

# Manual do TReport

## Descrição

Classe de impressão que substitui as funções SetPrint, SetDefault, RptStatus e Cabec.

A classe TReport permite que o usuário personalize as informações que serão apresentadas no relatório, alterando fonte (tipo, tamanho, etc), cor, tipo de linhas, cabeçalho, rodapé, etc.

Estrutura do componente TReport:

- O relatório (TReport) contém 1 ou mais seções (TRSection);
- Uma seção (TRSection) pode conter 1 ou mais seções;
- A seção (TRSection) contém células pré-definidas e células selecionadas pelo usuário;
- A seção (TRSection) também contém as quebras (TRBreak) para impressão de totalizadores (TRFunction);
- Os totalizadores são incluídos pela seção que automaticamente inclui no relatório (TReport).

## Métodos

| Método                | Descrição  |
|-----------------------|--|
| <u>New</u>            | Método construtor da classe TReport.   |
| <u>SetTitle</u>       | Define o título do componente.   |
| <u>SetParam</u>       | Define os parametros que serão utilizados pelo usuário para configurar o relatorio.  |
| <u>Preview</u>        | Apresenta preview do relatório.  |
| <u>SetAction</u>      | Define o bloco de código que será executado quando o usuário confirmar a impressão   |
| <u>Print</u>          | Inicia impressão do relatório executando o parametro bAction informado nos métodos New, SetAction ou PrintDialog.<br><br>Esse método é executado pelo método PrintDialog quando o usuário confirma a impressão do relatório.                               |
| <u>PrintDialog</u>    | Apresenta a tela de impressão para o usuário configurar o relatório.<br><br>O método PrintDialog executa o método Print quando o usuário confirma a impressão, executando o bloco de código informado pelo parametro bAction nos métodos New ou SetAction. |
| <u>Cancel</u>         | Retorna se o usuário cancelou a impressão do relatório.  |
| <u>SetMeter</u>       | Define o total da regua da tela de processamento do relatório.   |
| <u>IncMeter</u>       | Incrementa a régua da tela de processamento do relatório   |
| <u>SetDescription</u> | Define a descrição do componente.  |
| <u>ClassName</u>      | Retorna o nome da classe do componente.<br><br>Exemplo:<br><br><code>oReport := TReport():New('MATR010')</code><br><code>MsgStop(oReport:ClassName()) //Apresenta na tela 'TREPORT'</code>   |
| <u>ReportName</u>     | Retorna o nome do relatório (exemplo: MATR010)   |
| <u>Title</u>          | Retorna o título do componente   |

| <u>Description</u>    | Descrição do componente   |
|-----------------------|---|
| <u>TotalText</u>      | Retorna o texto que será impresso antes da impressão dos totalizadores  |
| <u>GetParam</u>       | Retorna a pergunta/bloco de código utilizada como parametro do relatório.   |
| <u>GetAction</u>      | Retorna bloco de código que será executado pelos métodos Print e PrintDialog quando usuário confirmar impressão.  |
| <u>FooterAtEnd</u>    | Retorna se será impressore rodapé no final da impressão   |
| <u>TotalInLine</u>    | Retorna se a impressão dos totalizadores será em linha  |
| <u>PageHeight</u>     | Retorna a altura da página baseado no tamanho da folha (A4, Carta, etc) informado pelo usuário  |
| <u>PageWidth</u>      | Retorna a largura da página baseado no tamanho da folha (A4, Carta, etc) informado pelo usuário   |
| <u>Row</u>            | Retorna a linha posicionada na impressão  |
| <u>Col</u>            | Retorna a coluna posicionada na impressão   |
| <u>Page</u>           | Retorna o número da página atual na impressão   |
| <u>GetOrientation</u> | Retorna a orientação de página (Retrato ou Paisagem) selecionada pelo usuário.  |
| <u>Printing</u>       | Retorna se relatório esta em processo de impressão.<br><br>O status de impressão é definido pelos métodos Init (que indica o início da impressão) e o método Finish (que indica a finalização).   |
| <u>Init</u>           | Incializa impressão.<br><br>Não é necessário executar o método Init se for utilizar o método Print ou PrintDialog, já que estes fazem o controle de inicialização e finalização da impressão.<br><br>Caso não esteja utilizando os método citados acima, sempre execute o método Init no inídio do relatório para verificar as configurações informadas pelo usuário e inicializar corretamente a primeira página do relatório.   |
| <u>Finish</u>         | Finaliza impressão inicializada pelo método Init.<br><br>Não é necessário executar o método Finish se for utilizar o método Print ou PrintDialog, já que estes fazem o controle de inicialização e finalização da impressão.<br><br>Caso não esteja utilizando os método citados acima, sempre execute o método Finish no final do relatório para impressão dos totalizadores que estão habilitados para o final do relatório, para fechar as queries e/ou índices temporários criados pelas seções e para outros tratamentos internos do componente. |
| <u>PrintText</u>      | Imprime o texto nas coordenadas informadas pelos parametros   |
| <u>EndPage</u>        | Finaliza um página de impressão   |
| <u>IncRow</u>         | Incrementa a linha de impressão baseado no parametro ou no valor definido pelo usuário para a altura das linhas do relatório.   |
| <u>SkipLine</u>       | Salta uma linha baseado na altura da linha informada pelo usuário   |
| <u>SetTotalText</u>   | Define o texto que será impresso antes da impressão dos totalizadores   |
| <u>SetTotalInLine</u> | Define se os totalizadores serão impressos em linha ou coluna   |
| <u>SetPortrait</u>    | Define orientação de página do relatório como retrato   |
| <u>SetLandscape</u>   | Define a orientação de página do relatório como paisagem  |
| <u>SetPreview</u>     | Define se será apresentado o preview o relatório no final da impressão  |
| <u>GetFunction</u>    | Retorna objeto da classe TRFunction (totalizadores) baseado no parametro  |

|                  |  |
|------------------|--|
|                  | informado.   |
| <u>Section</u>   | Retorna objeto da classe TRSection (seção) baseado no parametro informado.   |
| <u>Say</u>       | Imprime texto nas coordenadas informadas pelos parametros, utilizando as especificações do objeto TFont, cor da fonte e cor de fundo |
| <u>SayBitmap</u> | Imprime uma imagem nas coordenadas informadas pelos parametros   |
| <u>Line</u>      | Desenha uma linha baseado nas coordenadas informadas pelos parametros, utilizando as especificações do objeto TPen.                  |
| <u>Box</u>       | Desenha uma caixa baseado nas coordenadas informadas pelos parametros, utilizando as especificações do objeto TPen.                  |
| <u>FillRect</u>  | Preenche um retangulo na impressão utilizando as especificações do objeto TBrush.  |
| <u>ThinLine</u>  | Desenha uma linha simples iniciando na linha atual de impressão com a largura da página  |
| <u>FatLine</u>   | Desenha uma linha com altura maior que a do método ThinLine, iniciando na linha atual de impressão com a largura da página           |

## TReport sem usar método TRSection:Print

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```
#include "protheus.ch"

User Function MyReport()
Local oReport

If TRepInUse()      //verifica se relatorios personalizaveis esta disponivel
    Pergunte("MTR025",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection

oReport := TReport():New("MATR025","Relacao das Sugestoes de Orcamentos","MTR025",{ |oReport|
PrintReport(oReport)},"Este relatorio ira imprimir a relacao das Sugestoes de Venda conforme os parametros
solicitados.")

oSection := TRSection():New(oReport,OemToAnsi("Sugestoes de Orcamentos"),{"SBG","SBH","SB1"})

TRCell():New(oSection,"BG_PRODUTO","SBG")
TRCell():New(oSection,"B1_DESC","SB1")
TRCell():New(oSection,"BG_GERAPV","SBG")
TRCell():New(oSection,"BG_GERAOP","SBG")
TRCell():New(oSection,"BG_GERAOP1","SBG")
TRCell():New(oSection,"BG_GERAEMP","SBG")
TRCell():New(oSection,"BH_SEQUENC","SBH")
TRCell():New(oSection,"BH_CODCOMP","SBH")
TRCell():New(oSection,"B1_DESC","SB1")
TRCell():New(oSection,"BH_QUANT","SBH")
```

```

Return oReport

Static Function PrintReport(oReport)
Local oSection := oReport:Section(1)

DbSelectArea("SBG")
DbSetOrder(1)
DbSeek(xFilial()+MV_PAR01,.T.)

oReport:SetMeter(RecCount())

While ( !Eof() .And. xFilial("SBG") == SBG->BG_FILIAL .And. ;
        SBG->BG_PRODUTO >= MV_PAR01 .And. ;
        SBG->BG_PRODUTO <= MV_PAR02 )

    If oReport:Cancel()
        Exit
    EndIf

    DbSelectArea("SB1")
    DbSetOrder(1)
    DbSeek(xFilial("SB1")+SBG->BG_PRODUTO)

    oSection:Init()

    oSection:Cell("B1_DESC"):SetValue(SB1->B1_DESC)

    oSection:Cell("BG_PRODUTO"):Show()
    oSection:Cell("B1_DESC"):Show()
    oSection:Cell("BG_GERAPV"):Show()
    oSection:Cell("BG_GERAOP"):Show()
    oSection:Cell("BG_GERAOP1"):Show()
    oSection:Cell("BG_GERAEMP"):Show()

    dbSelectArea("SBH")
    dbSetOrder(1)
    dbSeek(xFilial("SBH")+SBG->BG_PRODUTO,.T.)

    While ( !Eof() .And. SBH->BH_FILIAL == xFilial("SBH") .And.;
            SBH->BH_PRODUTO == SBG->BG_PRODUTO )

        dbSelectArea("SB1")
        dbSetOrder(1)
        dbSeek(xFilial("SB1")+SBH->BH_CODCOMP)

        oSection:PrintLine()

        oSection:Cell("BG_PRODUTO"):Hide()
        oSection:Cell("B1_DESC"):Hide()
        oSection:Cell("BG_GERAPV"):Hide()
        oSection:Cell("BG_GERAOP"):Hide()
        oSection:Cell("BG_GERAOP1"):Hide()
        oSection:Cell("BG_GERAEMP"):Hide()

        dbSelectArea("SBH")
        dbSkip()

    End

    DbSelectArea("SBG")
    DbSkip()

    oSection:Finish()

    oReport:SkipLine()
    oReport:IncMeter()

```

```
End
Return
```

## TReport sem usar método TRSection:Print com seções

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```
#include "protheus.ch"

User Function MyReport()
Local oReport

If TRRepInUse()      //verifica se relatorios personalizaveis esta disponivel
    Pergunte("MTR025",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection1
Local oSection2

oReport := TReport():New("MATR025","Relacao das Sugestoes de Orcamentos","MTR025",{oReport|
PrintReport(oReport)},"Este relatorio ira imprimir a relacao das Sugestoes de Venda conforme os parametros
solicitados.")

oSection1 := TRSection():New(oReport,"Produtos",{ "SBG","SB1"})

TRCell():New(oSection1,"BG_PRODUTO","SBG")
TRCell():New(oSection1,"B1_DESC","SB1")
TRCell():New(oSection1,"BG_GERAPV","SBG")
TRCell():New(oSection1,"BG_GERAOP","SBG")
TRCell():New(oSection1,"BG_GERAOP1","SBG")
TRCell():New(oSection1,"BG_GERAEMP","SBG")

TRPosition(oSection1,"SB1",1,{|| xFilial("SB1") + SBG->BG_PRODUTO})

oSection2 := TRSection():New(oSection1,"Componentes",{ "SBH","SB1"})

TRCell():New(oSection2,"BH_SEQUENC","SBH")
TRCell():New(oSection2,"BH_CODCOMP","SBH")
TRCell():New(oSection2,"B1_DESC","SB1")
TRCell():New(oSection2,"BH_QUANT","SBH")

TRPosition(oSection2,"SB1",1,{|| xFilial("SB1") + SBH->BH_CODCOMP})

Return oReport

Static Function PrintReport(oReport)
Local oSection1 := oReport:Section(1)
Local oSection2 := oReport:Section(1):Section(1)

DbSelectArea("SBG")
DbSetOrder(1)
DbSeek(xFilial()+MV_PAR01,.T.)

oReport:SetMeter(RecCount())
```

```

While ( !Eof() .And. xFilial("SBG") == SBG->BG_FILIAL .And. ;
        SBG->BG_PRODUTO >= MV_PAR01 .And. ;
        SBG->BG_PRODUTO <= MV_PAR02 )

    If oReport:Cancel()
        Exit
    EndIf

    oSection1:Init()
    oSection1:PrintLine()

    dbSelectArea("SBH")
    dbSetOrder(1)
    dbSeek(xFilial("SBH")+SBG->BG_PRODUTO,.T.)

    oSection2:Init()

    While ( !Eof() .And. SBH->BH_FILIAL == xFilial("SBH") .And.;
            SBH->BH_PRODUTO == SBG->BG_PRODUTO )

        oSection2:PrintLine()

        dbSelectArea("SBH")
        dbSkip()
    End

    oSection2:Finish()

    DbSelectArea("SBG")
    DbSkip()

    oSection1:Finish()

    oReport:IncMeter()

End
Return

```

## TReport usando método TRSection:Print com 1 seção

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```

#include "protheus.ch"

User Function MyReport()
Local oReport

If TRepInUse()
    Pergunte("MTR530",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection
Local oBreak

```

```

oReport := TReport():New("MYREPORT","Relatorio de Visitas","MTR530",{oReport| PrintReport(oReport)},"Relatorio
de visitas de vendedores nos clientes")

oSection := TRSection():New(oReport,"Clientes",{ "SA1","SA3"})

TRCell():New(oSection,"A1_VEND","SA1")
TRCell():New(oSection,"A3_NOME","SA3")
TRCell():New(oSection,"A1_COD","SA1","Cliente")
TRCell():New(oSection,"A1_LOJA","SA1")
TRCell():New(oSection,"A1_NOME","SA1")
TRCell():New(oSection,"A1_ULTVIS","SA1")
TRCell():New(oSection,"A1_TEMVIS","SA1")
TRCell():New(oSection,"A1_CONTATO","SA1")
TRCell():New(oSection,"A1_TEL","SA1")

oBreak := TRBreak():New(oSection,oSection:Cell("A1_VEND"),"Sub Total Vendedores")

TRFunction():New(oSection:Cell("A1_COD"),NIL,"COUNT",oBreak)
TRFunction():New(oSection:Cell("A1_TEMVIS"),NIL,"SUM",oBreak)

Return oReport

Static Function PrintReport(oReport)
Local oSection := oReport:Section(1)
Local cPart
Local cFiltro := ""

#IFDEF TOP

    //Transforma parametros do tipo Range em expressao SQL para ser utilizada na query
    MakeSqlExpr("MTR530")

    oSection:BeginQuery()

    If ( mv_par03 == 1 )
        cPart := "%AND (" + Dtos(dDataBase) + " - A1_ULTVIS) > A1_TEMVIS%"
    Else
        cPart := "%%"
    EndIf

    BeginSql alias "QRYSA1"
        SELECT A1_COD,A1_LOJA,A1_NOME,A1_VEND,A1_ULTVIS,A1_TEMVIS,A1_TEL,A1_CONTATO,A3_NOME
        FROM %table:SA1% SA1,%table:SA3% SA3
        WHERE A1_VEND = A3_COD AND A1_FILIAL = %xfilial:SA1% AND
            A1_TEMVIS > 0 AND SA1.%notDel% %exp:cPart%
        ORDER BY A1_VEND
    EndSql

    /*
    Prepara relatorio para executar a query gerada pelo Embedded SQL passando como
    parametro a pergunta ou vetor com perguntas do tipo Range que foram alterados
    pela funcao MakeSqlExpr para serem adicionados a query
    */
    oSection:EndQuery(mv_par04)

#ELSE

    //Transforma parametros do tipo Range em expressao ADVPL para ser utilizada no filtro
    MakeAdvplExpr("MTR530")

    //Adiciona no filtro o parametro tipo Range
    If !Empty(mv_par04)
        cFiltro += mv_par04 + " .AND. "
    EndIf

    cFiltro += " A1_TEMVIS > 0 "

```

```

    If ( mv_par03 == 1 )
        cFiltro += ' .AND. ('+DtoC(dDataBase)+'-A1_ULTVIS) > A1_TEMVIS '
    EndIf

    oSection:SetFilter(cFiltro,"A1_VEND")

    TRPosition():New(oReport:Section(1),"SA3",1,{|| xFilial() + SA1->A1_VEND})

#ENDIF

oSection:Print()

Return

```

## TReport usando método TRSection:SetParentQuery

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```

#include "protheus.ch"

User Function MyReport()
Local oReport

If TRepInUse()
    Pergunte("MTR530",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection1
Local oSection2

oReport := TReport():New("MYREPORT","Relatorio de Visitas","MTR530",{||oReport| PrintReport(oReport)},"Relatorio
de visitas de vendedores nos clientes")

oSection1 := TRSection():New(oReport,"Vendedores","SA3")

TRCell():New(oSection1,"A3_COD","SA3","Vendedor")
TRCell():New(oSection1,"A3_NOME","SA3")

TRFunction():New(oSection1:Cell("A3_COD"),NIL,"COUNT",NIL,NIL,NIL,NIL,.F.)

oSection2 := TRSection():New(oSection1,"Clientes","SA1")

TRCell():New(oSection2,"A1_COD","SA1","Cliente")
TRCell():New(oSection2,"A1_LOJA","SA1")
TRCell():New(oSection2,"A1_NOME","SA1")
TRCell():New(oSection2,"A1_ULTVIS","SA1")
TRCell():New(oSection2,"A1_TEMVIS","SA1")
TRCell():New(oSection2,"A1_CONTATO","SA1")
TRCell():New(oSection2,"A1_TEL","SA1")

TRFunction():New(oSection2:Cell("A1_COD"),NIL,"COUNT")
TRFunction():New(oSection2:Cell("A1_TEMVIS"),NIL,"SUM")

Return oReport

```



```

Static Function PrintReport(oReport)
Local oSection1 := oReport:Section(1)
Local oSection2 := oReport:Section(1):Section(1)
Local cFiltro   := ""

#IFDEF TOP

    oSection1:BeginQuery()

    If ( mv_par03 == 1 )
        cPart := "%AND (" + Dtos(dDataBase) + " - A1_ULTVIS) > A1_TEMVIS%"
    Else
        cPart := "%%"
    EndIf

    BeginSql alias "QRYSA3"
        SELECT A1_COD,A1_LOJA,A1_NOME,A1_VEND,A1_ULTVIS,A1_TEMVIS,A1_TEL,A1_CONTATO,A3_NOME,A3_COD
        FROM %table:SA1% SA1,%table:SA3% SA3
        WHERE A1_VEND = A3_COD AND A1_FILIAL = %xfilial:SA1% AND
                A1_TEMVIS > 0 AND SA1.%notDel%
        ORDER BY A1_VEND
    EndSql

    /*
    Prepara relatorio para executar a query gerada pelo Embedded SQL passando como
    parametro a pergunta ou vetor com perguntas do tipo Range que foram alterados
    pela funcao MakeSqlExpr para serem adicionados a query
    */
    oSection1:EndQuery(mv_par04)

    oSection2:SetParentQuery()
    oSection2:SetParentFilter({|cParam| QRYSA3->A1_VEND >= cParam .and. QRYSA3->A1_VEND <= cParam},{||
QRYSA3->A3_COD})

#ELSE

    cFiltro := " A1_TEMVIS > 0 "

    If ( mv_par03 == 1 )
        cFiltro += ' .AND. ('+Dtos(dDataBase)+'-A1_ULTVIS) > A1_TEMVIS '
    EndIf

    oSection2:SetFilter(cFiltro,"A1_VEND")
    oSection2:SetRelation({|| SA3->A3_COD})
    oSection2:SetParentFilter({|cParam| SA1->A1_VEND >= cParam .and. SA1->A1_VEND <= cParam},{|| SA3-
>A3_COD})

#ENDIF

oSection1:Print()

Return

```

# Funções

## ResetReport

### Sintaxe

oObj:ResetReport ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Reinicia valor do totalizador para o relatório

## Box

### Sintaxe

oObj:Box ( < nRow > , < nCol > , < nBottom > , < nRight > , [ oPen ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição             |
|-----------|----------|-----------------------|
| nRow      | Numérico | Linha inicial         |
| nCol      | Numérico | Coluna inicial        |
| nBottom   | Numérico | Linha final           |
| nRight    | Numérico | Coluna final          |
| oPen      | Objeto   | Objeto da classe TPen |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Desenha uma caixa baseado nas coordenadas informadas pelos parametros, utilizando as especificações do objeto TPen.

## Cancel

### Sintaxe

oObj:Cancel ( ) --> ICanceled

### Retorno

| Tipo   | Descrição                                |
|--------|--|
| Lógico | Verdadeiro se usuário cancelou impressão |

### Descrição

Retorna se o usuário cancelou a impressão do relatório.

## ClassName

### Sintaxe

oObj:ClassName ( ) --> cClassName

### Retorno

| Tipo     | Descrição      |
|----------|----------------|
| Caracter | Nome da classe |

### Descrição

Retorna o nome da classe do componente.

Exemplo:

```
oReport := TReport():New('MATR010')  
MsgStop(oReport:ClassName()) //Apresenta na tela 'TREPORT'
```

## Col

### Sintaxe

oObj:Col ( ) --> nCol

### Retorno

| Tipo     | Descrição                       |
|----------|---------------------------------|
| Numérico | Coluna posicionada na impressão |

### Descrição

Retorna a coluna posicionada na impressão

## Description

### Sintaxe

oObj:Description ( ) --> cDescription

### Retorno

| Tipo     | Descrição               |
|----------|-------------------------|
| Caracter | Descrição do componente |

### Descrição

Descrição do componente

## EndPage

### Sintaxe

oObj:EndPage ( [ IFooter ] ) --> NIL

### Parâmetros

| Argumento | Tipo   | Descrição                              |
|-----------|--------|--|
| IFooter   | Lógico | Imprime rodapé.<br>Valor padrão: Falso |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Finaliza uma página de impressão

## FatLine

### Sintaxe

oObj:FatLine ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Desenha uma linha com altura maior que a do método ThinLine, iniciando na linha atual de impressão com a largura da página

## FillRect

### Sintaxe

oObj:FillRect ( < aRect > , [ oBrush ] ) --> NIL

### Parâmetros

| Argumento | Tipo   | Descrição  |
|-----------|--------|--|
| aRect     | Array  | Vetor com coordenadas no formato: {linha inicial, coluna inicial, linha final, coluna final} |
| oBrush    | Objeto | Objeto da classe TBrush  |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Preenche um retângulo na impressão utilizando as especificações do objeto TBrush.

## Finish

### Sintaxe

oObj:Finish ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Finaliza impressão inicializada pelo método Init.

Não é necessário executar o método Finish se for utilizar o método Print ou PrintDialog, já que estes fazem o controle de inicialização e finalização da impressão.

Caso não esteja utilizando os métodos citados acima, sempre execute o método Finish no final do relatório para impressão dos totalizadores que estão habilitados para o final do relatório, para fechar as queries e/ou índices temporários criados pelas seções e para outros tratamentos internos do componente.

## FooterAtEnd

## Sintaxe

oObj:FooterAtEnd ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna se será impressore rodapé no final da impressão

## GetAction

### Sintaxe

oObj:GetAction ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna bloco de código que será executado pelos métodos Print e PrintDialog quando usuário confirmar impressão.

## GetFunction

### Sintaxe

oObj:GetFunction ( < uFunction > ) --> oFunction

## Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| uFunction | (Qualquer) | Tipo Caracter: ID do objeto<br>Tipo Numérico: Índice do objeto segundo a ordem de criação dos componentes TRFunction |

## Retorno

| Tipo   | Descrição  |
|--------|--|
| Objeto | Objeto da classe TRFunction.<br>Se não for encontrado o objeto segundo o parametro informado será retornado Nulo |

## Descrição

Retorna objeto da classe TRFunction (totalizadores) baseado no parametro informado.

## GetOrientation

### Sintaxe

oObj:GetOrientation ( ) --> nOrientation

### Retorno

| Tipo     | Descrição  |
|----------|--|
| Numérico | Orientação da página:<br>1 - Retrato<br>2 - Paisagem |

### Descrição

Retorna a orientação de página (Retrato ou Paisagem) selecionada pelo usuário.

## GetParam

### Sintaxe

oObj:GetParam ( ) --> uParam

### Retorno

| Tipo       | Descrição  |
|------------|--|
| (Qualquer) | Pergunta (SX1) ou bloco de código utilizado como parametro |

### Descrição

Retorna a pergunta/bloco de código utilizada como parametro do relatório.

## IncMeter

### Sintaxe

oObj:IncMeter ( [ nInc ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição   |
|-----------|----------|---|
| nInc      | Numérico | Quantidade a incrementar na régua.<br>Valor padrão: 1 |

### Retorno

| Tipo | Descrição |
|------|-----------|
|------|-----------|

(NULO)

Retorno nulo.

## Descrição

Incrementa a régua da tela de processamento do relatório

## IncRow

### Sintaxe

oObj:IncRow ( [ nInc ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição  |
|-----------|----------|--|
| nInc      | Numérico | Quantidade a incrementar.<br>Valor padrão definida pelo usuário para a altura da linha |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Incrementa a linha de impressão baseado no parametro ou no valor definido pelo usuário para a altura das linhas do relatório.

## Init

### Sintaxe

oObj:Init ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Incializa impressão.

Não é necessário executar o método Init se for utilizar o método Print ou PrintDialog, já que estes fazem o controle de inicialização e finalização da impressão.

Caso não esteja utilizando os método citados acima, sempre execute o método Init no inídio do relatório para verificar as configurações informadas pelo usuário e inicializar corretamente a primeira página do relatório.



# Line

## Sintaxe

oObj:Line ( < nTop > , < nCol > , < nBottom > , < nRight > , [ oPen ] ) --> NIL

## Parâmetros

| Argumento | Tipo     | Descrição             |
|-----------|----------|-----------------------|
| nTop      | Numérico | Linha inicial         |
| nCol      | Numérico | Coluna inicial        |
| nBottom   | Numérico | Linha final           |
| nRight    | Numérico | Coluna final          |
| oPen      | Objeto   | Objeto da classe TPen |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Desenha uma linha baseado nas coordenadas informadas pelos parametros, utilizando as especificações do objeto TPen.

# Page

## Sintaxe

oObj:Page ( ) --> nPage

## Retorno

| Tipo     | Descrição        |
|----------|------------------|
| Numérico | Número da página |

## Descrição

Retorna o número da página atual na impressão

# PageHeight

## Sintaxe

oObj:PageHeight ( ) --> nPageHeight

## Retorno

| Tipo     | Descrição        |
|----------|------------------|
| Numérico | Altura da página |

### Descrição

Retorna a altura da página baseado no tamanho da folha (A4, Carta, etc) informado pelo usuário

## PageWidth

### Sintaxe

oObj:PageWidth ( ) --> nPageWidth

### Retorno

| Tipo     | Descrição         |
|----------|-------------------|
| Numérico | Largura da página |

### Descrição

Retorna a largura da página baseado no tamanho da folha (A4, Carta, etc) informado pelo usuário

## Preview

### Sintaxe

oObj:Preview ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Apresenta preview do relatório.

## Print

### Sintaxe

oObj:Print ( [ IDlg ] ) --> NIL

### Parâmetros

| Argumento | Tipo   | Descrição   |
|-----------|--------|---|
| IDlg      | Lógico | Verdadeiro para apresentar a tela de processamento do relatório |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Inicia impressão do relatório executando o parametro bAction informado nos métodos New, SetAction ou PrintDialog.

Esse método é executado pelo método PrintDialog quando o usuário confirma a impressão do relatório.

## PrintDialog

### Sintaxe

```
oObj:PrintDialog ( [ bAction ] ) --> IPrinted
```

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| bAction   | Code-Block | Bloco de código que será executado quando usuário confirmar a impressão do relatório |

## Retorno

| Tipo   | Descrição  |
|--------|--|
| Lógico | Verdadeiro se usuário confirmou impressão do relatório |

## Descrição

Apresenta a tela de impressão para o usuário configurar o relatório.

O método PrintDialog executa o método Print quando o usuário confirma a impressão, executando o bloco de código informado pelo parametro bAction nos métodos New ou SetAction.

## PrintText

### Sintaxe

```
oObj:PrintText ( < cText > , [ nRow ] , [ nCol ] ) --> NIL
```

### Parâmetros

| Argumento | Tipo     | Descrição  |
|-----------|----------|--|
| cText     | Caracter | Texto que será impresso  |
| nRow      | Numérico | Linha que o texto será impresso.<br>Valor padrão: linha posicionada da impressão   |
| nCol      | Numérico | Coluna que o texto será impresso.<br>Valor padrão: coluna posicionada da impressão |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Imprime o texto nas coordenadas informadas pelos parametros

## Printing

### Sintaxe

oObj:Printing ( ) --> IPrinting

## Retorno

| Tipo   | Descrição   |
|--------|---|
| Lógico | Verdadeiro se relatório esta no processo de impressão |

## Descrição

Retorna se relatório esta em processo de impressão.

O status de impressão é definido pelos métodos Init (que indica o início da impressão) e o método Finish (que indica a finalização).

## ReportName

### Sintaxe

oObj:ReportName ( ) --> cReport

## Retorno

| Tipo     | Descrição         |
|----------|-------------------|
| Caracter | Nome do relatório |

## Descrição

Retorna o nome do relatório (exemplo: MATR010)

## Row

### Sintaxe

oObj:Row ( ) --> nRow

## Retorno

| Tipo     | Descrição                      |
|----------|--------------------------------|
| Numérico | Linha posicionada na impressão |

## Descrição

Retorna a linha posicionada na impressão

## Say

### Sintaxe

oObj:Say ( < nRow > , < nCol > , < cText > , [ oFont ] , [ nWidth ] , [ nClrText ] , [ nBkMode ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição                                |
|-----------|----------|--|
| nRow      | Numérico | Linha para impressão do texto            |
| nCol      | Numérico | Coluna para impressão do texto           |
| cText     | Caracter | Texto que será impresso                  |
| oFont     | Objeto   | Objeto da classe TFont                   |
| nWidth    | Numérico | Tamanho em pixel do texto para impressão |
| nClrText  | Numérico | Cor da fonte                             |
| nBkMode   | Numérico | Cor de fundo                             |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Imprime texto nas coordenadas informadas pelos parametros, utilizando as especificações do objeto TFont, cor da fonte e cor de fundo

## SayBitmap

### Sintaxe

oObj:SayBitmap ( < nLinha > , < nCol > , < cBitmap > , [ nWidth ] , [ nHeight ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição   |
|-----------|----------|---|
| nLinha    | Numérico | Linha para impressão da imagem                                |
| nCol      | Numérico | Coluna para impressão da imagem                               |
| cBitmap   | Numérico | Nome da imagem (arquivo ou resource compilado no repositório) |
| nWidth    | Numérico | Largura da imagem   |
| nHeight   | Numérico | Altura da imagem  |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Imprime uma imagem nas coordenadas informadas pelos parametros

## Section

### Sintaxe

oObj:Section ( < uSection > ) --> oSection

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| uSection  | (Qualquer) | Tipo Caracter: Título do objeto<br>Tipo Numérico: Índice do objeto segundo a ordem de criação dos componentes TRSection |

## Retorno

| Tipo   | Descrição  |
|--------|--|
| Objeto | Objeto da classe TRFunction.<br>Se não for encontrado o objeto segundo o parametro informado será retornado Nulo |

## Descrição

Retorna objeto da classe TRSection (seção) baseado no parametro informado.

## SetAction

### Sintaxe

oObj:SetAction ( < bAction > ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| bAction   | Code-Block | Bloco de código que será executado quando usuário confirmar a impressão do relatório |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define o bloco de código que será executado quando o usuário confirmar a impressão

## SetDescription

### Sintaxe

oObj:SetDescription ( < cDesc > ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição               |
|-----------|----------|-------------------------|
| cDesc     | Caracter | Descrição do componente |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a descrição do componente.

## SetLandscape

### Sintaxe

oObj:SetLandscape ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a orientação de página do relatório como paisagem

## SetMeter

### Sintaxe

oObj:SetMeter ( < nTotal > ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição                       |
|-----------|----------|---------------------------------|
| nTotal    | Numérico | Total da regua de processamento |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define o total da regua da tela de processamento do relatório.

## SetParam

### Sintaxe

```
oObj:SetParam ( < uParam > ) --> NIL
```

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| uParam    | (Qualquer) | Tipo Caracter: Parametros do relatório cadastrado no Dicionário de Perguntas (SX1)<br>Tipo Bloco de código: bloco que será utilizado como parametro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define os parametros que serão utilizados pelo usuário para configurar o relatorio.

## SetPortrait

### Sintaxe

```
oObj:SetPortrait ( ) --> NIL
```

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define orientação de página do relatório como retrato

## SetPreview

### Sintaxe



oObj:SetPreview ( [ IPreview ] ) --> NIL

#### Parâmetros

| Argumento | Tipo   | Descrição  |
|-----------|--------|--|
| IPreview  | Lógico | Verdadeiro para apresentar preview no final da impressão<br>Valor padrão: Verdadeiro |

#### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

#### Descrição

Define se será apresentado o preview o relatório no final da impressão

## SetTitle

#### Sintaxe

oObj:SetTitle ( < cTitle > ) --> NIL

#### Parâmetros

| Argumento | Tipo     | Descrição            |
|-----------|----------|----------------------|
| cTitle    | Caracter | Título do componente |

#### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

#### Descrição

Define o título do componente.

## SetTotalInLine

#### Sintaxe

oObj:SetTotalInLine ( [ ITotalInLine ] ) --> NIL

#### Parâmetros

| Argumento    | Tipo   | Descrição  |
|--------------|--------|--|
| ITotalInLine | Lógico | Verdadeiro se totalizadores serão impressos em linha<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se os totalizadores serão impressos em linha ou coluna

## SetTotalText

### Sintaxe

oObj:SetTotalText ( < cTexto > ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição                                    |
|-----------|----------|--|
| cTexto    | Caracter | Texto para impressão antes dos totalizadores |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define o texto que será impresso antes da impressão dos totalizadores

## SkipLine

### Sintaxe

oObj:SkipLine ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Salta uma linha baseado na altura da linha informada pelo usuário

## ThinLine

### Sintaxe

oObj:ThinLine ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Desenha uma linha simples iniciando na linha atual de impressão com a largura da página

## Title

### Sintaxe

oObj:Title ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna o título do componente

## TotalInLine

### Sintaxe

oObj:TotalInLine ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna se a impressão dos totalizadores será em linha

## TotalText

### Sintaxe

oObj:TotalText ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

**Descrição**

Retorna o texto que será impresso antes da impressão dos totalizadores

# Exemplos

## Exemplo de utilização da função MPReport

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```
#include "protheus.ch"

User Function MyReport1()

//Informando o vetor com as ordens utilizadas pelo relatório

MPReport("MYREPORT1","SA1","Relacao de Clientes","Este relatório irá imprimir a relacao de clientes",{ "Por
Codigo","Alfabetica","Por "+RTrim(RetTitle("A1_CGC"))})
Return

User Function MyReport2()

//Informando para função carregar os índices do Dicionário de Índices (SIX) da tabela

MPReport("MYREPORT2","SA1","Relacao de Clientes","Este relatório irá imprimir a relacao de clientes",,.T.)
Return
```

## TReport sem usar método TRSection:Print

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```
#include "protheus.ch"

User Function MyReport()
Local oReport

If TRepInUse() //verifica se relatorios personalizaveis esta disponivel
    Pergunte("MTR025",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection

oReport := TReport():New("MATR025","Relacao das Sugestoes de Orcamentos","MTR025",{ |oReport|
PrintReport(oReport)},"Este relatorio ira imprimir a relacao das Sugestoes de Venda conforme os parametros
solicitados.")

oSection := TRSection():New(oReport,OemToAnsi("Sugestoes de Orcamentos"),{"SBG","SBH","SB1"})

TRCell():New(oSection,"BG_PRODUTO","SBG")
TRCell():New(oSection,"B1_DESC","SB1")
```

```

TRCell():New(oSection,"BG_GERAPV","SBG")
TRCell():New(oSection,"BG_GERAOP","SBG")
TRCell():New(oSection,"BG_GERAOP1","SBG")
TRCell():New(oSection,"BG_GERAEMP","SBG")
TRCell():New(oSection,"BH_SEQUENC","SBH")
TRCell():New(oSection,"BH_CODCOMP","SBH")
TRCell():New(oSection,"B1_DESC","SB1")
TRCell():New(oSection,"BH_QUANT","SBH")

Return oReport

Static Function PrintReport(oReport)
Local oSection := oReport:Section(1)

DbSelectArea("SBG")
DbSetOrder(1)
DbSeek(xFilial()+MV_PAR01,.T.)

oReport:SetMeter(RecCount())

While ( !Eof() .And. xFilial("SBG") == SBG->BG_FILIAL .And. ;
        SBG->BG_PRODUTO >= MV_PAR01 .And. ;
        SBG->BG_PRODUTO <= MV_PAR02 )

    If oReport:Cancel()
        Exit
    EndIf

    DbSelectArea("SB1")
    DbSetOrder(1)
    DbSeek(xFilial("SB1")+SBG->BG_PRODUTO)

    oSection:Init()

    oSection:Cell("B1_DESC"):SetValue(SB1->B1_DESC)

    oSection:Cell("BG_PRODUTO"):Show()
    oSection:Cell("B1_DESC"):Show()
    oSection:Cell("BG_GERAPV"):Show()
    oSection:Cell("BG_GERAOP"):Show()
    oSection:Cell("BG_GERAOP1"):Show()
    oSection:Cell("BG_GERAEMP"):Show()

    dbSelectArea("SBH")
    dbSetOrder(1)
    dbSeek(xFilial("SBH")+SBG->BG_PRODUTO,.T.)

    While ( !Eof() .And. SBH->BH_FILIAL == xFilial("SBH") .And.;
            SBH->BH_PRODUTO == SBG->BG_PRODUTO )

        dbSelectArea("SB1")
        dbSetOrder(1)
        dbSeek(xFilial("SB1")+SBH->BH_CODCOMP)

        oSection:PrintLine()

        oSection:Cell("BG_PRODUTO"):Hide()
        oSection:Cell("B1_DESC"):Hide()
        oSection:Cell("BG_GERAPV"):Hide()
        oSection:Cell("BG_GERAOP"):Hide()
        oSection:Cell("BG_GERAOP1"):Hide()
        oSection:Cell("BG_GERAEMP"):Hide()

        dbSelectArea("SBH")
        dbSkip()

```

```

End

DbSelectArea("SBG")
DbSkip()

oSection:Finish()

oReport:SkipLine()
oReport:IncMeter()

End
Return

```

## TReport sem usar método TRSection:Print com seções

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```

#include "protheus.ch"

User Function MyReport()
Local oReport

If TRRepInUse() //verifica se relatorios personalizaveis esta disponivel
    Pergunte("MTR025",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection1
Local oSection2

oReport := TReport():New("MATR025","Relacao das Sugestoes de Orcamentos","MTR025",{oReport|
PrintReport(oReport)},"Este relatorio ira imprimir a relacao das Sugestoes de Venda conforme os parametros
solicitados.")

oSection1 := TRSection():New(oReport,"Produtos",{ "SBG","SB1"})

TRCell():New(oSection1,"BG_PRODUTO","SBG")
TRCell():New(oSection1,"B1_DESC","SB1")
TRCell():New(oSection1,"BG_GERAPV","SBG")
TRCell():New(oSection1,"BG_GERAOP","SBG")
TRCell():New(oSection1,"BG_GERAOP1","SBG")
TRCell():New(oSection1,"BG_GERAEMP","SBG")

TRPosition(oSection1,"SB1",1,{|| xFilial("SB1") + SBG->BG_PRODUTO})

oSection2 := TRSection():New(oSection1,"Componentes",{ "SBH","SB1"})

TRCell():New(oSection2,"BH_SEQUENC","SBH")
TRCell():New(oSection2,"BH_CODCOMP","SBH")
TRCell():New(oSection2,"B1_DESC","SB1")
TRCell():New(oSection2,"BH_QUANT","SBH")

TRPosition(oSection2,"SB1",1,{|| xFilial("SB1") + SBH->BH_CODCOMP})

Return oReport

Static Function PrintReport(oReport)

```

```

Local oSection1 := oReport:Section(1)
Local oSection2 := oReport:Section(1):Section(1)

DbSelectArea("SBG")
DbSetOrder(1)
DbSeek(xFilial()+MV_PAR01,.T.)

oReport:SetMeter(RecCount())

While ( !Eof() .And. xFilial("SBG") == SBG->BG_FILIAL .And. ;
        SBG->BG_PRODUTO >= MV_PAR01 .And. ;
        SBG->BG_PRODUTO <= MV_PAR02 )

    If oReport:Cancel()
        Exit
    EndIf

    oSection1:Init()
    oSection1:PrintLine()

    dbSelectArea("SBH")
    dbSetOrder(1)
    dbSeek(xFilial("SBH")+SBG->BG_PRODUTO,.T.)

    oSection2:Init()

    While ( !Eof() .And. SBH->BH_FILIAL == xFilial("SBH") .And.;
            SBH->BH_PRODUTO == SBG->BG_PRODUTO )

        oSection2:PrintLine()

        dbSelectArea("SBH")
        dbSkip()

    End

    oSection2:Finish()

    DbSelectArea("SBG")
    DbSkip()

    oSection1:Finish()

    oReport:IncMeter()

End
Return

```

## TReport usando método TRSection:Print com 1 seção

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```

#include "protheus.ch"

User Function MyReport()
Local oReport

If TRepInUse()
    Pergunte("MTR530",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()

```



```

Local oReport
Local oSection
Local oBreak

oReport := TReport():New("MYREPORT","Relatorio de Visitas","MTR530",{oReport| PrintReport(oReport)},"Relatorio
de visitas de vendedores nos clientes")

oSection := TRSection():New(oReport,"Clientes",{ "SA1", "SA3"})

TRCell():New(oSection,"A1_VEND","SA1")
TRCell():New(oSection,"A3_NOME","SA3")
TRCell():New(oSection,"A1_COD","SA1","Cliente")
TRCell():New(oSection,"A1_LOJA","SA1")
TRCell():New(oSection,"A1_NOME","SA1")
TRCell():New(oSection,"A1_ULTVIS","SA1")
TRCell():New(oSection,"A1_TEMVIS","SA1")
TRCell():New(oSection,"A1_CONTATO","SA1")
TRCell():New(oSection,"A1_TEL","SA1")

oBreak := TRBreak():New(oSection,oSection:Cell("A1_VEND"),"Sub Total Vendedores")

TRFunction():New(oSection:Cell("A1_COD"),NIL,"COUNT",oBreak)
TRFunction():New(oSection:Cell("A1_TEMVIS"),NIL,"SUM",oBreak)

Return oReport

Static Function PrintReport(oReport)
Local oSection := oReport:Section(1)
Local cPart
Local cFiltro := ""

#ifdef TOP

    //Transforma parametros do tipo Range em expressao SQL para ser utilizada na query
    MakeSqlExpr("MTR530")

    oSection:BeginQuery()

    If ( mv_par03 == 1 )
        cPart := "%AND (" + Dtos(dDataBase) + " - A1_ULTVIS) > A1_TEMVIS%"
    Else
        cPart := "%%"
    EndIf

    BeginSql alias "QRYSA1"
        SELECT A1_COD,A1_LOJA,A1_NOME,A1_VEND,A1_ULTVIS,A1_TEMVIS,A1_TEL,A1_CONTATO,A3_NOME
        FROM %table:SA1% SA1,%table:SA3% SA3
        WHERE A1_VEND = A3_COD AND A1_FILIAL = %xfilial:SA1% AND
            A1_TEMVIS > 0 AND SA1.%notDel% %exp:cPart%
        ORDER BY A1_VEND
    EndSql

    /*
    Prepara relatorio para executar a query gerada pelo Embedded SQL passando como
    parametro a pergunta ou vetor com perguntas do tipo Range que foram alterados
    pela funcao MakeSqlExpr para serem adicionados a query
    */
    oSection:EndQuery(mv_par04)

#else

    //Transforma parametros do tipo Range em expressao ADVPL para ser utilizada no filtro
    MakeAdvplExpr("MTR530")

    //Adiciona no filtro o parametro tipo Range
    If !Empty(mv_par04)

```

```

        cFiltro += mv_par04 + " .AND. "

    EndIf

    cFiltro += " A1_TEMVIS > 0 "

    If ( mv_par03 == 1 )
        cFiltro += ' .AND. ('+DtoC(dDataBase)+'-A1_ULTVIS) > A1_TEMVIS '
    EndIf

    oSection:SetFilter(cFiltro,"A1_VEND")

    TRPosition():New(oReport:Section(1),"SA3",1,{|| xFilial() + SA1->A1_VEND})

#ENDIF

oSection:Print()

Return

```

## TReport usando método TRSection:SetParentQuery

Para utilizar o exemplo abaixo verifique se o seu repositório está com Release 4 do Protheus

```

#include "protheus.ch"

User Function MyReport()
Local oReport

If TRepInUse()
    Pergunte("MTR530",.F.)

    oReport := ReportDef()
    oReport:PrintDialog()
EndIf
Return

Static Function ReportDef()
Local oReport
Local oSection1
Local oSection2

oReport := TReport():New("MYREPORT","Relatorio de Visitas","MTR530",{oReport| PrintReport(oReport)},"Relatorio
de visitas de vendedores nos clientes")

oSection1 := TRSection():New(oReport,"Vendedores","SA3")

TRCell():New(oSection1,"A3_COD","SA3","Vendedor")
TRCell():New(oSection1,"A3_NOME","SA3")

TRFunction():New(oSection1:Cell("A3_COD"),NIL,"COUNT",NIL,NIL,NIL,NIL,.F.)

oSection2 := TRSection():New(oSection1,"Clientes","SA1")

TRCell():New(oSection2,"A1_COD","SA1","Cliente")
TRCell():New(oSection2,"A1_LOJA","SA1")
TRCell():New(oSection2,"A1_NOME","SA1")
TRCell():New(oSection2,"A1_ULTVIS","SA1")
TRCell():New(oSection2,"A1_TEMVIS","SA1")
TRCell():New(oSection2,"A1_CONTATO","SA1")
TRCell():New(oSection2,"A1_TEL","SA1")

TRFunction():New(oSection2:Cell("A1_COD"),NIL,"COUNT")
TRFunction():New(oSection2:Cell("A1_TEMVIS"),NIL,"SUM")

```

```

Return oReport

Static Function PrintReport(oReport)
Local oSection1 := oReport:Section(1)
Local oSection2 := oReport:Section(1):Section(1)
Local cFiltro   := ""

#IFDEF TOP

    oSection1:BeginQuery()

    If ( mv_par03 == 1 )
        cPart := "%AND (" + Dtos(dDataBase) + " - A1_ULTVIS) > A1_TEMVIS%"
    Else
        cPart := "%%"
    EndIf

    BeginSql alias "QRYSA3"
        SELECT A1_COD,A1_LOJA,A1_NOME,A1_VEND,A1_ULTVIS,A1_TEMVIS,A1_TEL,A1_CONTATO,A3_NOME,A3_COD
        FROM %table:SA1% SA1,%table:SA3% SA3
        WHERE A1_VEND = A3_COD AND A1_FILIAL = %xfilial:SA1% AND
              A1_TEMVIS > 0 AND SA1.%notDel%
        ORDER BY A1_VEND
    EndSql

    /*
    Prepara relatorio para executar a query gerada pelo Embedded SQL passando como
    parametro a pergunta ou vetor com perguntas do tipo Range que foram alterados
    pela funcao MakeSqlExpr para serem adicionados a query
    */
    oSection1:EndQuery(mv_par04)

    oSection2:SetParentQuery()
    oSection2:SetParentFilter({|cParam| QRYSA3->A1 VEND >= cParam .and. QRYSA3->A1 VEND <= cParam},{||
QRYSA3->A3_COD})

#ELSE

    cFiltro := " A1_TEMVIS > 0 "

    If ( mv_par03 == 1 )
        cFiltro += ' .AND. ('+Dtos(dDataBase)+'-A1_ULTVIS) > A1_TEMVIS '
    EndIf

    oSection2:SetFilter(cFiltro,"A1_VEND")
    oSection2:SetRelation({|| SA3->A3_COD})
    oSection2:SetParentFilter({|cParam| SA1->A1 VEND >= cParam .and. SA1->A1 VEND <= cParam},{|| SA3-
>A3_COD})

#ENDIF

oSection1:Print()

Return

```

# TRSection

## Descrição

Seção de um relatório que utiliza a classe TReport.

A classe TRSection pode ser entendida como um layout do relatório, por conter células, quebras e totalizadores que darão um formato para sua impressão.

Com a classe TRSection é possível definir uma query, filtro ou índice com filtro (IndRegua) que será utilizada por ela para processamento do relatório, através do método Print e utilizando as células de posicionamento (TRPosition) e componente de posicionamento de seção (TRPosSection), sem a necessidade de escrever um fonte para processar os resultados da query, filtro ou índice com filtro.

## Métodos

| Método               | Descrição  |
|----------------------|--|
| <u>New</u>           | <p>Método construtor da classe TRSection</p> <p>No parametro onde é informado as tabelas utilizadas pela seção, a primeira será a principal, sendo utilizada para o processamento pelo método Print. As outras tabelas serão utilizadas para que o usuário possa incluir os campos como célula.</p> <p>O relacionamento entre as tabelas poderá ser feito pela query ou então pelo uso dos objetos da classe TRPosition.</p> <p>Dos elementos do parametro onde são informadas as ordens utilizada pela seção serão criados os objetos da classe TROrder automaticamente, utilizando como índice da tabela principal a ordem do elemento no vetor.</p> |
| <u>Parent</u>        | Retorna o objeto pai da classe TRSection   |
| <u>Report</u>        | Retorna o objeto da classe TReport que a classe TRSection pertence   |
| <u>PageBreak</u>     | Retorna se salta a página na quebra de seção   |
| <u>HeaderBreak</u>   | Retorna se cabeçalho das células será impresso após uma quebra (TRBreak)   |
| <u>HeaderSection</u> | Retorna se cabeçalho das células será impresso na quebra de seção  |
| <u>Visible</u>       | Retorna se o componente esta habilitada para impressão   |
| <u>Alias</u>         | Retorna o alias utilizado pela query da seção, que foi definida pelo Embedded SQL com os métodos BeginQuery e EndQuery   |
| <u>Show</u>          | Habilita a impressão do componente   |
| <u>Hide</u>          | Desabilita a impressão da seção, porém todas as linhas serão processadas, somente não aparecerão no relatório.   |
| <u>IReadOnly</u>     | Retorna se o usuário pode alterar o layout da seção  |
| <u>Init</u>          | <p>Inicializa seção.</p> <p>Não é necessário executar o método Init se for utilizar o método Print, já que este faz o controle de inicialização e finalização das seções.</p> <p>Caso não esteja utilizando o método Print, sempre execute o método Init para inicializar a seção, para controle de quebra de página entre seções,</p>   |

|                         |   |
|-------------------------|---|
|                         | impressão do cabeçalho das células e para outros tratamentos internos do componente.  |
| <u>Finish</u>           | <p>Finaliza seção inicializada pelo método Init.</p> <p>Não é necessário executar o método Finish se for utilizar o método Print, já que este faz o controle de inicialização e finalização das seções.</p> <p>Caso não esteja utilizando o método Print, sempre execute o método Finish ao finalizar uma seção, para controle da impressão dos totalizadores das quebras e dos que estão habilitados para o final da seção, para controle de quebra de página entre seções e para outros tratamentos internos do componente.</p>                   |
| <u>PrintLine</u>        | Imprime a linha baseado nas células selecionadas pelo usuário   |
| <u>Print</u>            | <p>Processa as informações da tabela principal ou da query definida pelo Embedded SQL com os métodos BeginQuery e EndQuery</p> <p>O método Print faz o loop na tabela ou na query, executando o método PrintLine para cada linha e o método Print de todas as seções filhas. Nas seções filhas, o posicionamento inicial pode ser baseado na query da seção pai (através do método SetParentQuery) ou através da regra informada pelo método SetRelation, junto com a regra informada pelo método SetParentFilter que controla a saída do loop.</p> |
| <u>Cell</u>             | Retorna objeto da classe TRCell (célula) baseado no parametro informado.  |
| <u>Order</u>            | Retorna objeto da classe TROrder (ordem) basead no parametro informado  |
| <u>LoadCells</u>        | Carrega campos do Dicionário de Campos (SX3) como células da seção, respeitando nível, uso e contexto do campo  |
| <u>LoadOrder</u>        | Carrega índices do Dicionário de Índices (SIX) como ordem da seção  |
| <u>SetPageBreak</u>     | Define se salta a página na quebra de seção   |
| <u>SetHeaderBreak</u>   | Define se imprime cabeçalho das células após uma quebra (TRBreak)   |
| <u>SetHeaderSection</u> | Define de imprime cabeçalho das células na quebra de seção  |
| <u>SetReadOnly</u>      | <p>Define se o usuário não poderá alterar o layout da seção, ou seja, não poderá remover as células pré-definidas.</p> <p>As células que forem adicionadas na seção serão impressas na próxima página, após ter impresso as células pré-definidas.</p>  |
| <u>GetQuery</u>         | Retorna a query que foi definida pelo Embedded SQL com os métodos BeginQuery e EndQuery, já com as células selecionadas e filtro criado pelo usuário e as perguntas tipo Range incluídas.   |
| <u>BeginQuery</u>       | Indica que será utilizado o Embedded SQL para criação de uma nova query que será utilizada pela seção   |
| <u>EndQuery</u>         | <p>Indica a query criada utilizando o Embedded SQL para a seção.</p> <p>O método EndQuery ira juntar na query as células selecionadas e o filtro criado pelo usuário, mais a expressão das perguntas do tipo Range, que foram convertidas anteriormente pela função MakeSQLExpr.</p> <p>Após sua execução a query esta pronta para ser utilizada tanto pela classe TRSection como para o fonte que esta criando os componentes, sem a necessidade de utilizar TCGenQuery, ChangeQuery e TCSetField.</p>   |
| <u>SetParentQuery</u>   | Define se a seção filha utilizara a query da seção pai no processamento do método Print   |
| <u>SetParentFilter</u>  | Define a regra de saída do loop de processamento do método Print das seções filhas  |

|                    |   |
|--------------------|---|
| <u>SetFilter</u>   | Define um filtro ou um índice com filtro (IndRegua) para a tabela principal da seção.<br><br>Se não informado o parametro com a chave de índice, será executado SET FILTER TO com o primeiro parametro, senão será criado um índice com filtro (IndRegua) e sua ordem será retornada pelo método GetIdxOrder    |
| <u>SetOrder</u>    | Define a ordem (TROrder) que será utilizada pela seção  |
| <u>SetIdxOrder</u> | Define uma ordem de índice para a tabela principal  |
| <u>GetOrder</u>    | Retorna a ordem selecionada pelo usuário  |
| <u>GetIdxOrder</u> | Retorna o índice utilizado pela tabela principal da seção para processamento pelo método Print.<br><br>Se o retorno for zero, então a seção esta utilizando uma ordem selecionada pelo usuário (TROrder)  |
| <u>SetRelation</u> | Define a fórmula de relacionamento de uma seção filha com sua seção pai, caso não utilize a query da seção pai (SetParentQuery)<br><br>O método SetRelation irá criar o componente TRRelation apresentado no documento 'Estrutura do componente TReport', que faz o controle do relacionamento entre as seções. |

## Alias

### Sintaxe

oObj:Alias ( ) --> cAlias

### Retorno

| Tipo     | Descrição                  |
|----------|----------------------------|
| Caracter | Alias utilizado pela query |

### Descrição

Retorna o alias utilizado pela query da seção, que foi definida pelo Embedded SQL com os métodos BeginQuery e EndQuery

## BeginQuery

### Sintaxe

oObj:BeginQuery ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Indica que será utilizado o Embedded SQL para criação de uma nova query que será utilizada pela seção

## Cell

### Sintaxe

oObj:Cell ( < uCell > ) --> oCell

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| uCell     | (Qualquer) | Tipo Caracter: Nome ou título do objeto<br>Tipo Numérico: Índice do objeto segundo a ordem de criação dos componentes TRCell |

### Retorno

| Tipo   | Descrição  |
|--------|--|
| Objeto | Objeto da classe TRCell.<br>Se não for encontrado o objeto segundo o parametro informado será retornado Nulo |

### Descrição

Retorna objeto da classe TRCell (célula) baseado no parametro informado.

## EndQuery

### Sintaxe

oObj:EndQuery ( [ uParam ] ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| uParam    | (Qualquer) | Pergunta (exemplo: mv_par01) ou lista de perguntas (exemplo: {mv_par01, mv_par02}) do tipo Range que serão incluídas na query |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Indica a query criada utilizando o Embedded SQL para a seção.

O método EndQuery ira juntar na query as células selecionadas e o filtro criado pelo usuário, mais a expressão das perguntas do tipo Range, que foram convertidas anteriormente pela função MakeSQLExpr.

Após sua execução a query esta pronta para ser utilizada tanto pela classe TRSection como para o fonte que esta criando os componentes, sem a necessidade de utilizar TCGenQuery, ChangeQuery e TCSetField.

## Finish

### Sintaxe

oObj:Finish ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Finaliza seção inicializada pelo método Init.

Não é necessário executar o método Finish se for utilizar o método Print, já que este faz o controle de inicialização e finalização das seções.

Caso não esteja utilizando o método Print, sempre execute o método Finish ao finalizar uma seção, para controle da impressão dos totalizadores das quebras e dos que estão habilitados para o final da seção, para controle de quebra de página entre seções e para outros tratamentos internos do componente.

## GetIdxOrder

### Sintaxe

oObj:GetIdxOrder ( ) --> nIdxOrder

### Retorno

| Tipo     | Descrição        |
|----------|------------------|
| Numérico | Índice utilizado |

### Descrição

Retorna o índice utilizado pela tabela principal da seção para processamento pelo método Print.

Se o retorno for zero, então a seção esta utilizando uma ordem selecionada pelo usuário (TROrder)

## GetOrder

### Sintaxe

oObj:GetOrder ( ) --> nOrder

### Retorno



| Tipo     | Descrição                      |
|----------|--------------------------------|
| Numérico | Ordem selecionada pelo usuário |

**Descrição**

Retorna a ordem selecionada pelo usuário

**GetQuery**

**Sintaxe**

oObj:GetQuery ( ) --> cQuery

**Retorno**

| Tipo     | Descrição                  |
|----------|----------------------------|
| Caracter | Query utilizada pela seção |

**Descrição**

Retorna a query que foi definida pelo Embedded SQL com os métodos BeginQuery e EndQuery, já com as células selecionadas e filtro criado pelo usuário e as perguntas tipo Range incluídas.

**HeaderBreak**

**Sintaxe**

oObj:HeaderBreak ( ) --> IHeaderBreak

**Retorno**

| Tipo   | Descrição                       |
|--------|---------------------------------|
| Lógico | Verdadeiro se imprime cabeçalho |

**Descrição**

Retorna se cabeçalho das células será impresso após uma quebra (TRBreak)

**HeaderSection**

**Sintaxe**

oObj:HeaderSection ( ) --> IHeaderSection

**Retorno**

| Tipo   | Descrição                       |
|--------|---------------------------------|
| Lógico | Verdadeiro se imprime cabeçalho |

## Descrição

Retorna se cabeçalho das células será impresso na quebra de seção

## Hide

### Sintaxe

oObj:Hide ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Desabilita a impressão da seção, porém todas as linhas serão processadas, somente não aparecerão no relatório.

## Init

### Sintaxe

oObj:Init ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Inicializa seção.

Não é necessário executar o método Init se for utilizar o método Print, já que este faz o controle de inicialização e finalização das seções.

Caso não esteja utilizando o método Print, sempre execute o método Init para inicializar a seção, para controle de quebra de página entre seções, impressão do cabeçalho das células e para outros tratamentos internos do componente.

## LoadCells

### Sintaxe

oObj:LoadCells ( [ cTable ] , [ aCells ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição   |
|-----------|----------|---|
| cTable    | Caracter | Tabela que será utilizada para carregar os campos. Se não for informada será utilizado todas as tabelas informadas no método New da classe TRSection. |
| aCells    | Array    | Vetor com nome dos campos que serão carregados como células da seção  |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Carrega campos do Dicionário de Campos (SX3) como células da seção, respeitando nível, uso e contexto do campo

## LoadOrder

### Sintaxe

oObj:LoadOrder ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Carrega índices do Dicionário de Índices (SIX) como ordem da seção

## New

### Sintaxe

TRSection():New ( < oParent > , [ cTitle ] , [ uTable ] , [ aOrder ] , [ ILoadCells ] , [ ILoadOrder ] ) --> TRSection

## Parâmetros

| Argumento  | Tipo       | Descrição  |
|------------|------------|--|
| oParent    | Objeto     | Objeto da classe TReport ou TRSection que será o pai da classe TRSection                           |
| cTitle     | Caracter   | Título da seção  |
| uTable     | (Qualquer) | Tabela ou lista de tabelas que serão utilizadas pela seção   |
| aOrder     | Array      | Vetor com a descrição das ordens   |
| ILoadCells | Lógico     | Verdadeiro para carregar os campos do Dicionário de Campos (SX3) das tabelas da seção como células |

|            |        |   |
|------------|--------|---|
|            |        | Valor padrão: Falso   |
| ILoadOrder | Lógico | Verdadeiro para carregar os índices do Dicionário de Índices (SIX)<br>Valor padrão: Falso |

## Retorno

| Tipo   | Descrição                  |
|--------|----------------------------|
| Objeto | Objeto da classe TRSection |

## Descrição

Método construtor da classe TRSection

No parametro onde é informado as tabelas utilizadas pela seção, a primeira será a principal, sendo utilizada para o processamento pelo método Print. As outras tabelas serão utilizadas para que o usuário possa incluir os campos como célula.  
O relacionamento entre as tabelas poderá ser feito pela query ou então pelo uso dos objetos da classe TRPosition.

Dos elementos do parametro onde são informadas as ordens utilizada pela seção serão criados os objetos da classe TROrder automaticamente, utilizando como índice da tabela principal a ordem do elemento no vetor.

## Order

### Sintaxe

oObj:Order ( < uOrder > ) --> oOrder

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| uOrder    | (Qualquer) | Tipo Caracter: Nickname da ordem<br>Tipo Numérico: Índice do objeto segundo a ordem de criação dos componentes TROrder |

## Retorno

| Tipo   | Descrição                |
|--------|--------------------------|
| Objeto | Objeto da classe TROrder |

## Descrição

Retorna objeto da classe TROrder (ordem) basead no parametro informado

## PageBreak

### Sintaxe

oObj:PageBreak ( ) --> IPageBreak

## Retorno

| Tipo   | Descrição                     |
|--------|-------------------------------|
| Lógico | Verdadeiro se quebra a página |

## Descrição

Retorna se salta a página na quebra de seção

## Parent

### Sintaxe

```
oObj:Parent ( ) --> oParent
```

## Retorno

| Tipo   | Descrição                             |
|--------|---------------------------------------|
| Objeto | Objeto da classe TReport ou TRSection |

## Descrição

Retorna o objeto pai da classe TRSection

## Print

### Sintaxe

```
oObj:Print ( ) --> NIL
```

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Processa as informações da tabela principal ou da query definida pelo Embedded SQL com os métodos BeginQuery e EndQuery

O método Print faz o loop na tabela ou na query, executando o método PrintLine para cada linha e o método Print de todas as seções filhas. Nas seções filhas, o posicionamento inicial pode ser baseado na query da seção pai (através do método SetParentQuery) ou através da regra informada pelo método SetRelation, junto com a regra informada pelo método SetParentFilter que controla a saída do loop.

## PrintLine

### Sintaxe

```
oObj:PrintLine ( ) --> NIL
```

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Imprime a linha baseado nas células selecionadas pelo usuário

## Report

### Sintaxe

```
oObj:Report ( ) --> NIL
```

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna o objeto da classe TReport que a classe TRSection pertence

## SetFilter

### Sintaxe

```
oObj:SetFilter ( < cFilter > , [ cIndexKey ] ) --> NIL
```

### Parâmetros

| Argumento | Tipo     | Descrição                            |
|-----------|----------|--------------------------------------|
| cFilter   | Caracter | Expressão de filtro no formato ADVPL |
| cIndexKey | Caracter | Chave de índice                      |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define um filtro ou um índice com filtro (IndRegua) para a tabela principal da seção.

Se não informado o parametro com a chave de índice, será executado SET FILTER TO com o primeiro parametro, senão será criado um índice com filtro (IndRegua) e sua ordem será retornada pelo método GetIdxOrder

# SetHeaderBreak

## Sintaxe

oObj:SetHeaderBreak ( [ IHeaderBreak ] ) --> NIL

## Parâmetros

| Argumento    | Tipo   | Descrição  |
|--------------|--------|--|
| IHeaderBreak | Lógico | Verdadeiro para imprimir o cabeçalho<br>Valor padrão: verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se imprime cabeçalho das células após uma quebra (TRBreak)

# SetHeaderSection

## Sintaxe

oObj:SetHeaderSection ( [ IHeaderSection ] ) --> NIL

## Parâmetros

| Argumento      | Tipo   | Descrição  |
|----------------|--------|--|
| IHeaderSection | Lógico | Verdadeiro para imprimir o cabeçalho<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define de imprime cabeçalho das células na quebra de seção

# SetIdxOrder

## Sintaxe

oObj:SetIdxOrder ( < nIdxOrder > ) --> NIL

## Parâmetros

| Argumento | Tipo     | Descrição       |
|-----------|----------|-----------------|
| nIdxOrder | Numérico | Ordem do índice |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define uma ordem de índice para a tabela principal

# SetOrder

## Sintaxe

oObj:SetOrder ( < nOrdem > ) --> NIL

## Parâmetros

| Argumento | Tipo     | Descrição                |
|-----------|----------|--------------------------|
| nOrdem    | Numérico | Ordem que será utilizada |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a ordem (TROrder) que será utilizada pela seção

# SetPageBreak

## Sintaxe

oObj:SetPageBreak ( [ IPageBreak ] ) --> NIL

## Parâmetros

| Argumento  | Tipo   | Descrição   |
|------------|--------|---|
| IPageBreak | Lógico | Verdadeiro para saltar a página<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo | Descrição |
|------|-----------|
|------|-----------|



(NULO)

Retorno nulo.

## Descrição

Define se salta a página na quebra de seção

## SetParentFilter

### Sintaxe

```
oObj:SetParentFilter ( < bFilter > , [ bParam ] ) --> NIL
```

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| bFilter   | Code-Block | Bloco de código com a regra para saída do loop  |
| bParam    | Code-Block | Bloco de código com a expressão que retornará o valor que será enviado como parametro para o parametro com a regra para saída do loop |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a regra de saída do loop de processamento do método Print das seções filhas

## SetParentQuery

### Sintaxe

```
oObj:SetParentQuery ( [ IParentQuery ] ) --> NIL
```

### Parâmetros

| Argumento    | Tipo   | Descrição   |
|--------------|--------|---|
| IParentQuery | Lógico | Verdadeiro para utilizar query da seção pai<br>Valor padrão: Verdadeiro |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se a seção filha utilizara a query da seção pai no processamento do método Print

# SetReadOnly

## Sintaxe

oObj:SetReadOnly ( [ SetReadOnly ] ) --> NIL

## Parâmetros

| Argumento   | Tipo   | Descrição   |
|-------------|--------|---|
| SetReadOnly | Lógico | Verdadeiro para desabilitar alteração do layout<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se o usuário não poderá alterar o layout da seção, ou seja, não poderá remover as células pré-definidas.

As células que forem adicionadas na seção serão impressas na próxima página, após ter impresso as células pré-definidas.

# SetRelation

## Sintaxe

oObj:SetRelation ( < bFormula > , [ cAlias ] , [ uOrder ] , [ ISeek ] ) --> NIL

## Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| bFormula  | Code-Block | Bloco de código com a expressão para relacionamento entre as seções                                    |
| cAlias    | Caracter   | Tabela utilizada pela fórmula  |
| uOrder    | (Qualquer) | Ordem utilizada tabela<br>Tipo Caracter: Nickname da ordem de índice<br>Tipo Numérico: Ordem do índice |
| ISeek     | Lógico     | Indica se será executado DbSeek com o conteúdo da fórmula<br>Valor padrão: Verdadeiro                  |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a fórmula de relacionamento de uma seção filha com sua seção pai, caso não utilize a query da seção pai (SetParentQuery)

O método SetRelation irá criar o componente TRRelation apresentado no documento 'Estrutura do componente TReport', que faz o controle do relacionamento entre as seções.

## Show

### Sintaxe

oObj:Show ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Habilita a impressão do componente

## Visible

### Sintaxe

oObj:Visible ( ) --> IVisible

### Retorno

| Tipo   | Descrição                                 |
|--------|---|
| Lógico | Verdadeiro se esta habilitado a impressão |

### Descrição

Retorna se o componente esta habilitada para impressão

## IReadOnly

### Sintaxe

oObj:IReadOnly ( ) --> IReadOnly

### Retorno

| Tipo   | Descrição                                       |
|--------|---|
| Lógico | Verdadeiro se usuário não pode alterar o layout |

### Descrição

Retorna se o usuário pode alterar o layout da seção

# TRBreak

## Descrição

Quebra de uma seção de um relatório que utiliza a classe TReport.

A classe TRBreak é utilizada para impressão dos totalizadores da seção antes do final da mesma ou do relatório, dependendo da regra de quebra informada.

## Métodos

| Método          | Descrição  |
|-----------------|--|
| <u>New</u>      | Método construtor da classe TRBreak  |
| <u>SetBreak</u> | Define a regra para a quebra   |
| <u>Execute</u>  | Executa a regra e indica se houve a quebra.<br><br>O método Execute é executado para cada linha impressa pelo método PrintLine da seção que a quebra pertence. |

## Execute

### Sintaxe

oObj:Execute ( ) --> IBreak

### Retorno

| Tipo   | Descrição                    |
|--------|------------------------------|
| Lógico | Indique que houve uma quebra |

## Descrição

Executa a regra e indica se houve a quebra.

O método Execute é executado para cada linha impressa pelo método PrintLine da seção que a quebra pertence.

## New

### Sintaxe

TRBreak():New ( < oParent > , [ uBreak ] , [ cTitle ] , [ lTotalInLine ] ) --> TRBreak

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| oParent   | Objeto     | Objeto da classe TRSection que a quebra pertence |
| uBreak    | (Qualquer) | Regra para quebra                                |

|              |          |  |
|--------------|----------|--|
|              |          | Tipo Objeto: Objeto da classe TRCell<br>Tipo Caracter: Expressão ADVPL para macro execução<br>Tipo Bloco de Código: Bloco de código com expressão que será executada |
| cTitle       | Caracter | Título da quebra   |
| lTotalInLine | Lógico   | Verdadeiro se totalizadores serão impressos em linha<br>Valor padrão: Falso  |

## Retorno

| Tipo   | Descrição                |
|--------|--------------------------|
| Objeto | Objeto da classe TRBreak |

## Descrição

Método construtor da classe TRBreak

## SetBreak

### Sintaxe

oObj:SetBreak ( < uBreak > ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| uBreak    | (Qualquer) | Regra para quebra<br>Tipo Objeto: Objeto da classe TRCell<br>Tipo Caracter: Expressão ADVPL para macro execução<br>Tipo Bloco de Código: Bloco de código com expressão que será executada |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a regra para a quebra

# TRCell

## Descrição

Célula de impressão de uma seção (TRSection) de um relatório que utiliza a classe TReport

## Métodos

| Método              | Descrição   |
|---------------------|---|
| <u>New</u>          | <p>Método construto da classe TRCell.</p> <p>Se o nome da célula informada pelo parametro for encontrado no Dicionário de Campos (SX3), as informações do campo serão carregadas para a célula, respeitando os parametros de título, picture e tamanho. Dessa forma o relatório sempre estará atualizado com as informações do Dicionário de Campos (SX3).</p> <p>O nome da célula será utilizado junto ao alias informado pelo parametro como conteúdo para a impressão da célula, como por exemplo a impressão do campo A1_COD da tabela SA1.</p> <p>Se for informado o parametro com o bloco de código, o retorno deste será utilizado como conteúdo para impressão da célula, com a picture, tamanho e título definidos para a célula. Assim sendo é possível criar células calculadas com formato de impressão baseada no Dicionário de Campos (SX3)</p> |
| <u>Enable</u>       | Habilita impressão da célula  |
| <u>Disable</u>      | Desabilita impressão da célula, ou seja, a célula não será impressa e seu poscionamento não será calculado  |
| <u>Hide</u>         | Desabilita impressão da célula, porém continua calculando seu poscionamento   |
| <u>ColPos</u>       | Retorna a coluna que a célula será impressa pelo método PrintLine da seção que a célula pertence  |
| <u>Execute</u>      | Avalia as informações da célula para a impressão do conteúdo pelo método PrintLine da seção que a célula pertence   |
| <u>GetWidth</u>     | Retorna a largura da célula   |
| <u>GetText</u>      | Retorna o texto que será impresso pelo método PrintLine da seção que a célula pertence  |
| <u>GetFieldInfo</u> | Carrega informações (título, picture, tamanho, etc) do campo informado pelo parametro baseado no Dicionário de Campos (SX3)   |
| <u>SetPicture</u>   | Define a máscara de impressão da célula   |
| <u>SetSize</u>      | Define o tamanho da célula, em pixel ou número de caracteres  |
| <u>SetBlock</u>     | <p>Define o bloco de código que retornará o conteúdo de impressão da célula.</p> <p>Definindo o bloco de código para a célula, esta não utilizara mais o nome mais o alias para retornar o conteúdo de impressão.</p>   |
| <u>SetValue</u>     | <p>Define um valor constante para a célula.</p> <p>Definindo um valor constante para a célula, esta não utilizara o nome mais o alias para impressão do conteúdo, ou seja, a célula passará a se comportar como um váriavel de um programa.</p>   |

|                 |  |
|-----------------|--|
|                 | Se o parametro não for informado (Nulo), então a célula deixará de usar o valor constante, voltando a imprimir seu conteúdo utilizando nome mais alias.  |
| <u>SetAlign</u> | <p>Define o alinhamento da célula na impressão.</p> <p>O alinhamento padrão sem o uso do método SetAlign depende do tipo do conteúdo da célula no momento da impressão:</p> <ul style="list-style-type: none"> <li>Tipo numérico: alinhamento a direita</li> <li>Outros tipos: alinhamento a esquerda</li> </ul>   |
| <u>SetCBox</u>  | <p>Define que a célula irá imprimir a descrição do item de um campo que tenha o formato ComboBox, utilizado no Dicionário de Campos (SX3)</p> <p>O formato do parametro é o mesmo utilizado no Dicionário de Campos (SX3), como no exemplo abaixo:<br/>oCell:SetCBox('1=Sim;2=Não').</p> <p>Na impressão será utilizado a descrição do item. No exemplo acima se o conteúdo da célula for '1', será impresso 'Sim', sendo que o método já ira calcular o tamanho da célula baseado na maior descrição informada.</p> |
| <u>GetValue</u> | Retorna o valor constante da célula definido pelo método SetValue  |

## ColPos

### Sintaxe

oObj:ColPos ( ) --> nCol

### Retorno

| Tipo     | Descrição                     |
|----------|-------------------------------|
| Numérico | Coluna de impressão da célula |

### Descrição

Retorna a coluna que a célula será impressa pelo método PrintLine da seção que a célula pertence

## Disable

### Sintaxe

oObj:Disable ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição



Desabilita impressão da célula, ou seja, a célula não será impressa e seu posicionamento não será calculado

## Enable

### Sintaxe

oObj:Enable ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Habilita impressão da célula

## Execute

### Sintaxe

oObj:Execute ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Avalia as informações da célula para a impressão do conteúdo pelo método PrintLine da seção que a célula pertence

## GetFieldInfo

### Sintaxe

oObj:GetFieldInfo ( < cField > ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição     |
|-----------|----------|---------------|
| cField    | Caracter | Nome do campo |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Carrega informações (título, picture, tamanho, etc) do campo informado pelo parametro baseado no Dicionário de Campos (SX3)

## GetText

### Sintaxe

oObj:GetText ( ) --> cText

### Retorno

| Tipo     | Descrição            |
|----------|----------------------|
| Caracter | Texto para impressão |

## Descrição

Retorna o texto que será impresso pelo método PrintLine da seção que a célula pertence

## GetValue

### Sintaxe

oObj:GetValue ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna o valor constante da célula definido pelo método SetValue

## GetWidth

### Sintaxe

oObj:GetWidth ( ) --> nWidth

### Retorno

| Tipo     | Descrição         |
|----------|-------------------|
| Numérico | Largura da célula |

## Descrição

Retorna a largura da célula

# Hide

## Sintaxe

oObj:Hide ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Desabilita impressão da célula, porém continua calculando seu poscionamento

# New

## Sintaxe

TRCell():New ( < oParent > , < cName > , [ cAlias ] , [ cTitle ] , [ cPicture ] , [ nSize ] , [ IPixel ] , [ bBlock ] ) --> TRCell

## Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| oParent   | Objeto     | Objeto da classe TRSection que a célula pertence    |
| cName     | Caracter   | Nome da célula                                      |
| cAlias    | Caracter   | Alias utilizado pela célula                         |
| cTitle    | Caracter   | Título da célula                                    |
| cPicture  | Caracter   | Máscara de impressão da célula                      |
| nSize     | Numérico   | Tamanho da célula, em pixel ou número de caracteres |
| IPixel    | Lógico     | Indica se tamanho esta sendo informado em pixel     |
| bBlock    | Code-Block | Bloco de código que retornará o conteúdo da célula  |

## Retorno

| Tipo   | Descrição               |
|--------|-------------------------|
| Objeto | Objeto da classe TRCell |

## Descrição

Método construto da classe TRCell.

Se o nome da célula informada pelo parametro for encontrado no Dicionário de Campos (SX3), as informações do campo serão carregadas para a célula, respeitando os parametros de título, picture e tamanho. Dessa forma o relatório sempre estará atualizado com as informações do Dicionário de Campos (SX3).

O nome da célula será utilizado junto ao alias informado pelo parametro como conteúdo para a impressão da célula, como por exemplo a impressão do campo A1\_COD da tabela SA1.  
Se for informado o parametro com o bloco de código, o retorno deste será utilizado como conteúdo para impressão da célula, com a picture, tamanho e título definidos para a célula. Assim sendo é possível criar células calculadas com formato de impressão baseada no Dicionário de Campos (SX3)

## SetAlign

### Sintaxe

oObj:SetAlign ( < nAlign > ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição   |
|-----------|----------|---|
| nAlign    | Caracter | Tipo de alinhamento<br>"LEFT" - esquerda<br>"RIGH" - direita<br>"CENTER" - centro |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Define o alinhamento da célula na impressão.

O alinhamento padrão sem o uso do método SetAlign depende do tipo do conteúdo da célula no momento da impressão:

- Tipo numérico: alinhamento a direita
- Outros tipos: alinhamento a esquerda

## SetBlock

### Sintaxe

oObj:SetBlock ( < bBlock > ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| bBlock    | Code-Block | Bloco de código com expressão que retornará o conteúdo de impressão |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define o bloco de código que retornará o conteúdo de impressão da célula.

Definindo o bloco de código para a célula, esta não utilizará mais o nome mais o alias para retornar o conteúdo de impressão.

## SetCBox

### Sintaxe

```
oObj:SetCBox ( < cCBox > ) --> NIL
```

### Parâmetros

| Argumento | Tipo     | Descrição                                  |
|-----------|----------|--|
| cCBox     | Caracter | Texto com os itens e descrição do ComboBox |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define que a célula irá imprimir a descrição do item de um campo que tenha o formato ComboBox, utilizado no Dicionário de Campos (SX3)

O formato do parametro é o mesmo utilizado no Dicionário de Campos (SX3), como no exemplo abaixo:  
oCell:SetCBox('1=Sim;2=Não').

Na impressão será utilizado a descrição do item. No exemplo acima se o conteúdo da célula for '1', será impresso 'Sim', sendo que o método já irá calcular o tamanho da célula baseado na maior descrição informada.

## SetPicture

### Sintaxe

```
oObj:SetPicture ( < cPicture > ) --> NIL
```

### Parâmetros

| Argumento | Tipo     | Descrição  |
|-----------|----------|--|
| cPicture  | Caracter | Máscara de impressão da célula, no formato ADVPL |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a máscara de impressão da célula

## SetSize

### Sintaxe

oObj:SetSize ( < nSize > , [ IPixel ] ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição  |
|-----------|----------|--|
| nSize     | Numérico | Tamanho da célula  |
| IPixel    | Lógico   | Indica se tamanho esta sendo informado em pixel<br>Valor padrão: Falso |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define o tamanho da célula, em pixel ou número de caracteres

## SetValue

### Sintaxe

oObj:SetValue ( [ uParam ] ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição                            |
|-----------|------------|--------------------------------------|
| uParam    | (Qualquer) | Define valor constante para a célula |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define um valor constante para a célula.

Definindo um valor constante para a célula, esta não utilizara o nome mais o alias para impressão do conteúdo, ou seja, a célula passará a se comportar como um variável de um programa.

Se o parametro não for informado (Nulo), então a célula deixará de usar o valor constante, voltando a imprimir seu conteúdo utilizando nome mais alias.

# TRFunction

## Descrição

Totalizador de uma quebra, seção ou relatório que utiliza a classe TReport.

Um totalizador pode executar uma das seguintes funções abaixo, utilizando como referencia uma célula da seção ou o retorno de uma fórmula definida para ele:

- SUM (somar)
- COUNT (contar)
- MAX (valor máximo)
- MIN (valor mínimo)
- AVERAGE (valor médio)

Na criação do totalizador sempre é informado uma seção (TRSection) a qual ele pertence, e automaticamente o totalizador será incluído no relatório (TReport) que a seção pertence.

## Métodos

| Método              | Descrição   |
|---------------------|---|
| <u>New</u>          | Método construtor da classe TRFunction.<br><br>O parametro que indica qual célula o totalizador se refere ,será utilizado para posicionamento de impressão do totalizador quando estiver definido que a impressão será por coluna e como conteúdo para a função definida caso não seja informada uma fórmula para o totalizador |
| <u>Formula</u>      | Retorna a fórmula do totalizador  |
| <u>SectionValue</u> | Retorna o valor atual do totalizador para a seção   |
| <u>PageValue</u>    | Retorna o valor atual do totalizador para a página  |
| <u>ReportValue</u>  | Retorna o valor atual do totalizador para o relatório   |
| <u>GetFunction</u>  | Retorna a função (SUM, COUNT, MAX, MIN, AVERAGE) utilizada pelo totalizador.  |
| <u>EndSection</u>   | Retorna se o totalizador será impresso na quebra de seção   |
| <u>EndPage</u>      | Retorna se totalizador será impresso no final da página   |
| <u>EndReport</u>    | Retorna se totalizador será impresso no final do relatório  |
| <u>Execute</u>      | Executa a função do totalizador, utilizando a fórmula definida pelo método New ou pelo método SetFormula.<br><br>O método Execute é executado para cada linha impressa pelo método PrintLine da seção que o totalizador pertence.   |
| <u>Reset</u>        | Reinicia valor do totalizador   |
| <u>ResetSection</u> | Reinicia valor do totalizador para a seção  |
| <u>ResetPage</u>    | Reinicia valor do totalizador para a página   |
| <u>ResetReport</u>  | Reinicia valor do totalizador para o relatório  |
| <u>SetFormula</u>   | Define a fórmula utilizada pelo totalizador ao invés do conteúdo da célula que ele se refere  |
| <u>SetBreak</u>     | Define a quebra (TRBreak) que o totalizador será impresso   |
| <u>SetFunction</u>  | Define a função (SUM, COUNT, MAX, MIN, AVERAGE, HOUR) que será  |



|                      |   |
|----------------------|---|
|                      | utilizada pelo totalizador                                  |
| <u>SetEndSection</u> | Define se o totalizador será impresso na quebra de seção    |
| <u>SetEndPage</u>    | Define se o totalizador será impresso no final da página    |
| <u>SetEndReport</u>  | Define se o totalizador será impresso no final do relatório |

## EndPage

### Sintaxe

oObj:EndPage ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Retorna se totalizador será impresso no final da página

## EndReport

### Sintaxe

oObj:EndReport ( ) --> IEndReport

### Retorno

| Tipo   | Descrição  |
|--------|--|
| Lógico | Verdadeiro para imprimir no final do relatório<br>Valor padrão: Verdadeiro |

### Descrição

Retorna se totalizador será impresso no final do relatório

## EndSection

### Sintaxe

oObj:EndSection ( ) --> IEndSection

### Retorno

| Tipo   | Descrição                                |
|--------|--|
| Lógico | Verdadeiro se imprime na quebra de seção |

### Descrição

Retorna se o totalizador será impresso na quebra de seção

## Execute

### Sintaxe

oObj:Execute ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Executa a função do totalizador, utilizando a fórmula definida pelo método New ou pelo método SetFormula.

O método Execute é executado para cada linha impressa pelo método PrintLine da seção que o totalizador pertence.

## Formula

### Sintaxe

oObj:Formula ( ) --> uFormula

### Retorno

| Tipo       | Descrição   |
|------------|---|
| (Qualquer) | - Objeto da classe TRCell ou ;<br>- texto para macro execução ou ;<br>- bloco de código |

### Descrição

Retorna a fórmula do totalizador

## GetFunction

### Sintaxe

oObj:GetFunction ( ) --> cFunction

### Retorno

| Tipo     | Descrição                         |
|----------|-----------------------------------|
| Caracter | Função utilizada pelo totalizador |

## Descrição

Retorna a função (SUM, COUNT, MAX, MIN, AVERAGE) utilizada pelo totalizador.

## New

### Sintaxe

TRFunction():New ( < oParent > , [ cID ] , < cFunction > , [ oBreak ] , [ cTitle ] , [ cPicture ] , [ uFormula ] , [ lEndSection ] , [ lEndReport ] ) --> NIL

### Parâmetros

| Argumento   | Tipo       | Descrição   |
|-------------|------------|---|
| oParent     | Objeto     | Objeto da classe TRCell que o totalizador se refere   |
| cID         | Caracter   | Identificação do totalizador  |
| cFunction   | Caracter   | Função que será utilizada pelo totalizador (SUM, COUNT, MAX, MIN, AVERAGE, HOUR)  |
| oBreak      | Objeto     | Objeto da classe TRBreak que define em qual quebra o totalizador será impresso  |
| cTitle      | Caracter   | Título do totalizador. Se não informado será utilizado o título da célula que o totalizador se refere                           |
| cPicture    | Caracter   | Máscara de impressão do totalizador. Se não informado será utilizado a máscara da célula que o totalizador se refere            |
| uFormula    | (Qualquer) | Tipo Caracter: Expressão ADVPL para macro execução<br>Tipo Bloco de Código: Bloco de Código com a expressão ADVPL para execução |
| lEndSection | Lógico     | Indica se totalizador será impresso na quebra de seção<br>Valor padrão: Verdadeiro  |
| lEndReport  | Lógico     | Indica se totalizador será impresso no final do relatório<br>Valor padrão: Verdadeiro   |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Método construtor da classe TRFunction.

O parametro que indica qual célula o totalizador se refere ,será utilizado para posicionamento de impressão do totalizador quando estiver definido que a impressão será por coluna e como conteúdo para a função definida caso não seja informada uma fórmula para o totalizador

## PageValue

### Sintaxe

oObj:PageValue ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Retorna o valor atual do totalizador para a página

## ReportValue

### Sintaxe

oObj:ReportValue ( ) --> uReportValue

## Retorno

| Tipo       | Descrição                    |
|------------|------------------------------|
| (Qualquer) | Valor atual para o relatório |

## Descrição

Retorna o valor atual do totalizador para o relatório

## Reset

### Sintaxe

oObj:Reset ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Reinicia valor do totalizador

## ResetPage

### Sintaxe

oObj:ResetPage ( ) --> NIL

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Reinicia valor do totalizador para a página

## ResetReport

### Sintaxe

oObj:ResetReport ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Reinicia valor do totalizador para o relatório

## ResetSection

### Sintaxe

oObj:ResetSection ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Reinicia valor do totalizador para a seção

## SectionValue

### Sintaxe

oObj:SectionValue ( ) --> uSectionValue

### Retorno

| Tipo       | Descrição                |
|------------|--------------------------|
| (Qualquer) | Valor atual para a seção |

## Descrição

Retorna o valor atual do totalizador para a seção

# SetBreak

## Sintaxe

oObj:SetBreak ( < oBreak > ) --> NIL

## Parâmetros

| Argumento | Tipo   | Descrição                |
|-----------|--------|--------------------------|
| oBreak    | Objeto | Objeto da classe TRBreak |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a quebra (TRBreak) que o totalizador será impresso

# SetEndPage

## Sintaxe

oObj:SetEndPage ( [ IEndPage ] ) --> NIL

## Parâmetros

| Argumento | Tipo   | Descrição   |
|-----------|--------|---|
| IEndPage  | Lógico | Verdadeiro para imprimir no final da página<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se o totalizador será impresso no final da página

# SetEndReport

## Sintaxe

oObj:SetEndReport ( [ IEndReport ] ) --> NIL

## Parâmetros

| Argumento  | Tipo   | Descrição  |
|------------|--------|--|
| IEndReport | Lógico | Verdadeiro para imprimir no final do relatório<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se o totalizador será impresso no final do relatório

## SetEndSection

### Sintaxe

oObj:SetEndSection ( [ IEndSection ] ) --> NIL

### Parâmetros

| Argumento   | Tipo   | Descrição   |
|-------------|--------|---|
| IEndSection | Lógico | Verdadeiro para imprimir na quebra de seção<br>Valor padrão: Verdadeiro |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define se o totalizador será impresso na quebra de seção

## SetFormula

### Sintaxe

oObj:SetFormula ( < uFormula > ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição   |
|-----------|------------|---|
| uFormula  | (Qualquer) | Tipo Caracter: Expressão ADVPL para macro execução<br>Tipo Bloco de Código: Bloco de Código com a expressão ADVPL para execução |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

**Descrição**

Define a fórmula utilizada pelo totalizador ao invés do conteúdo da célula que ele se refere

**SetFunction**

**Sintaxe**

oObj:SetFunction ( < cFunction > ) --> NIL

**Parâmetros**

| Argumento | Tipo     | Descrição                                  |
|-----------|----------|--|
| cFunction | Caracter | Função que será utilizada pelo totalizador |

**Retorno**

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

**Descrição**

Define a função (SUM, COUNT, MAX, MIN, AVERAGE, HOUR) que será utilizada pelo totalizador



# TRPosition

## Descrição

Célula de posicionamento de uma seção de um relatório que utiliza a class TReport.

As células de posicionamento são executas em toda execução do método PrintLine da seção que elas pertencem, para posicionar as tabelas secundárias da seção que o usuário pode utilizar para incluir células baseadas em seus campos.

## Métodos

| Método             | Descrição  |
|--------------------|--|
| <u>New</u>         | Método construtor da classe TRPosition   |
| <u>SetTable</u>    | Define a tabela que será utilizada pela fórmula para o posicionamento  |
| <u>SetOrder</u>    | Define a ordem do índice para o posicionamento   |
| <u>SetNickName</u> | Define a ordem do índice pelo seu nickname   |
| <u>Execute</u>     | Executa a fórmula de posicionamento definida no parametro do método New ou pelo método SetFormula            |
| <u>Enable</u>      | Habilita execução da célula de posicionamento  |
| <u>Disable</u>     | Desabilita execução da célula de posicionamento  |
| <u>SetFormula</u>  | Define a fórmula de posicionamento da tabela informada por parametro pelo método New ou pelo método SetTable |

## Disable

### Sintaxe

oObj:Disable ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Desabilita execução da célula de posicionamento

## Enable

### Sintaxe

oObj:Enable ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Habilita execução da célula de posicionamento

## Execute

### Sintaxe

oObj:Execute ( ) --> NIL

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Executa a fórmula de posicionamento definida no parametro do método New ou pelo método SetFormula

## New

### Sintaxe

TRPosition():New ( < oParent > , [ cAlias ] , [ uOrder ] , [ uFormula ] , [ lSeek ] ) --> TRPosition

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| oParent   | Objeto     | Objeto da classe TRSection que a célula pertence   |
| cAlias    | Caracter   | Tabela que será utilizada para posicionamento  |
| uOrder    | (Qualquer) | Tipo Caracter: Nickname da ordem do índice da tabela<br>Tipo Numérico: Ordem do índice da tabela   |
| uFormula  | (Qualquer) | Fórmula de posicionamento<br>Tipo Caracter: Expressão ADVPL para macro execução<br>Tipo Bloco de Código: Bloco de Código com a expressão ADVPI para execução |
| lSeek     | Lógico     | Verdadeiro para executar o DbSeek com o conteúdo retornado pela fórmula<br>Valor padrão: Verdadeiro  |

### Retorno

| Tipo   | Descrição                   |
|--------|-----------------------------|
| Objeto | Objeto da classe TRPosition |

## Descrição

Método construtor da classe TRPosition

## SetFormula

### Sintaxe

oObj:SetFormula ( < uFormula > , [ ISeek ] ) --> NIL

### Parâmetros

| Argumento | Tipo       | Descrição  |
|-----------|------------|--|
| uFormula  | (Qualquer) | Fórmula de posicionamento<br>Tipo Caracter: Expressão ADVPL para macro execução<br>Tipo Bloco de Código: Bloco de Código com a expressão ADVPI para execução |
| ISeek     | Lógico     | Verdadeiro para executar o DbSeek com o conteúdo retornado pela fórmula<br>Valor padrão: Verdadeiro  |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Define a fórmula de posicionamento da tabela informada por parametro pelo método New ou pelo método SetTable

## SetNickName

### Sintaxe

oObj:SetNickName ( < cNickname > ) --> NIL

### Parâmetros

| Argumento | Tipo     | Descrição          |
|-----------|----------|--------------------|
| cNickname | Caracter | Nickname do índice |

### Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

### Descrição

Define a ordem do índice pelo seu nickname

# SetOrder

## Sintaxe

oObj:SetOrder ( < nOrder > ) --> NIL

## Parâmetros

| Argumento | Tipo     | Descrição       |
|-----------|----------|-----------------|
| nOrder    | Numérico | Ordem do índice |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a ordem do índice para o posicionamento

# SetTable

## Sintaxe

oObj:SetTable ( < cAlias > ) --> NIL

## Parâmetros

| Argumento | Tipo     | Descrição                  |
|-----------|----------|----------------------------|
| cAlias    | Caracter | Tabela para posicionamento |

## Retorno

| Tipo   | Descrição     |
|--------|---------------|
| (NULO) | Retorno nulo. |

## Descrição

Define a tabela que será utilizada pela fórmula para o posicionamento