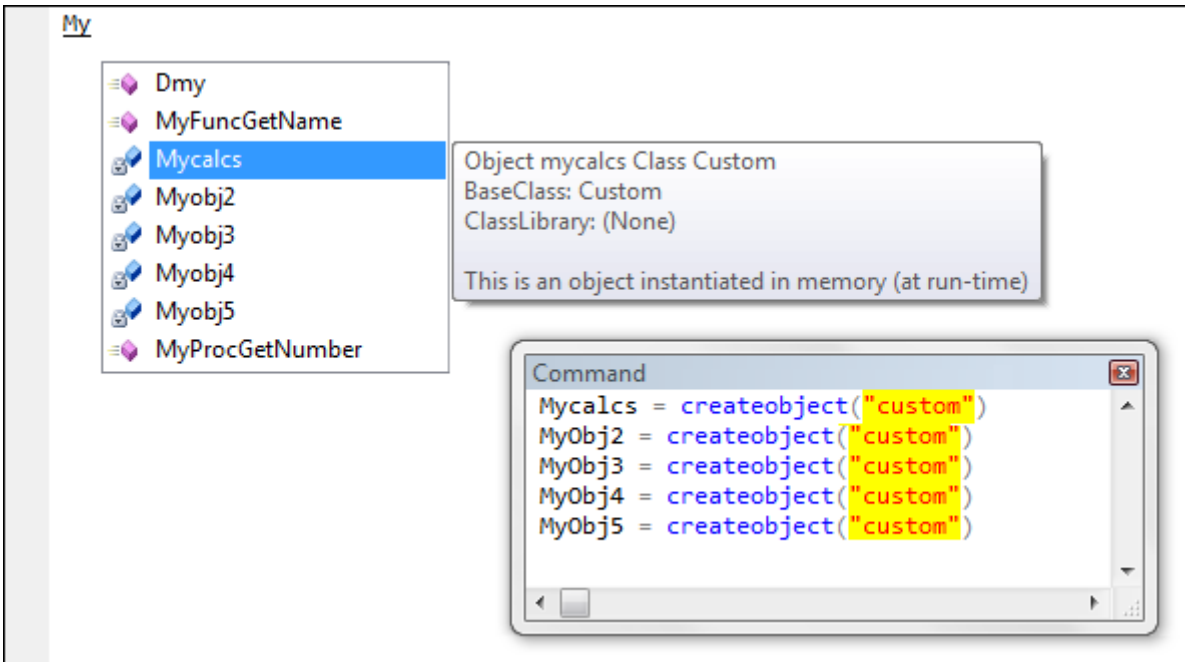


Figura 16.10

17- Atalho incremental aos controles contidos no form ou class designer.

Digite diretamente o nome do controle inserido sem a necessidade de digitar "this." ou "thisform.".
Quando selecionado o item no IntelliSense, "this." ou "thisform." é inserido automaticamente.
Observe que para melhor identificação, o caption do controle é apresentado no tooltip do IntelliSense, assim como as propriedades BaseClass e ClassLibrary.

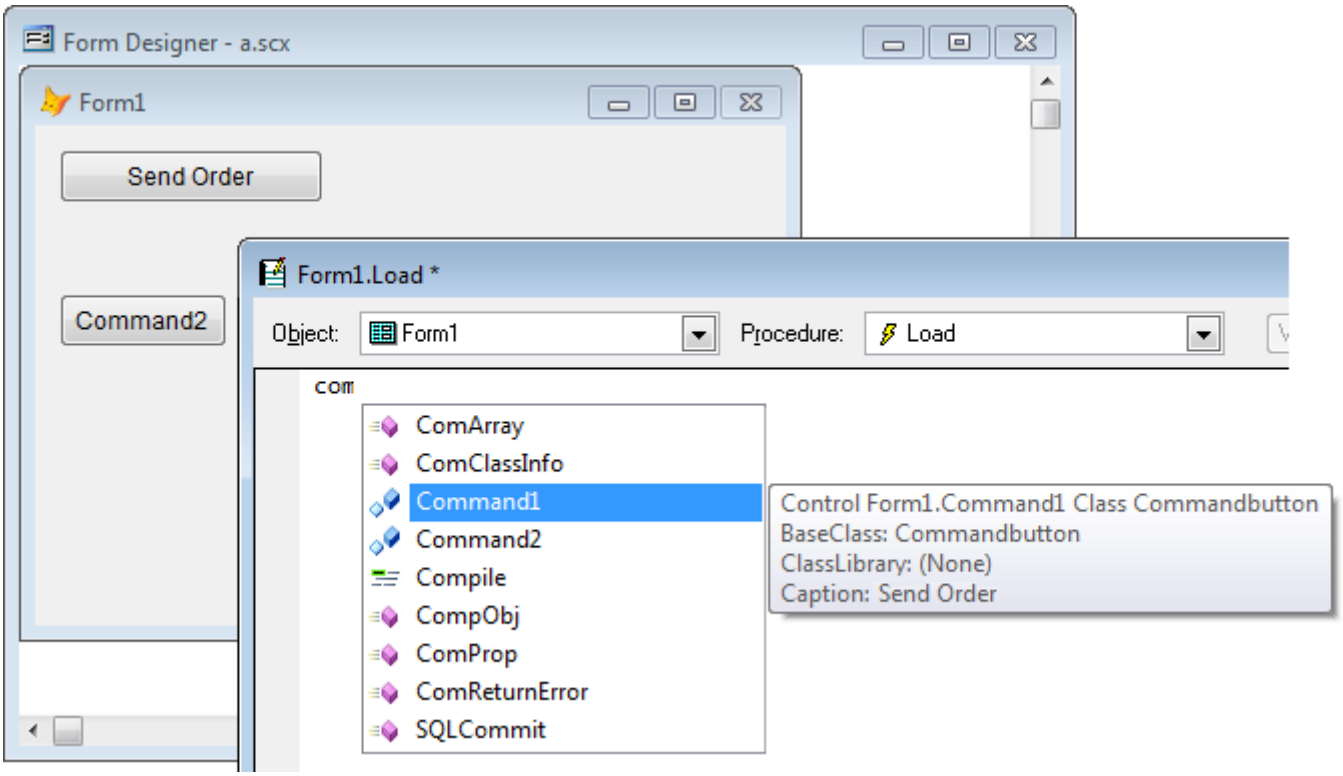


Figura 17.10

No tooltip o nome do controle é precedido do nome do objeto parent. Exemplo: ***form1****.Mypowercheckboss1*

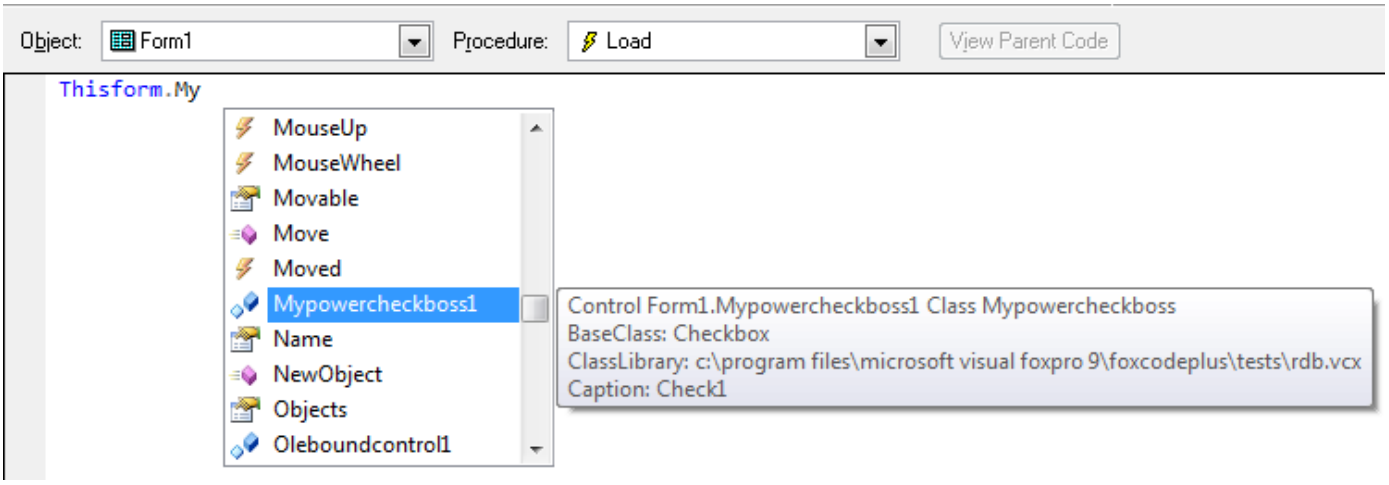


Figura 17.11

18- Substituição do IntelliSense nativo no form ou class designer.

*Se a opção "**Manage Controls at design-time**" estiver marcada o FoxcodePlus ira substituir o IntelliSense nativo no form ou class designer. Uma vez substituido, o FoxcodePlus passa a interagir diretamente e prover mais funcionalidades.*

19- Novos IntelliSense para alguns comandos

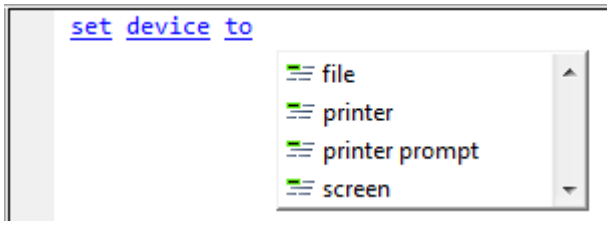


Figura 19.10

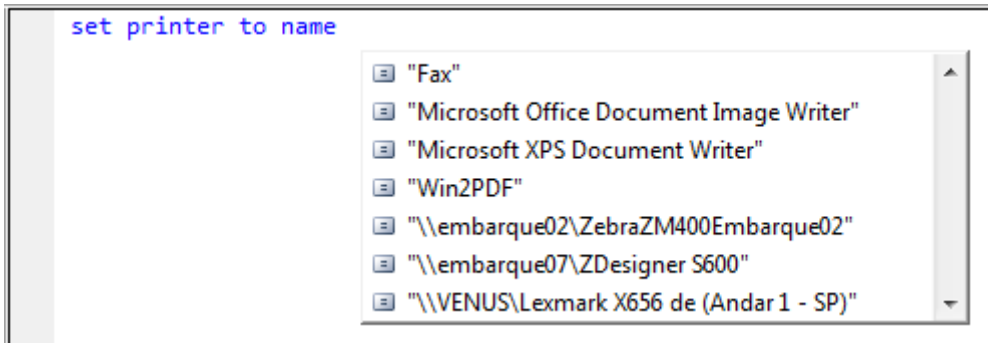


Figura 19.11

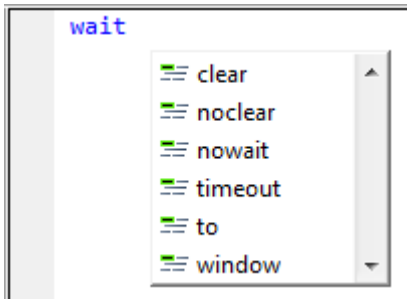


Figura 19.12

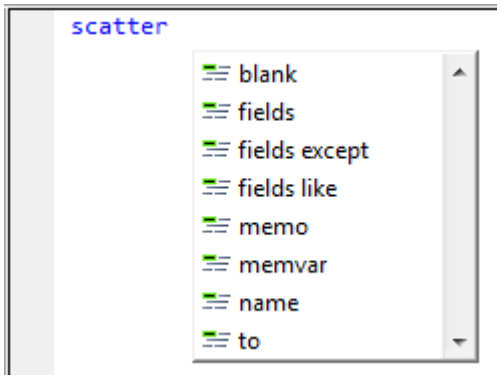


Figura 19.13

Figura 19.14

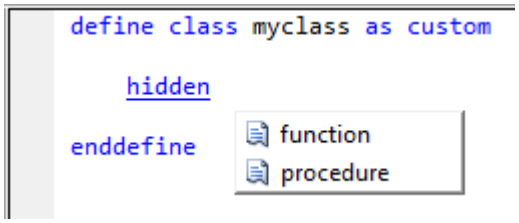


Figura 19.15

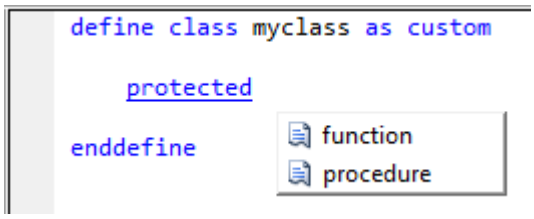


Figura 19.16

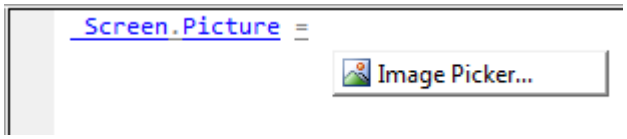


Figura 19.17

Figura 19.18

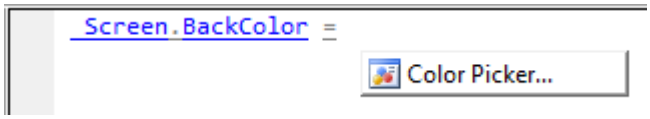


Figura 19.19

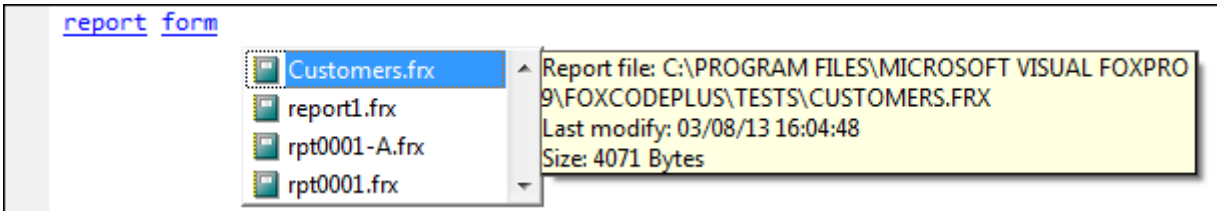


Figura 19.20

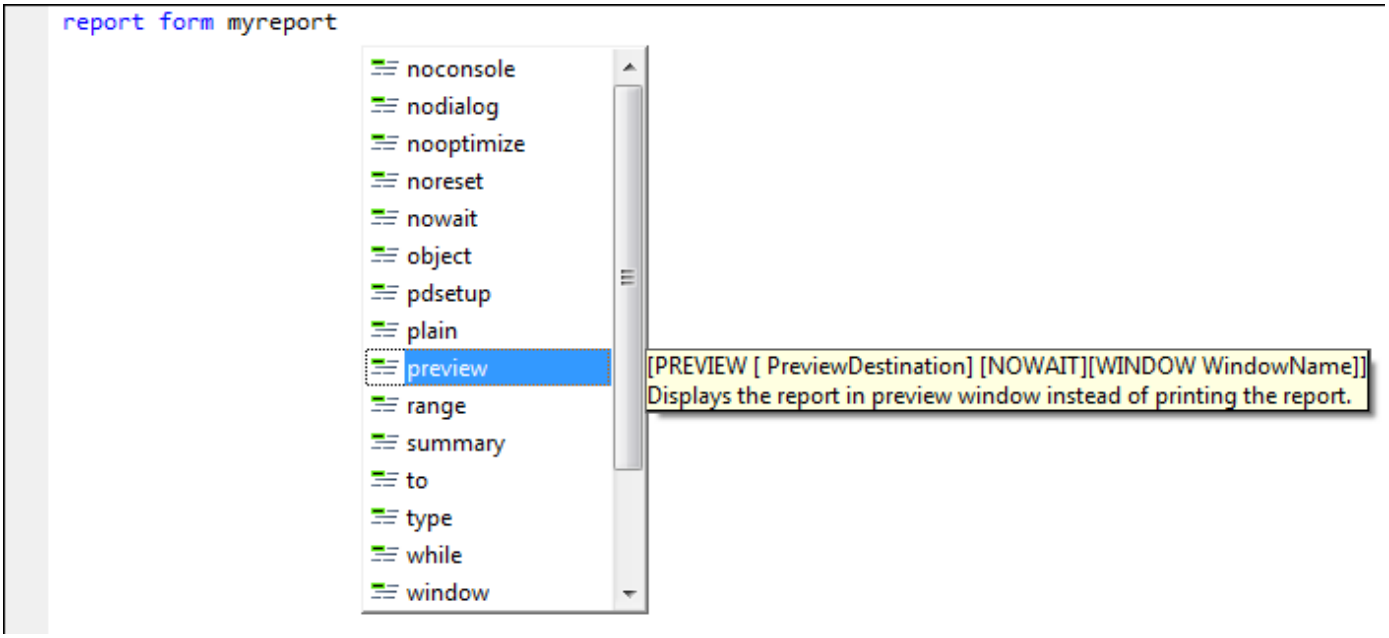


Figura 19.21

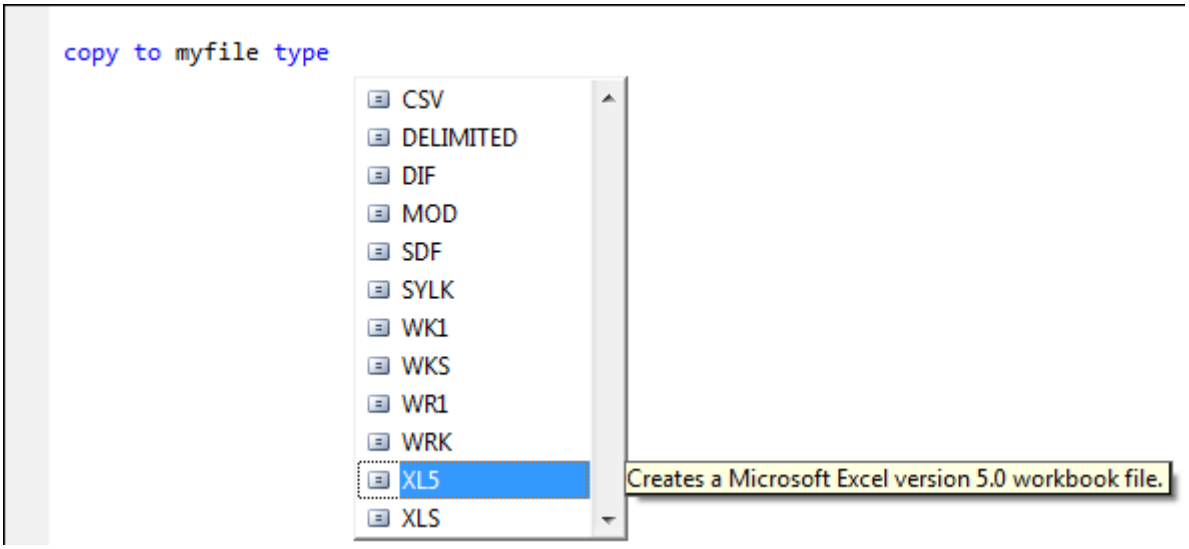


Figura 19.22

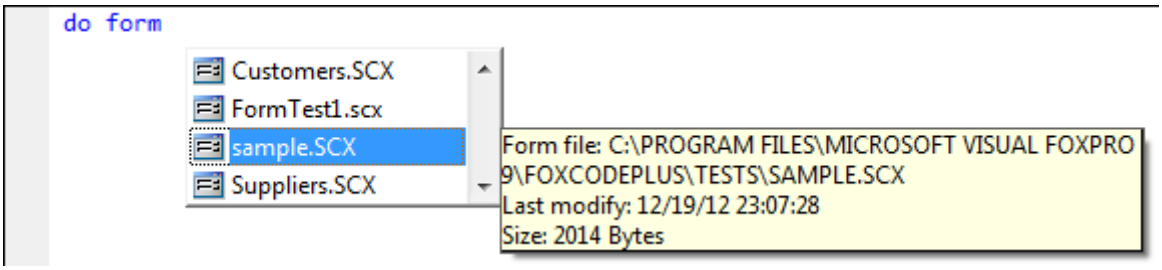


Figura 19.23

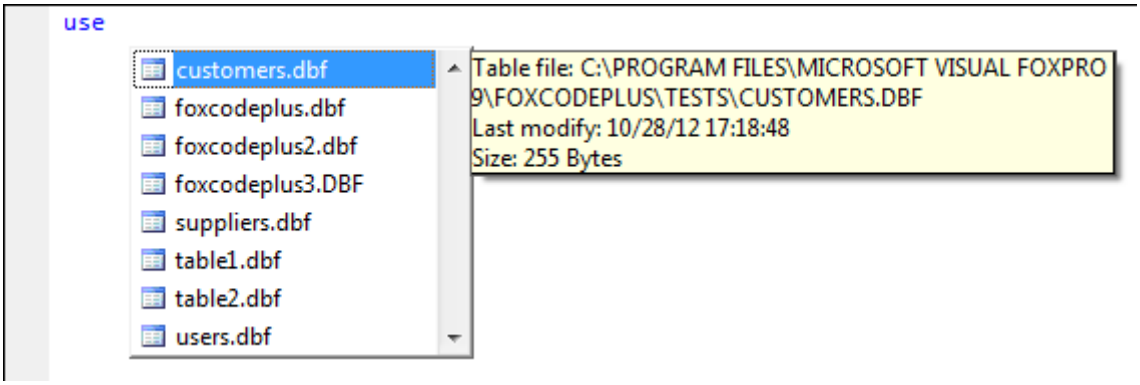


Figura 19.24

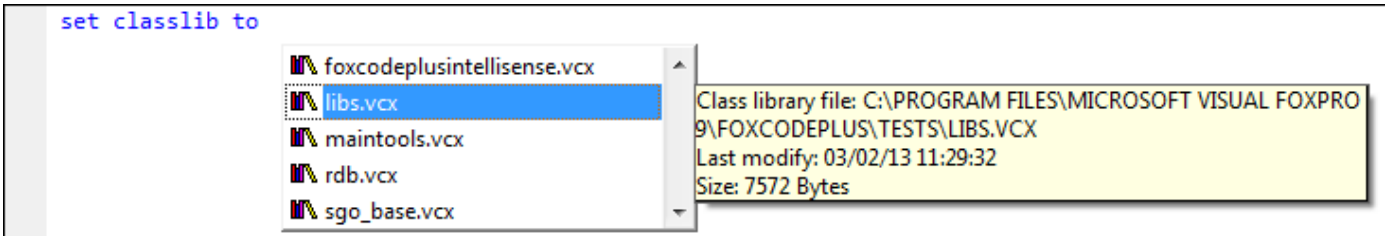


Figura 19.25

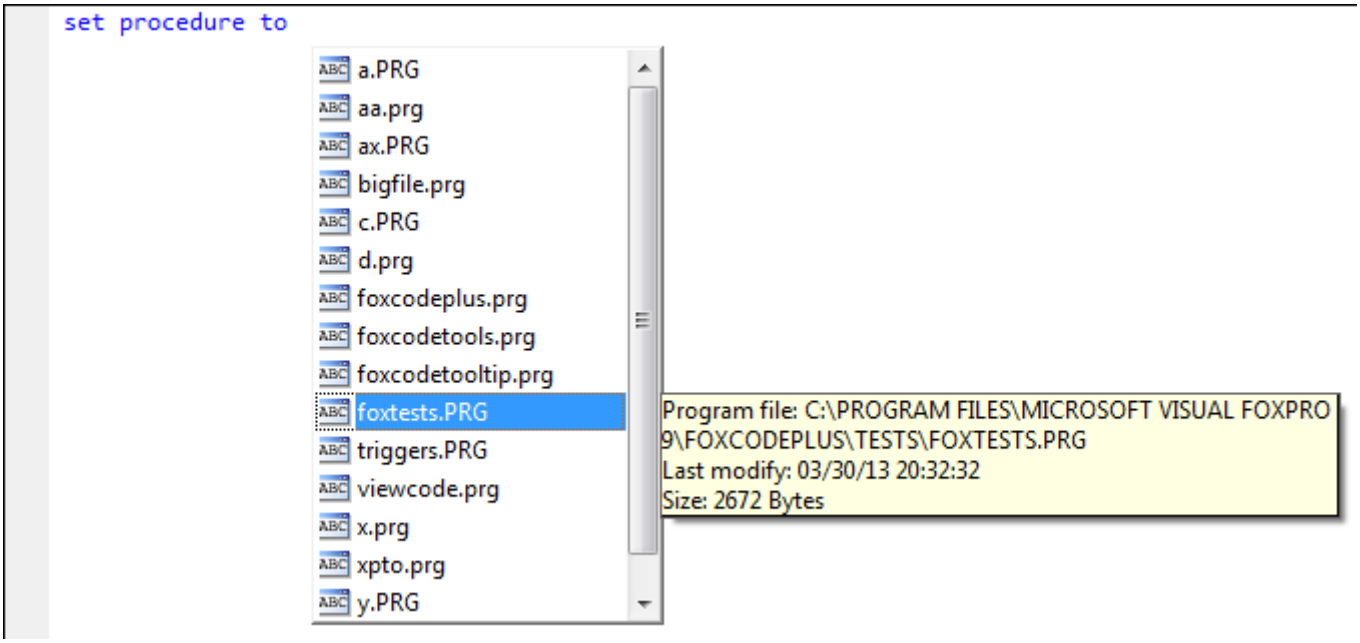


Figura 19.26

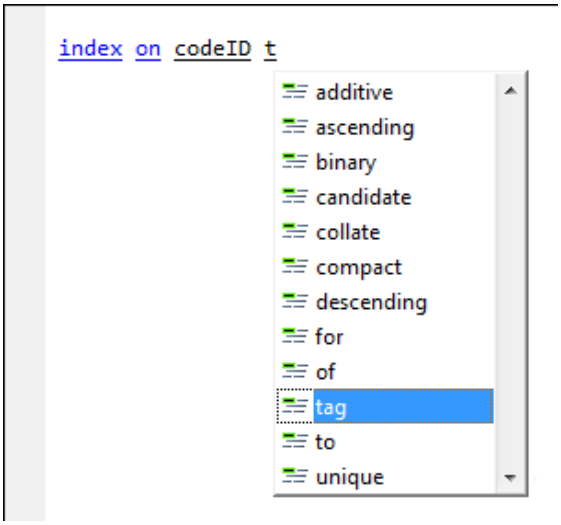


Figura 19.27

20- Code Snippets

*Como no Visual Studio, os Code Snippets nativos ou customizados são incluídos no IntelliSense para que possam ser selecionados. Para que apareçam no IntelliSense a opção "**Show code snippets**" no IntelliSense Manager deve estar marcada.*

A lista de Code Snippets pode ser consultada no IntelliSense Manager:

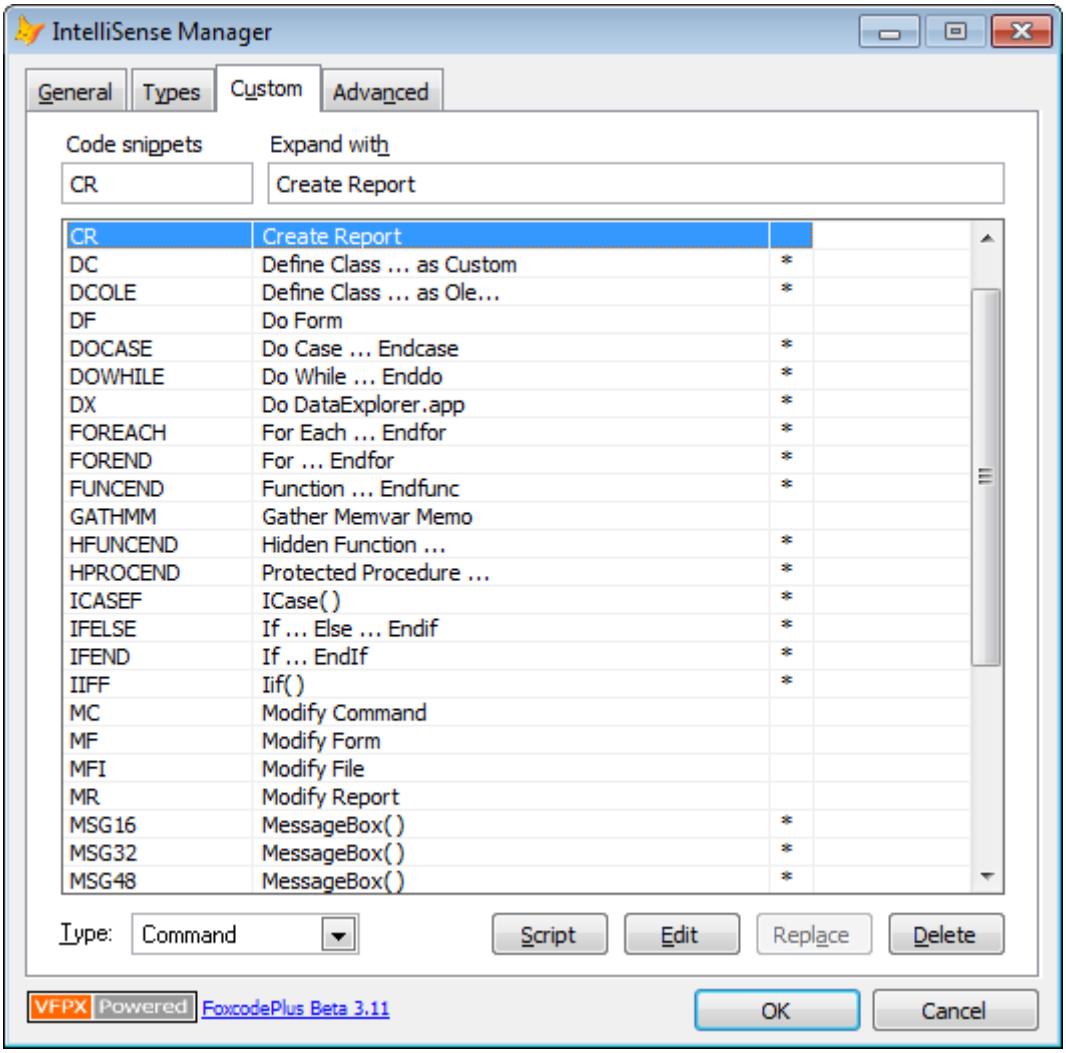


Figura 20.10

Selecionando o item no IntelliSense com a tecla SPACE, um bloco de código é inserido para reduzir a quantidade de código digitado. Ou, pode-se pressionar SPACE ao lado do texto digitado que á a forma padrão do Visual FoxPro acionar o code snippet.

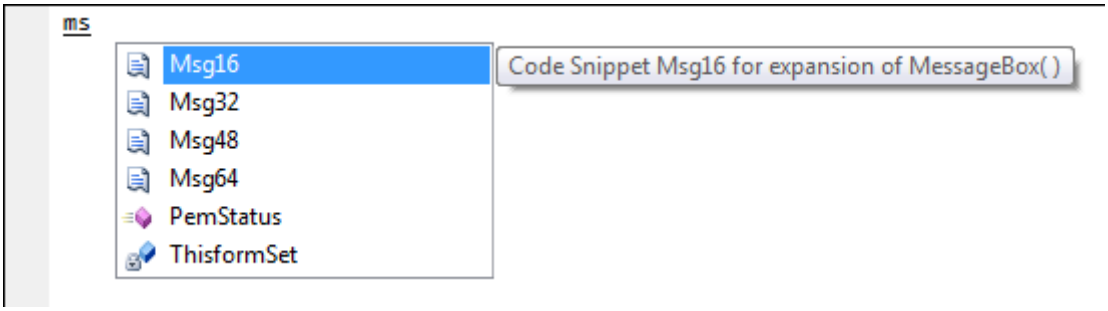


Figura 20.11

```
messagebox("mymessage",16,"error")
```

21- IntelliSense em write-time para objetos criados com as funções CreateObject(), CreateObjectEx() e NewObject()

Figura 21.10

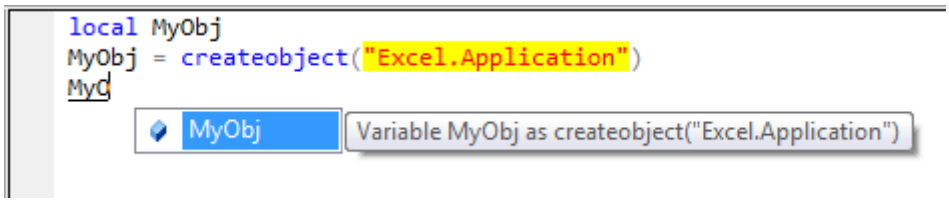


Figura 21.11

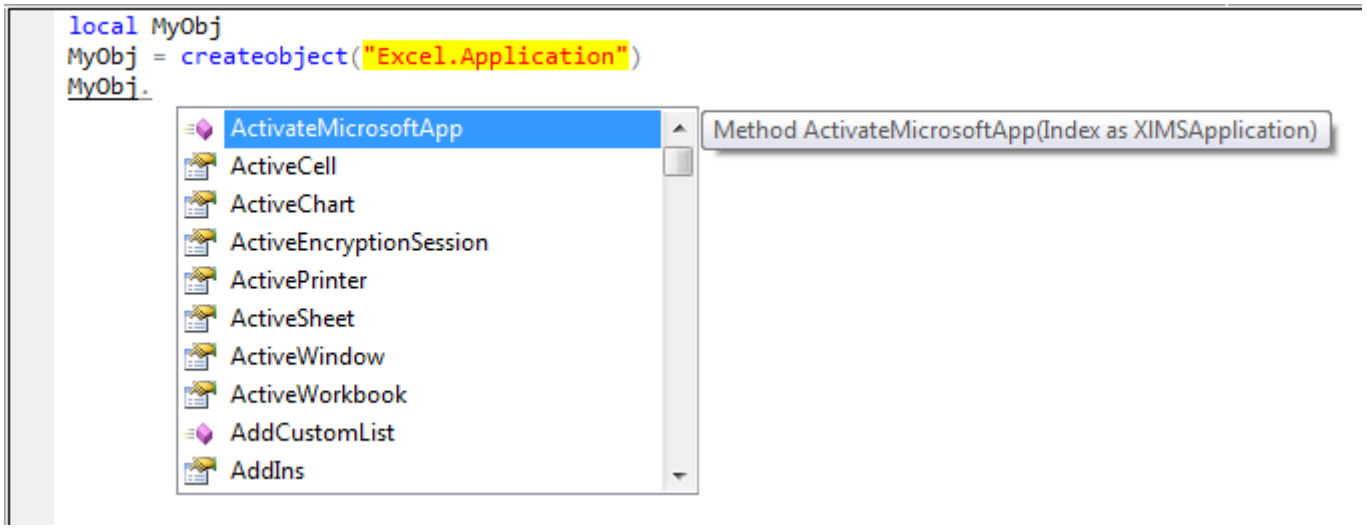


Figura 21.12

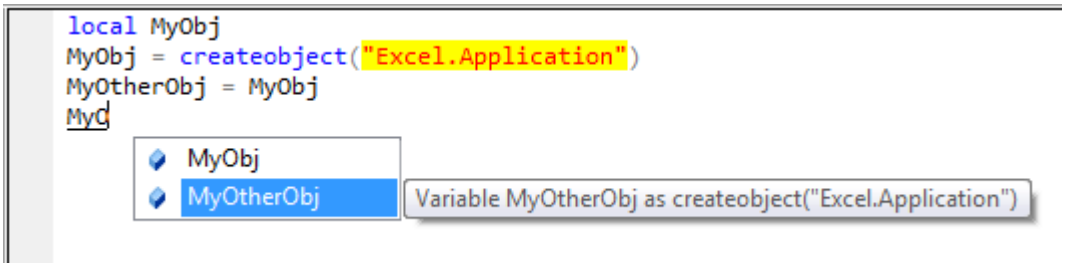


Figura 21.13

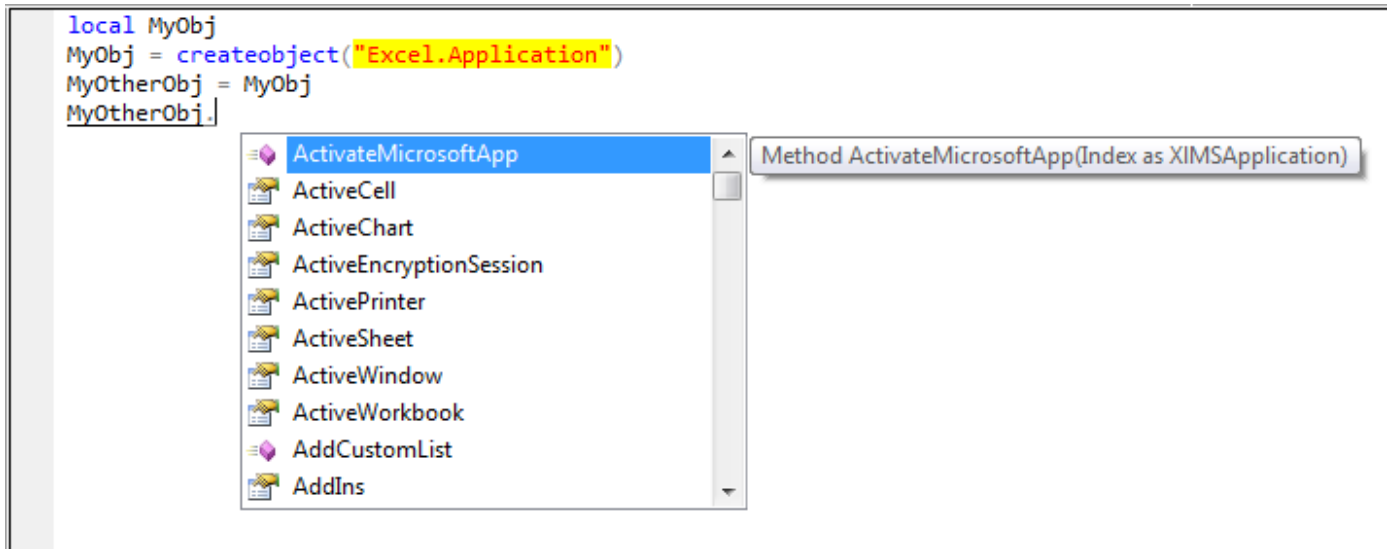


Figura 21.14

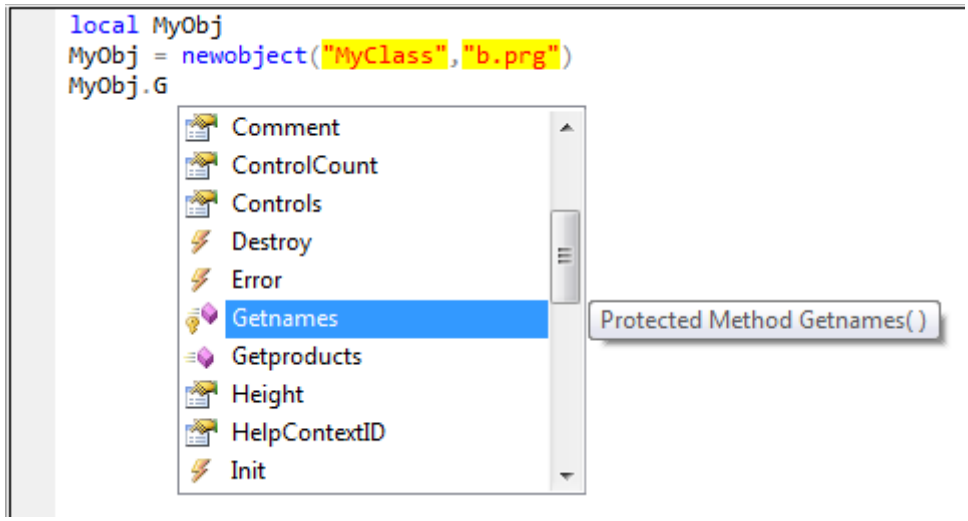


Figura 21.15

Suporte para _MemberData property indicando qual a propriedade que teve capitalization.

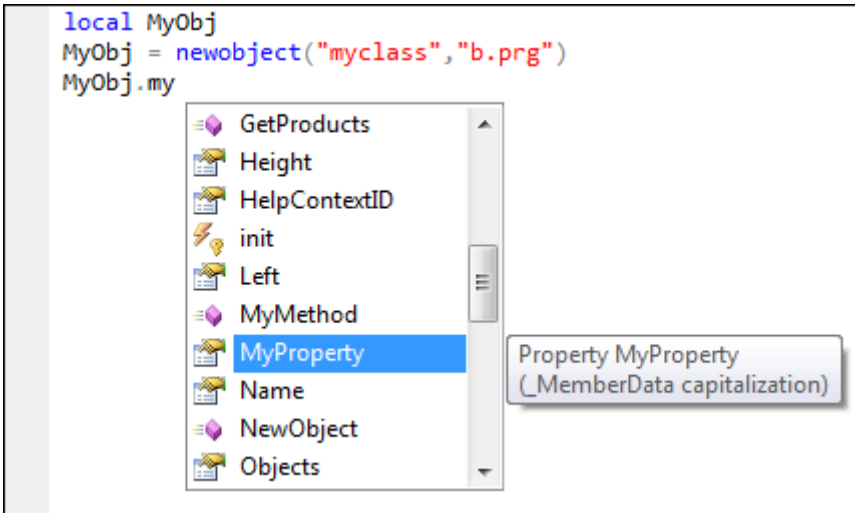


Figura 21.16

22- FOR EACH para objetos collections em run-time e designer-time.

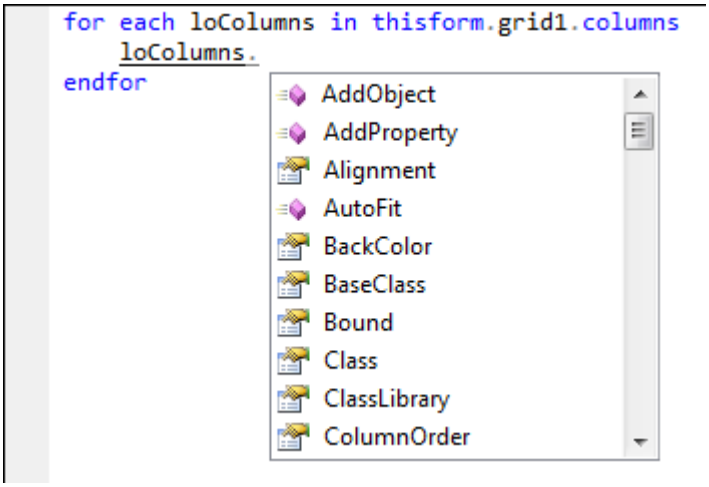


Figura 22.10

23- IntelliSense para objetos collection em run-time e designer-time.

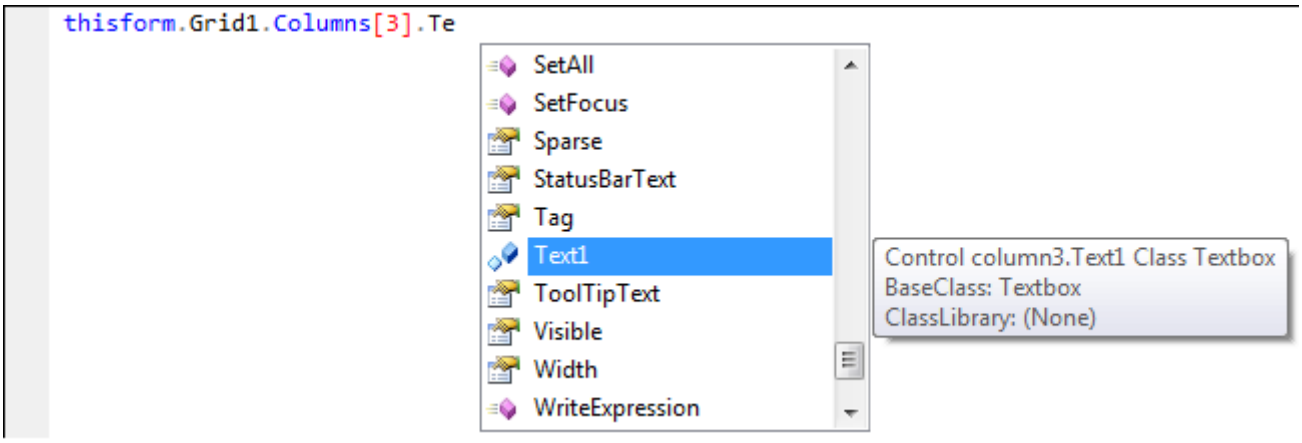


Figura 23.10

24- Referenciando um objecto em run-time e designer-time para uma variavel em write-time

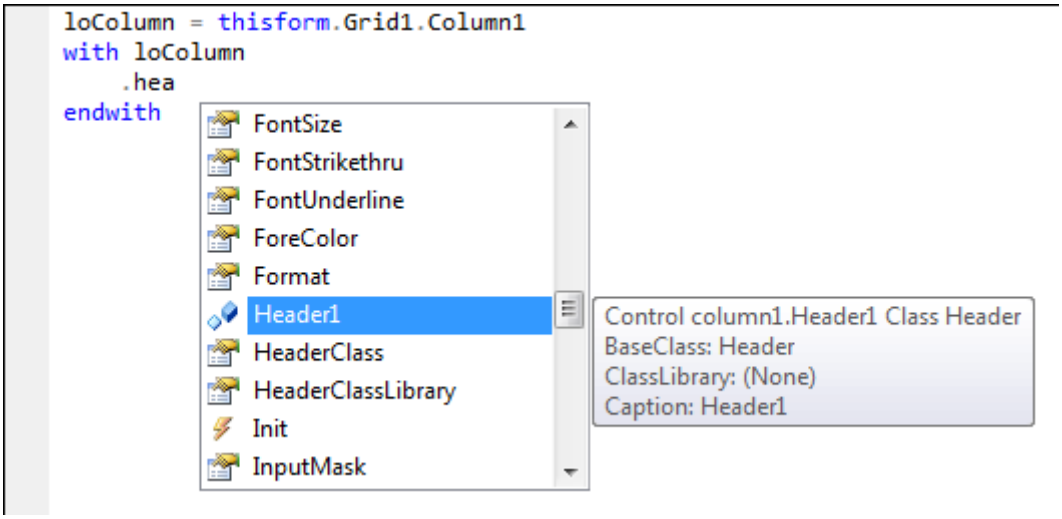


Figura 24.10

25- Documentando propriedades em programas (semelhante ao summary)

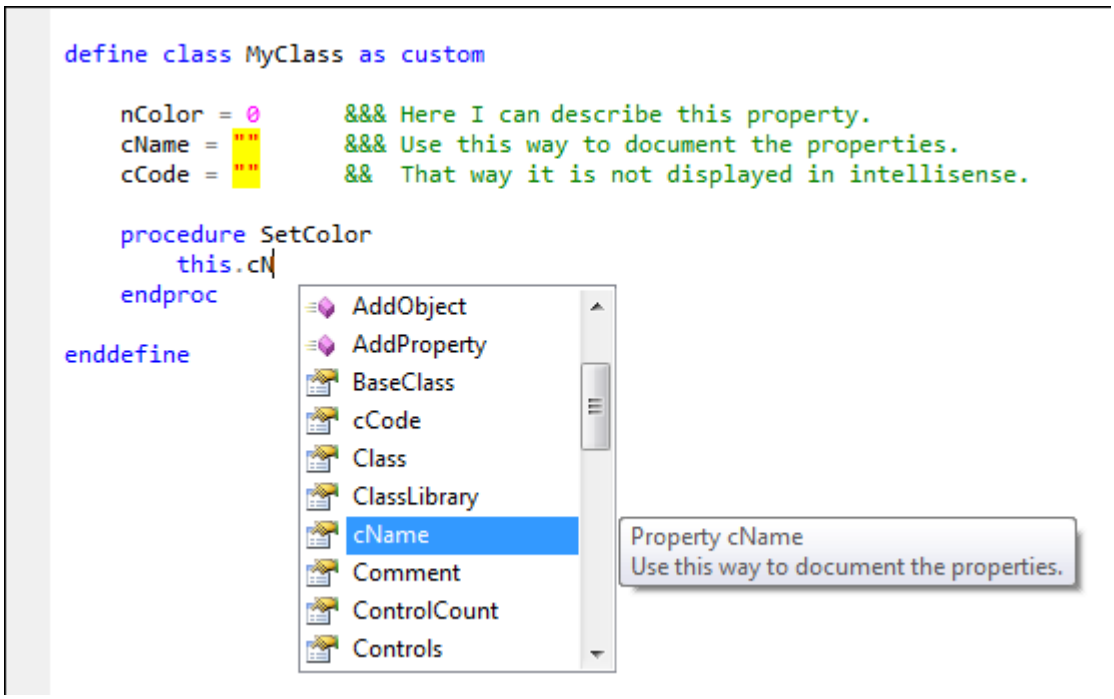


Figura 25.10

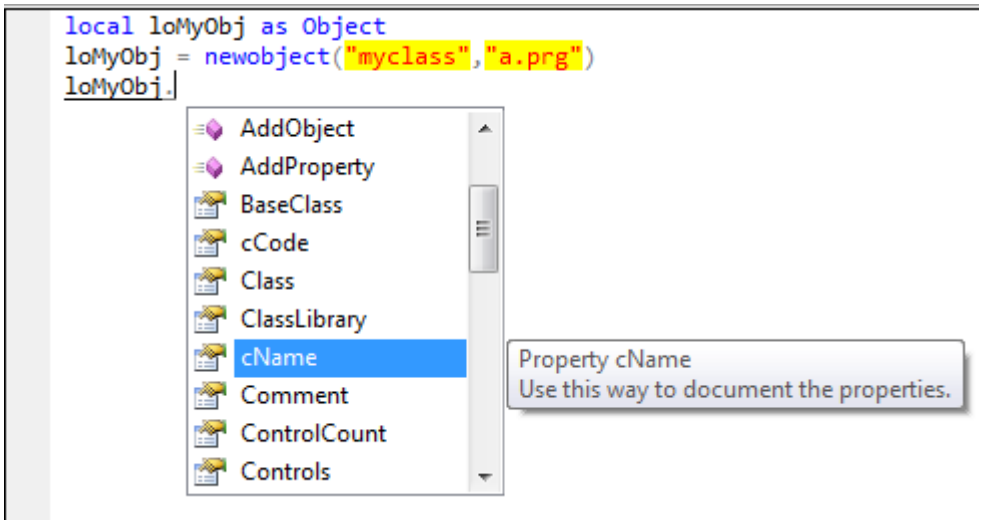


Figura 25.11

26- Help pressionando F1

Pressionando F1 com o IntelliSense aberto, o VFP irá abrir o help com o comando, função método ou evento posicionado.

27- Assinatura de procedures, funções, métodos e eventos.

Procedures e funções criadas dentro do PRG corrente ou invocadas pelo SET PROCEDURE TO... agora exibem o tooltip com a assinatura e tambem tem suporte ao summary, ou seja, conforme parametro posicionado o tooltip apresenta a respectiva informação do parametro definida no summary e o numero do parametro.

OBS: Funções nativas sem suporte a esta funcionalidade, assim como, varios pontos na utilização de métodos não oferecem suporte ao tooltip com a assinatura do método. Sera implementado nas próximas versões.

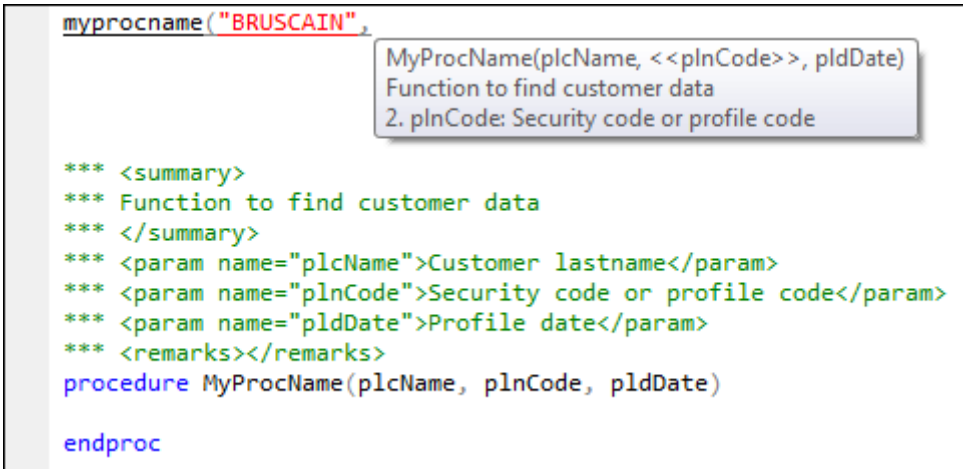


Figura 27.10

Caso ultrapasse a quantidade de parametros definidos na assinatura um erro em write-time será apresentado.

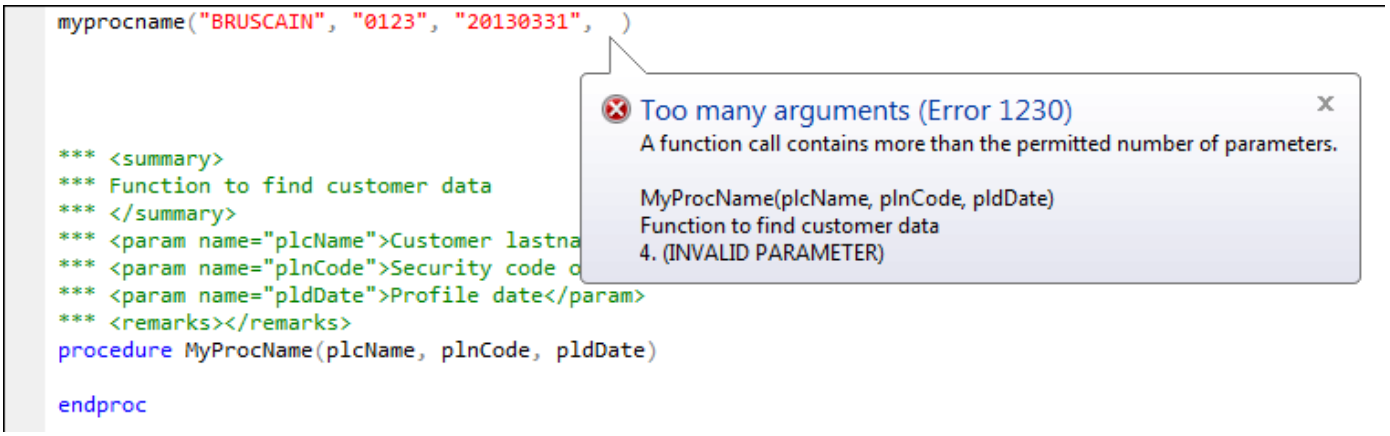


Figura 27.11

Para métodos e eventos a funcionalidade é a mesma.

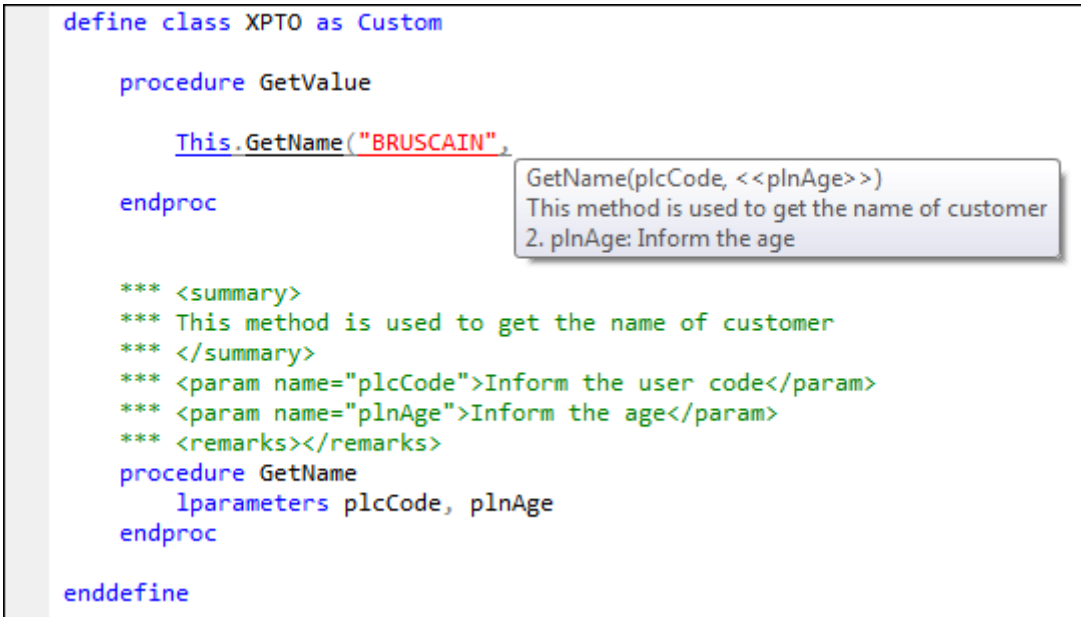


Figura 27.12

28- SELECT, INSERT, UPDATE e DELETE para banco de dados conectado. (Testado com MS Sql Server)

Esta funcionalidade apresenta o IntelliSence buscando informações de um banco de dados conectado.
É possível trabalhar disconectado, porem o IntelliSence apresenta somente as tabelas (sem campos) incluidas no comando SQL corrente. OBS: É necessário colocar o comando SQL em um bloco TEXT…ENDTEXT.

Você pode escolher as apções apresentadas abaixo e escolher a melhor forma de trabalhar.

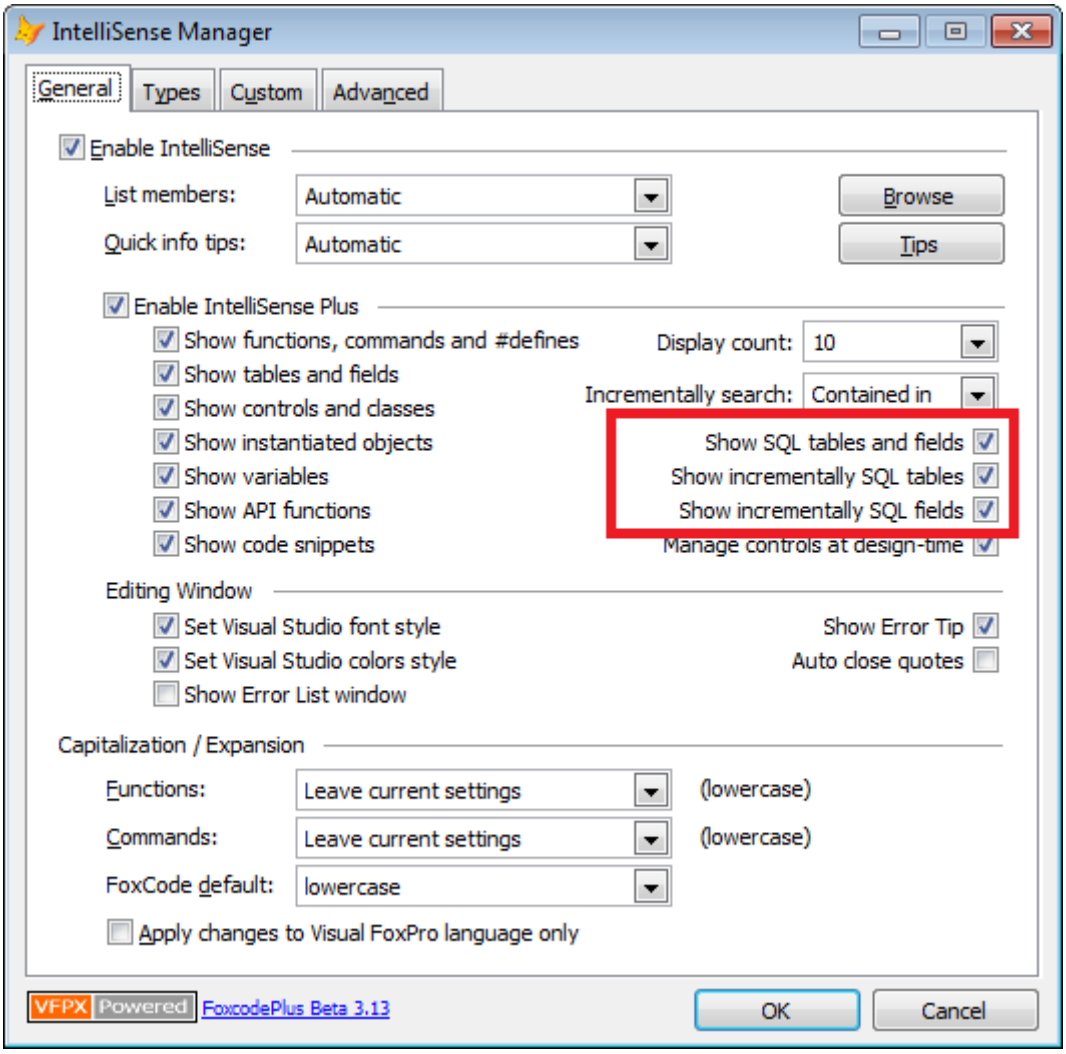


Figura 28.10

SELECT – Depois da clausula "FROM" e "JOIN" uma lista de tabelas e Alias de tabelas do banco de dados corrente é apresentada (modo não incremental)

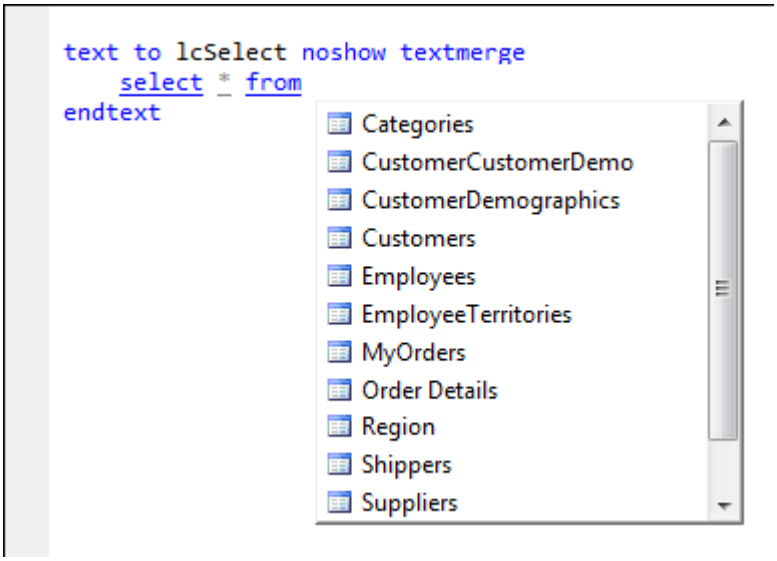


Figura 28.11

```
text to lcSelect noshow textmerge
    select  Customers.CustomerID, Customers.CompanyName, Customers.City,
            Orders.OrderID, Orders.OrderDate, orderitems.ProductID,
            Products.ProductName, MyOrders.ShipAddress
    from Orders MyOrders
    inner join Customers (nolock) on Orders.CustomerID = customers.CustomerID
    inner join [Order Details] OrderItems (nolock) on Orders.OrderID = OrderItems.OrderID
    inner join Products (nolock) on OrderItems.ProductID = Products.ProductID
    inner join
endtext
```

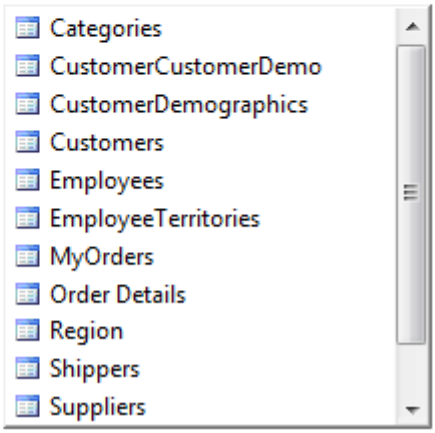


Figura 28.12

*SELECT – Como no **SQL Server Management Studio,** tabelas e Alias de tabelas são apresentados no IntelliSense no modo incremental.Todos os campos que pertencem a cada tabela incluida na SELECT tambem são apresentados.*

```
text to lcSelect noshow textmerge
    select  Customers.CustomerID, Customers.CompanyName, Customers.City,
            Orders.OrderID, Orders.OrderDate, orderitems.ProductID,
            Products.ProductName, m
    from Orders MyOrders
    inner join Customers (nolock) on
    inner join [Order Details] OrderI
    inner join Products (nolock) on C
    order by CustomerID, OrderID,
endtext
```

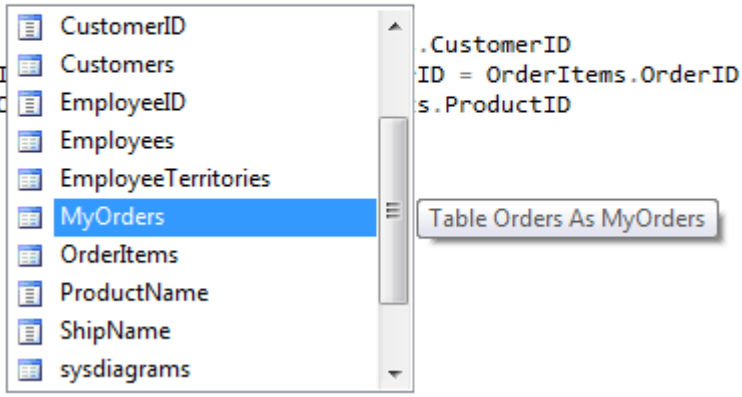


Figura 28.13

INSERT – IntelliSense para campos através de um Alias

```
text to lcSelect noshow textmerge
    select * from Categories Categ
    where Categ.
endtext
```

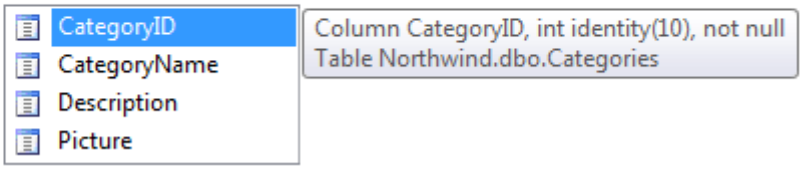


Figura 28.14

INSERT – Depois da clausula "INTO" uma lista de tabelas do banco de dados corrente é apresentada (modo não incremental)

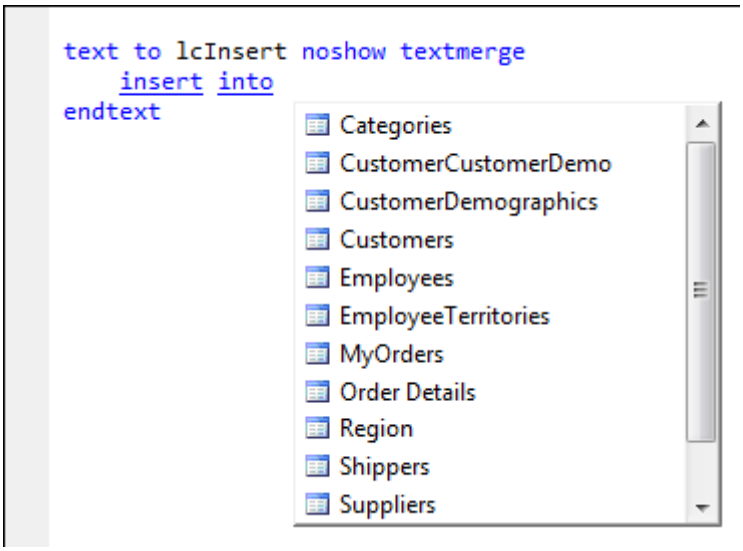


Figura 28.15

No modo incremental, os campos que pertencem a tabela defina após a clausula "INTO" são apresentados.

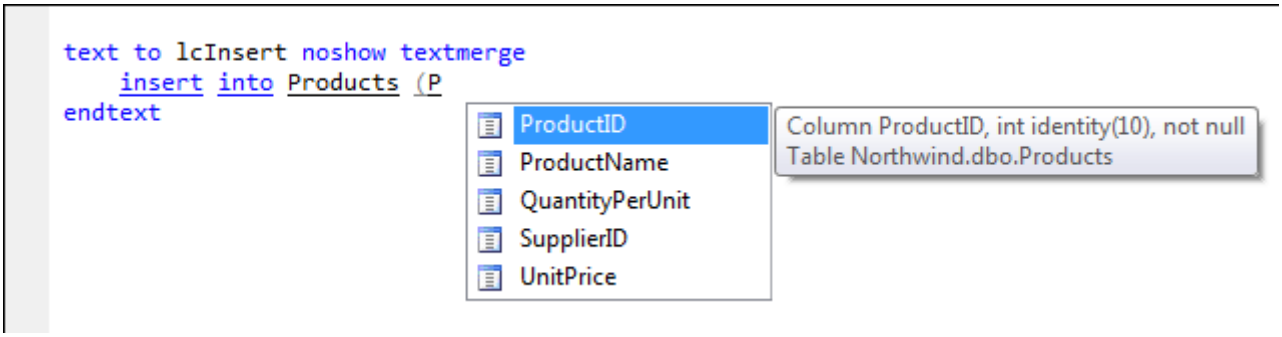


Figura 28.16

UPDATE – Depois do comado "UPDATE", uma lista de tabelas do banco de dados corrente é apresentada (modo não incremental)

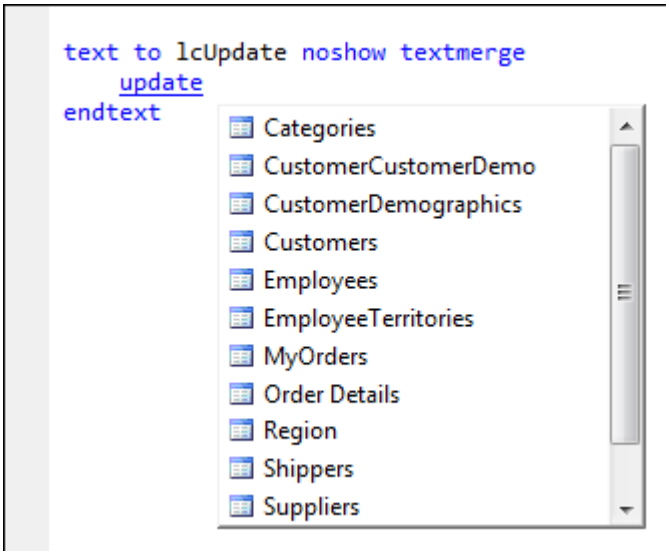


Figura 28.17

No modo incremental, os campos que pertencem a tabela definida depois do comando "UPDATE" são apresentados.

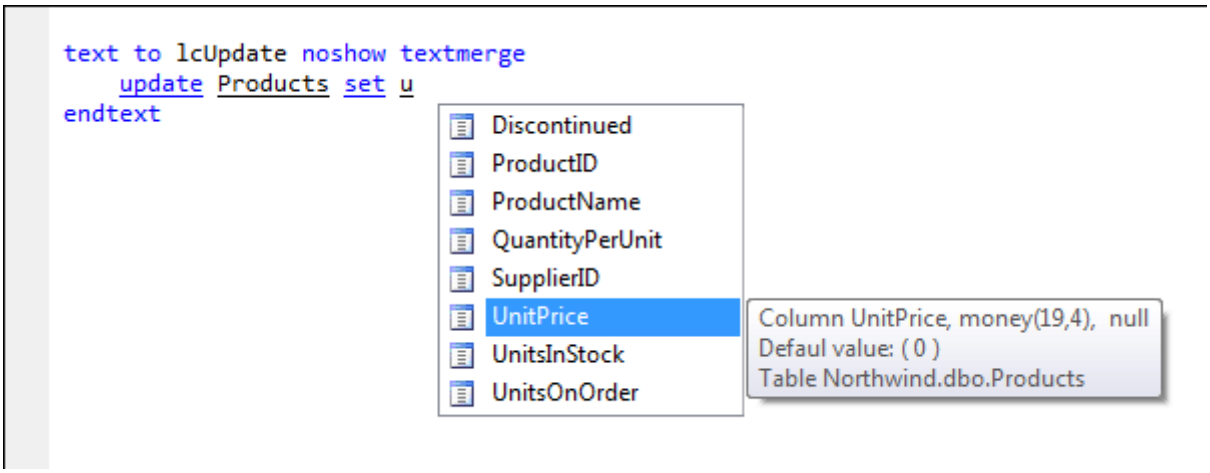


Figura 28.18

Depois da clausula "WHERE", no modo incremental, as tabelas do banco de dados corrente e os campos que pertencem a tabela definida depois do comando "UPDATE" são apresentos.

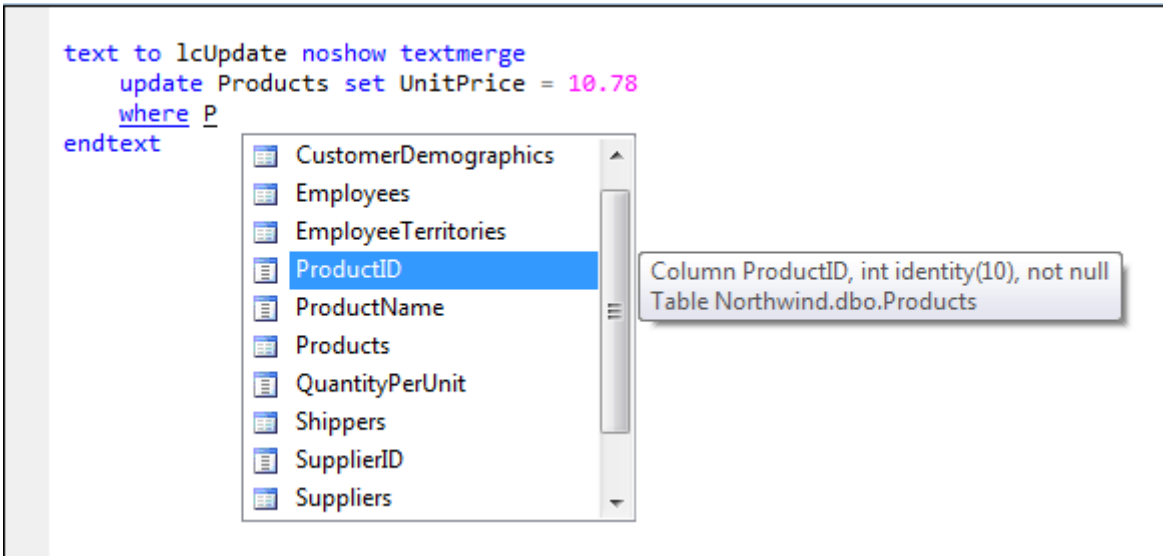


Figura 28.19

Depois da clausula "WHERE", no modo incremental, as tabelas do banco de dados corrente e os campos que pertencem a tabela definida depois da clausula "FROM" são apresentos.

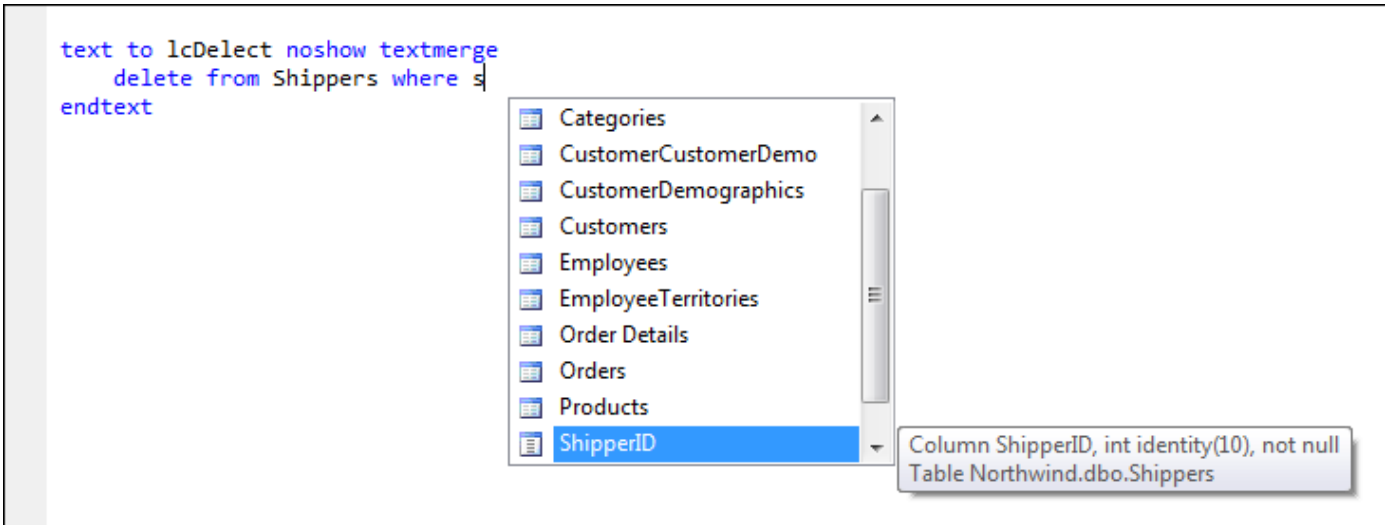


Figura 28.20

29- Error list

Apresenta os erros de programa através da compilação em write-time.

Clicando no erro indicado na lista, o VFP irá posicionar na linha de programa que contem o erro.

OBS: Esta opção pode deixar o VFP lento dependendo do tamanho do PRG.

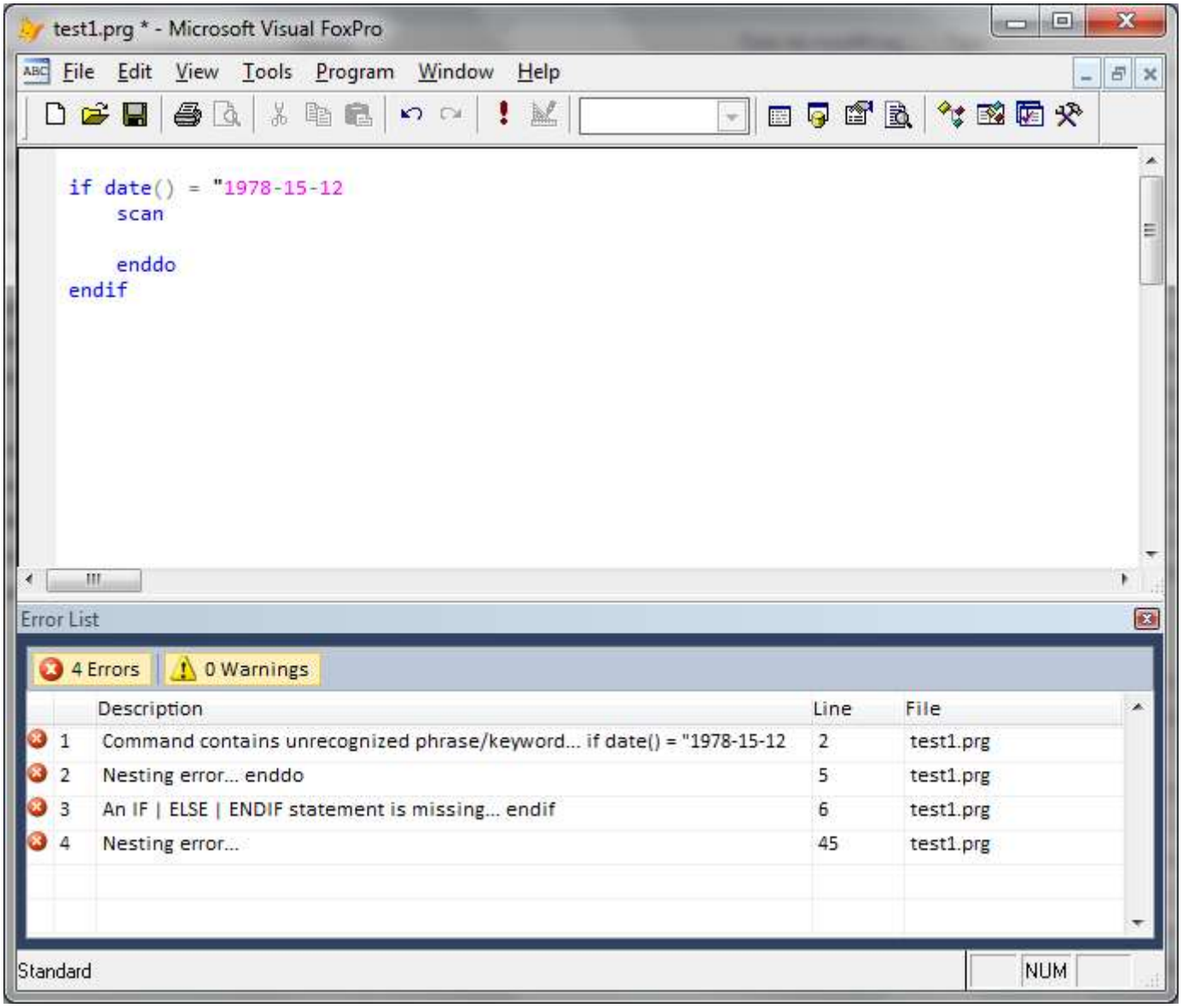


Figura 29.10

*Para ativar a janela "Error List" use a opção conforme abaixo.Você tambem pode configurar o VFP para que sempre inicie com a janela "Error List" aberta, para isso utilize o IntelliSense Manager marcando o checkbox "**Show Error List window**"*

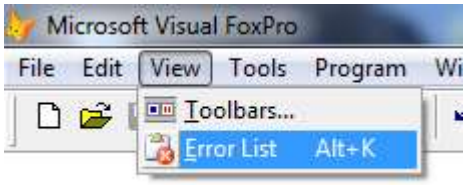


Figura 29.11

30- Error Tip

*Apresenta alguns possíveis erros de run-time em write-time. Para que os erros sejam apresentados é necessario que o checkbox "**Show Error Tip**" seja marcado no IntelliSense Manager*

Abaixo os Errors Tips disponiveis:

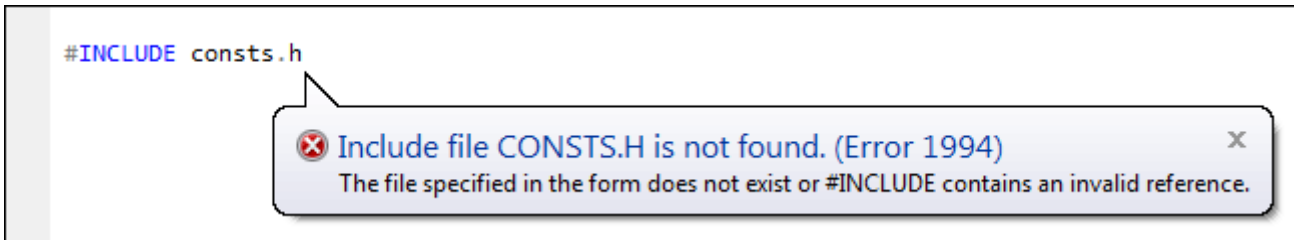


Figura 30.10

A mensagem abaixo é apresentada para THIS, THISFORM e THISFORMSET

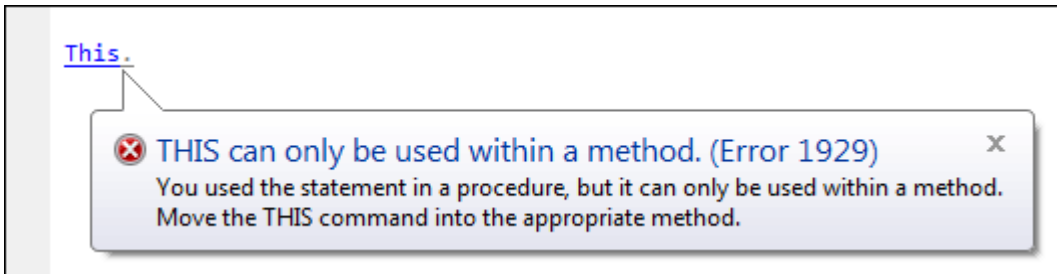


Figura 30.11

A mensagem abaixo é apresentada para PRG|MPR|QPR|FXP|APP|EXE

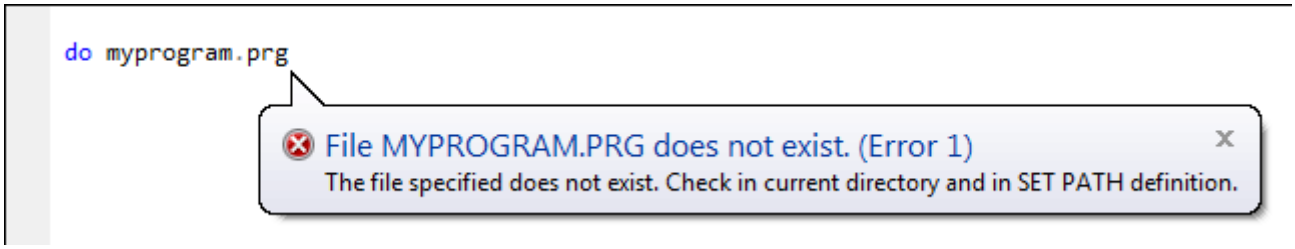


Figura 30.12

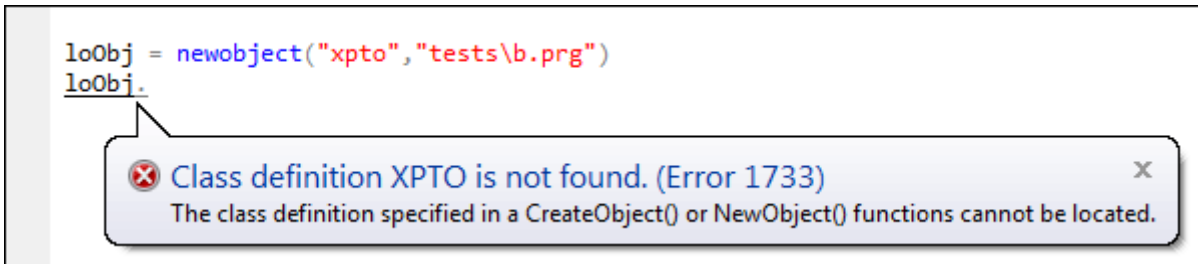


Figura 30.13

Figura 30.14

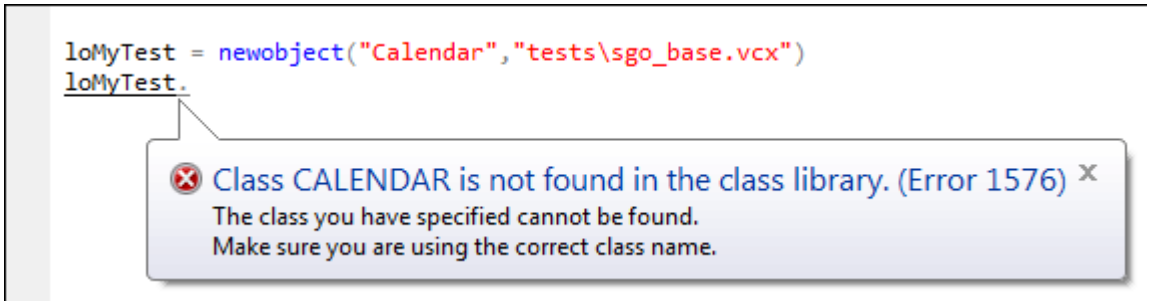


Figura 30.15

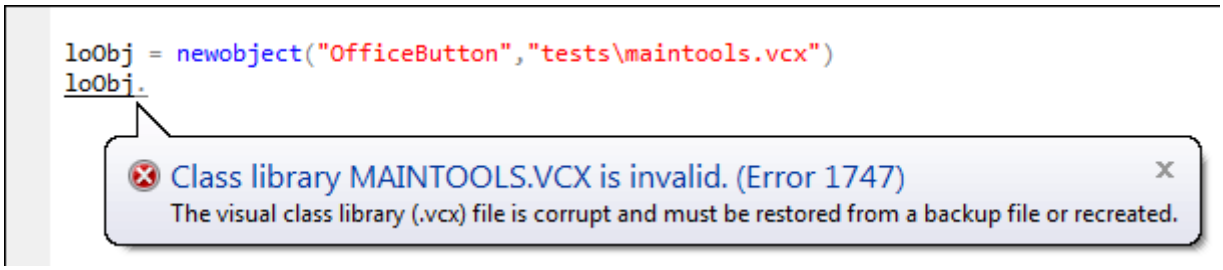


Figura 30.16

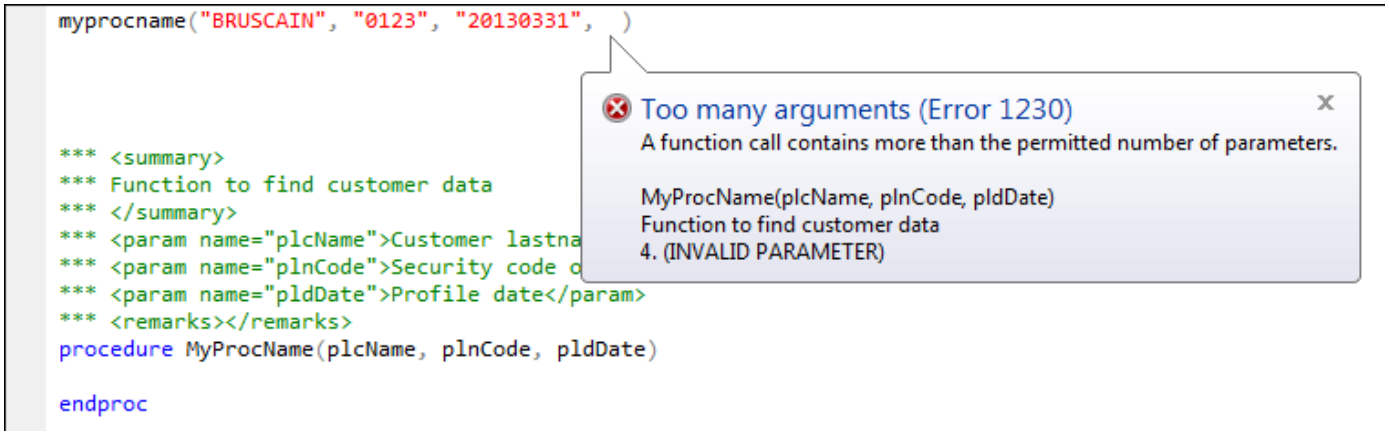


Figura 30.17

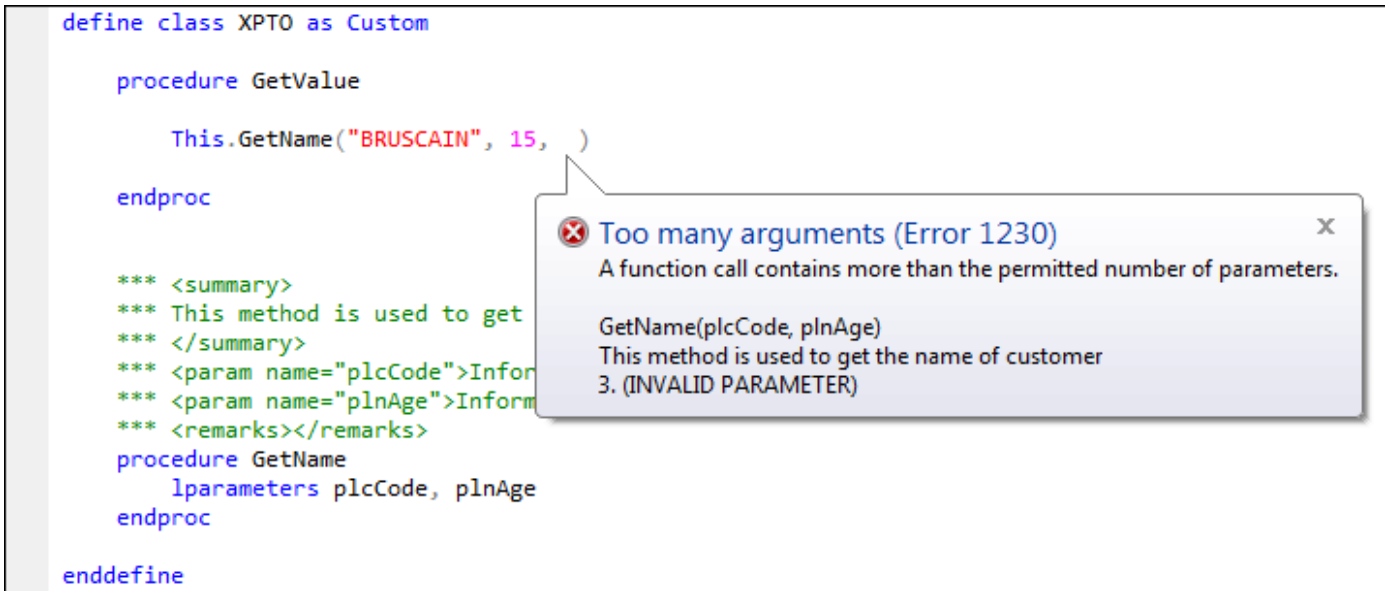
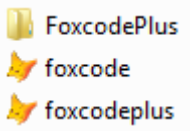


Figura 30.18

31- Instalando o FoxcodePlus (Somente para Visual FoxPro 9)

Arquivos disponibilizados:



Se você tem o FoxcodePlus 3.10 ou INFERIOR instalado, você deve substituir os arquivos foxcode.app , foxcodeplus.app e toda a pasta ...\Foxcodeplus*.*. Também é necessario fazer o 7 passo. Se você tem a versão 3.11 ou superior, apenas substitua os arquivos foxcode.app e foxcodeplus.app.

Siga abaixo os 7 passos para instalar pela **primeira vez**:

1) Se o VFP estiver aberto, feche-o.
2) Abra a pasta onde Microsoft Visual FoxPro 9 está instalado.
3) Renomeie o arquivo FoxCode.App
4) Copie os novos arquivos disponibilizados para a pasta.

5) Abra o VFP e acesse o IntelliSense Manager conforme figura abaixo:

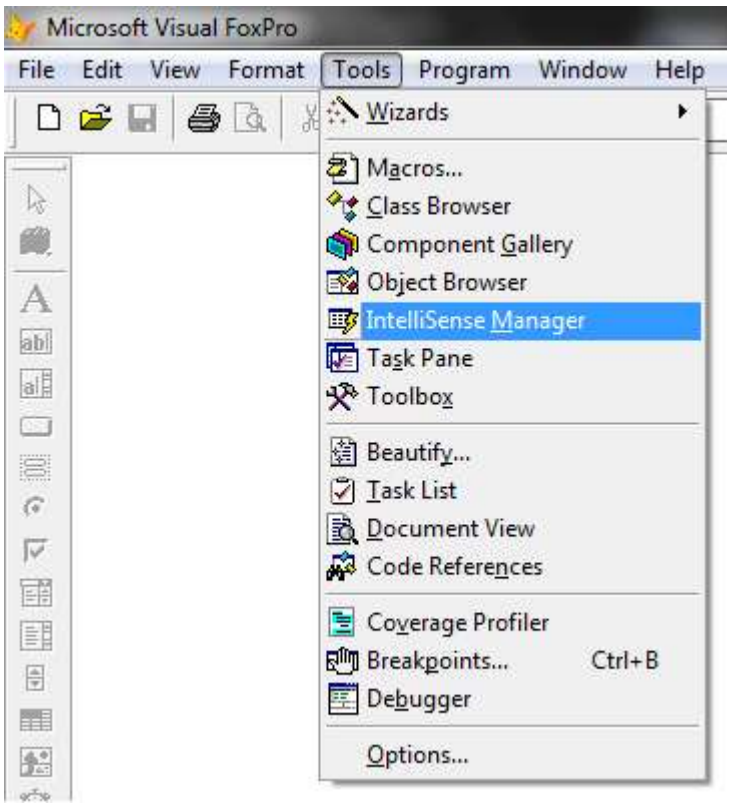


Figura 31.10

6) Faça as configurações conforme abaixo.

*Agora o IntelliSense Manager passa a ter algumas novas opções conforme destacado abaixo em vermelho. Ative o checkbox "**Enable IntelliSense Plus**", marque ou desmarque as opções conforme sua escolha. Por padrão o checkbox "**Show Error List Window**", "**Auto close quotes**" e "**Incremental started according typed**" não vêm marcadas.*

OBS 1: Caso o seu VFP esteja customizado com algum programa como "STARTUP" o mesmo será removido. Para resolver sua necessidade passe a utilizar um programa especifico para o "STARTUP" com a sua necessidade e a chamada do FOXCODEPLUS.APP e configure manualmente.

OBS 2: As cores estabelecidas pelo checkbox "Set Visual Studio colors style" podem ser reconfiguradas pelas opções gerais do editor do VFP.

OBS 3: Ao ativar o checkbox "Enable IntelliSense Plus" o arquivo CONFIG.FPW será criado, caso já exista, será modificado para o funcionamento correto do FOXCODEPLUS.APP.

OBS 4: Nem todos os novos recursos do FoxcodePlus funcionam na "Command Window".

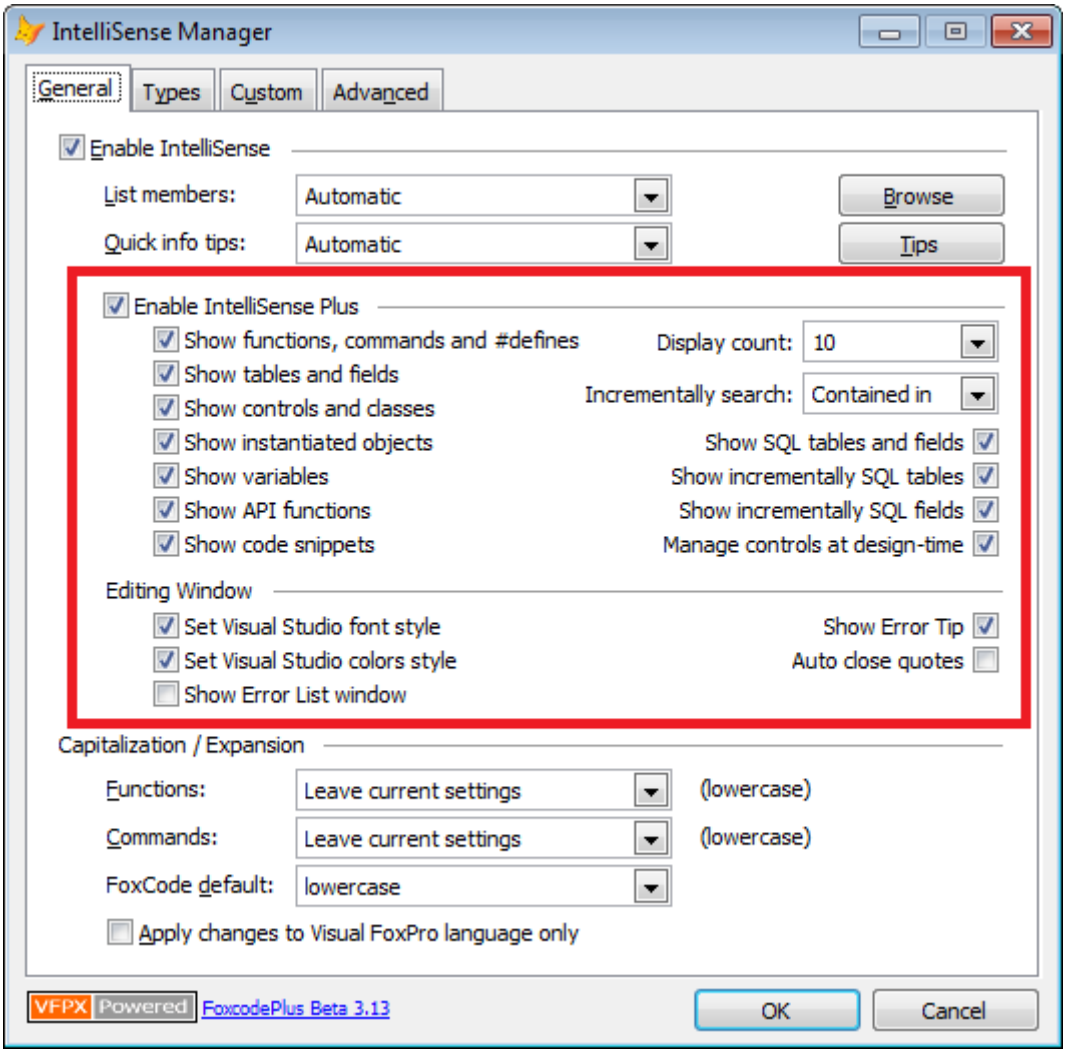


Figura 31.11

7) Ultimo passo, atualização da tabela FoxCode.

Com esta atualização você irá manter as suas customizações da tabela FoxCode. Um backup da sua tabela FoxCode será gerado automaticamente e um arquivo de log será gerado para indicar todas as atualizações e inclusões na sua tabela FoxCode.

A atualização irá alterar alguns registros existentes e irá incluir novos registros na sua tabela FoxCode. Com isso, existe a possibilidade da atualização sobrepor alguma customização caso a customização tenha a mesma informação no campo “TYPE”, “ABBREV” and “EXPANDED”.

Se você não tem nenhuma customização na tabela FoxCode, então não se preocupe. Se você tem, execute a atualização e no LOG verifique se oque foi alterado compromete alguma customização. Se foi afetado, você deve ajustar manualmente a sua tabela FoxCode.

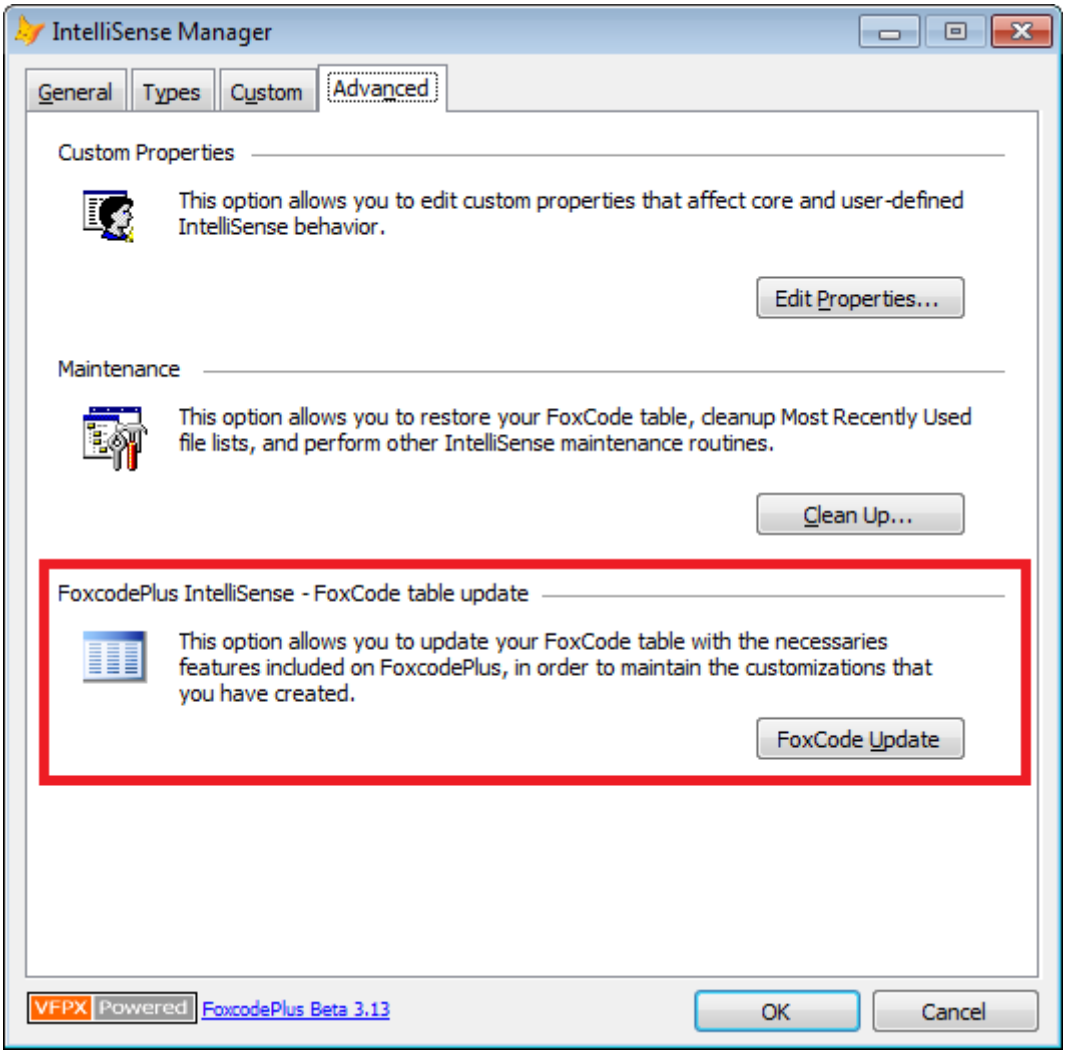


Figura 31.12

Obrigado a todos e bom divertimento.

Rodrigo Duarte Bruscain
Visual Studio | Visual C# | MS SQL Server | Visual FoxPro
https://www.mcpvirtualbusinesscard.com/VBCServer/rodrigobruscain/profile
www.linkedin.com/in/rodrigobruscain