

Funções para VT100



COMANDOS E FUNÇÕES PARA VT100

- 1 -

Funções para VT100

Índice

<u>COMANDOS PARA TELNET VT100.</u>	3
<u>@... VTSay</u>	3
<u>VTClear Screen</u>	4
<u>@... VTClear TO</u>	4
<u>@... VTGet</u>	4
<u>VTRead</u>	5
<u>VTSave Screen TO</u>	6
<u>VTSave Screen</u>	6
<u>VTRestore Screen FROM</u>	6
<u>VTRestore Screen</u>	7
<u>VTPause</u>	7
<u>@... VTPause</u>	7
<u>VTSetSize</u>	8
<u>VTSet Key</u>	8
<u>FUNCOES PARA TELNET VT100</u>	9
<u>VTReadVar()</u>	9
<u>VTSave()</u>	9
<u>VTRestore()</u>	9
<u>VTScroll()</u>	10
<u>VTLastKey()</u>	10
<u>VTSetKey()</u>	11
<u>VTKeyBoard()</u>	11
<u>VTRow()</u>	12
<u>VTCol()</u>	12
<u>VTInkey()</u>	12
<u>VTMaxCol()</u>	13
<u>VTMaxRow()</u>	13
<u>VTBeep()</u>	13
<u>VTReverso()</u>	14
<u>VTClearBuffer()</u>	14
<u>VTAlert()</u>	15
<u>VTYesNo()</u>	15
<u>VTChoice()</u>	16
<u>VTABrowse()</u>	17
<u>VTDBBrowse()</u>	19
<u>VTModelo()</u>	20
<u>Exemplo aplicação Telnet VT100</u>	21

- 1 -

Funções para VT100

COMANDOS PARA TELNET VT100.

@...VTSay

Tipo: TELNET VT100

Exibe dados em uma linha e coluna especificadas

Sintaxe

@ <nLin>, <nCol> [VTSAY <exp> [PICTURE <cSayPicture>]]

Parâmetros

<nLin> e <nCol> são as coordenadas de linha e coluna da saída.

Os valores de linha podem variar entre zero e VTMAXROW(). O mesmo vale para colunas.

VTSAY <exp> exibe o resultado de uma expressão de qualquer tipo.

PICTURE <cSayPicture> define a máscara para a saída de exp.

Exemplo

```
nQtd :=15.45
cDesc := "Teste descricao"
@ 1, 1 VTSAY nQtd PICTURE "@9999.99"
@ 2, 1 VTSAY "Teste VTG100"
@ 4, 1 VTSAY cDesc PICTURE "@!"
```

VTClear Screen

Tipo: TELNET VT100

Apaga a tela e coloca o cursor na posição inicial

Sintaxe

VTCLEAR [SCREEN]

@... VTClear TO

Tipo: TELNET VT100

Apaga a tela somente nas coordenadas informadas.

Sintaxe

@ <top>, <left> VTCLEAR TO <bottom>, <right>

Parâmetros

<top>	-	Linha inicial;
<left>	-	Coluna Inicial;
<bottom>	-	Linha final;
<right>	-	Coluna final;

- 1 -

Funções para VT100

@...VTGet

Tipo: TELNET VT100

Cria um novo objeto VTGET e o coloca em exibição na tela

Sintaxe

```
@ <nLin>, <nCol> [VTSAY <exp> [PICTURE <cSayPicture>]] VTGET <idVar>
    [PICTURE <cGetPicture>] [WHEN <IPreCondicao>] [VALID <IPosCondicao>]
    [PASSWORD] [F3<tabela>]
```

Parâmetros

<nLin> e <nCol> São as coordenadas de linha e coluna para a operação. Se a clausula VTSAY está presente, especificam as coordenadas para o VTSAY, e o VTGET exibido a direita deste.

VTSAY exhibe o valor de <exp> nas coordenadas especificadas. Caso a PICTURE <cSayPicture> seja especificada.

VTGET <idVar> define o nome da variável de qualquer tipo de dados a ser editada. Ela pode ser caractere, data ou numérica.

PICTURE <cGetPicture> especifica uma mascara para exibição e as regras para edição do VTGET.

WHEN <IPreCondicao> especifica uma expressão que deve ser satisfeita antes do cursor entrar na região de edição de VTGET. Se <IPreCondicao>, avaliada como verdadeira (.T.), é permitida a entrada do cursor; de outra forma, o VTGET corrente é apagado e o cursor move-se para o próximo VTGET.

VALID <IPosCondicao> especifica uma expressão que deve ser satisfeita antes que o cursor possa deixar a região de edição do VTGET corrente. Se <IPosCondicao>, avaliado sempre que o usuário tenta deixar a região de edição do VTGET, a menos que a tecla Esc seja pressionada, é verdadeira (.T.), o controle retorna ao VTGET e o usuário não pode deixar o VTGET até que <IPosCondicao> retorne verdadeiro (.T.)

usuário aperte Esc. Um VALID <IPosCondicao> pode conter ou ser uma função definida pelo usuário, permitindo-lhe executar buscas e outros tipos de operações de validação.

PASSWORD Monta o VTGET para entrada de dados com * na tela, utilizado para SENHAS

F3 Associa este VTGET a uma tabela do SXB ou Sx5.

Descrição

Quando um comando VTREAD , especificado, um VTGET executa uma edição do conteúdo <idVar> de qualquer tipo de dado. Quando um objeto VTGET , criado, o nome e valor corrente de <idVar> são guardados no objeto VTGET. O valor de <idVar> fica armazenado no buffer chamado de buffer do VTGET. O buffer de VTGET , o que é realmente mostrado na tela editado.

- 1 -

Funções para VT100

Exemplo

```
#include 'apvt100.ch'
nNumber = 0
@ 0, 0 VTSAY "Digite um numero";
VTGET nNumber;
VALID nNumber > 0
```

VTRead

Tipo: TELNET VT100

Ativa edição em tela usando objetos GET

Sintaxe

VTREAD

Descrição

O comando READ executa um módulo de edição em tela usando todos os objetos VTGET criados e adicionados.

Dentro de um READ, o usuário pode editar o buffer de cada objeto VTGET bem como mover se de um objeto GET para outro. Antes que o usuário possa entrar com um objeto VTGET controle passa para o respectivo WHEN.

Quando o usuário pressiona uma tecla de saída de VTGET, o controle passa V~~A~~ respectivo, caso tenha sido especificada.

O exemplo abaixo define vários VTGETs e a seguir usa o comando READ:

```
#include 'apvt100.ch'
cVar1 := cVar2 := cVar3 := SPACE(10)
@ 1, 1 VTSAY "Um :" VTGET cVar1 VALID !EMPTY(cVar1)
@ 2, 1 VTSAY "Dois:" VTGET cVar2 WHEN RTRIM(cVar1) != "Um"
@ 3, 1 VTSAY "Tres:" VTGET cVar3 VALID !EMPTY(cVar3)
VTREAD
```

VTSave Screen TO

Tipo: TELNET VT100

Grava a tela corrente numa variável

- 1 -

Funções para VT100

Sintaxe

VTSAVE SCREEN TO <idVar>

Parâmetros

TO <idVar> especifica a variável que serão atribuídos os conteúdos da tela corrente.

Exemplo

```
#include 'apvt100.ch'
VTSave Screen To aTela
cVar1 := cVar2 := cVar3 := SPACE(10)
@ 1, 1 VTSAY " Um :" VTGET cVar1 VALID !EMPTY(cVar1)
@ 2, 1 VTSAY " Dois:" VTGET cVar2 WHEN RTRIM(cVar1) != "Um"
@ 3, 1 VTSAY " Tres:" VTGET cVar3 VALID !EMPTY(cVar3)
VTREAD
VTRestore Screen From aTela
```

VTSave Screen

Tipo: TELNET VT100

Grava a tela as coordenadas informadas numa variável

Sintaxe

VTSAVE SCREEN VAR <var> FROM <top>, <left> TO <bottom>, <right>

Parâmetros

<Var> especifica a variável que serão atribuídos os conteúdos da tela corrente.
 <top> Linha inicial;
 <left> Coluna inicial;
 <bottom> Linha final;
 <right> Coluna final.

VTRestore Screen FROM

Tipo: TELNET VT100

Exibe uma tela guardada

Sintaxe

```
VTRESTORE SCREEN [FROM <aTela>]
```

Parâmetros

FROM <aTela> especifica uma variável que contem o conteúdo da tela a ser exibida.

Exemplo

```
#include 'apvt100.ch'
```

- 1 -

Funções para VT100

```
VTSave Screen To aTela
cVar1 := cVar2 := cVar3 := SPACE(10)
@ 1, 1 VTSAY " Um :" VTGET cVar1 VALID !EMPTY(cVar1)
@ 2, 1 VTSAY " Dois:" VTGET cVar2 WHEN RTRIM(cVar1) != "Um"
@ 3, 1 VTSAY " Tres:" VTGET cVar3 VALID !EMPTY(cVar3)
VTREAD
VTRestore Screen From aTela
```

VTRestore Screen

Tipo: TELNET VT100

Exibe uma tela guardada com as coordenadas informadas.

Sintaxe

```
VTRESTORE SCREEN VAR <aVar> FROM <top>, <left> TO <bottom>, <right>
```

Parâmetros

<aVar> especifica uma variável que contem o conteúdo da tela a ser exibida.
 <top> Linha inicial;
 <left> Coluna inicial;

<botton> Linha final;
<right> Coluna final.

VTPause

Tipo: TELNET VT100

Suspende a execução de um programa até que seja pressionada a tecla ENTER

Sintaxe

```
#include 'apvt100.ch'  
VTPAUSE
```

@...VTPause

Tipo: TELNET VT100

Exibe dados em uma linha e coluna especificadas e para a execução de um programa até seja pressionada a tecla ENTER

Sintaxe

```
@ <nLin>, <nCol>  
[VTPAUSE <exp> [PICTURE <cSayPicture>]]
```

Parâmetros

<nLin> e <nCol> são as coordenadas de linha e coluna da saída.

Os valores de linha podem variar entre zero e VTMAXROW(). O mesmo vale para colunas.

- 1 -

Funções para VT100

VTPAUSE <exp> exibe o resultado de uma expressão de qualquer tipo.

PICTURE <cSayPicture> define a mascara para a saída de exp.

Exemplo

```
#include 'apvt100.ch'  
@ 7, 1 VTPause "Tecla ENTER p/ Continuar"
```

VTSetSize

Tipo: TELNET VT100

Seta o limite da área de trabalho.

Sintaxe

```
VTSETSIZE <nLin>, <nCol>
```


Parâmetros

<nLin> e <nCol> são as coordenadas máximas de linha e coluna.

Exemplo

```
#include 'apvt100.ch'  
VTSetSize 8,20
```

VTSet Key

Tipo: TELNET VT100

Atribui a chamada de uma rotina a uma tecla

Sintaxe

```
VTSET KEY <nCodigoTecla> TO [<idRotina>]
```

Parâmetros

<nCodigoTecla> , o valor VTINKEY() da tecla a qual se atribui a rotina.

TO <idRotina> especifica o nome da rotina que é executada quando se aperta uma tecla. <idRotina> não é especificada, a definição corrente é liberada.

Exemplo

Este exemplo demonstra como usar VTSET KEY para invocar uma rotina quando o usu aperta. A tecla 'A'.

```
#include 'apvt100.ch'  
VTSET KEY 65 TO TESTE  
CCodigo := space(6)  
@ 1, 1 VTGET cCodigo  
VTREAD  
RETURN
```

```
FUNCTION Teste()
```

- 1 -

Funções para VT100

```
@ 2,1 VTSay 'TESTE'  
RETURN NIL
```

FUNCOES PARA TELNET VT100

VTReadVar()

Tipo: TELNET VT100

Retorna o nome da variável VTGET corrente

Sintaxe

VTREADVAR() --> cNomeVar

Retorna

VTREADVAR() retorna o nome da variável associada ao objeto VTGET corrente.

VTSave()

Tipo: TELNET VT100

Grava uma região de tela para posterior exibição

Sintaxe

VTSAVE (<nTopo>, <nEsquerda>, <nBase>, <nDireita>) --> cTela

Parâmetros

<nTopo>, <nEsquerda>, <nBase>, e <nDireita> definem as coordenadas da região de tela a ser gravada. Caso <nBase> ou <nDireita> seja maior do que VTMAXROW() ou VTMAXCOL(), a tela é cortada.

Retorna

VTSAVE () retorna a região de tela especificada na forma de uma cadeia de caracteres.

Exemplo

```
#include 'apvt100.ch'
aTela := VTSave(0,0,4,10)
cVar1 := cVar2 := cVar3 := SPACE(10)
@ 1, 1 VTSAY " Um :" VTGET cVar1 VALID !EMPTY(cVar1)
@ 2, 1 VTSAY " Dois:" VTGET cVar2 WHEN RTRIM(cVar1) != "Um"
@ 3, 1 VTSAY " Tres:" VTGET cVar3 VALID !EMPTY(cVar3)
```

- 1 -

Funções para VT100

```
VTREAD
VTRestore(0,0,4,10,aTela)
```

VTRestore()

Tipo: TELNET VT100

Exibe uma região de tela gravada em uma localização especificada

Sintaxe

```
VTRESTORE(<nTopo>, <nEsquerda>,
  <nBase>, <nDireita>, <aTela>) --> NIL
```

Parâmetros

<nTopo>, <nEsquerda>, <nBase>, e <nDireita> definem as coordenadas da informação de tela contida em <aTela>. <aTela> , uma variável conteúdo o conteúdo da tela gravada.

Retorna

VTRESTORE () sempre retorna NIL.

Exemplo

```
#include 'apvt100.ch'
aTela := VTSave(0,0,4,10)
cVar1 := cVar2 := cVar3 := SPACE(10)
@ 1, 1 VTSAY " Um :" VTGET cVar1 VALID !EMPTY(cVar1)
@ 2, 1 VTSAY " Dois:" VTGET cVar2 WHEN RTRIM(cVar1) != "Um"
@ 3, 1 VTSAY " Tres:" VTGET cVar3 VALID !EMPTY(cVar3)
VTREAD
VTRestore(0,0,4,10,aTela)
```

VTScroll()

Tipo: TELNET VT100

Rola uma região de tela para cima ou para baixo

Sintaxe

```
VTSCROLL(<nTopo>, <nEsquerda>,
  <nBase>, <nDireita>, <nLinhas>) --> NIL
```

Parâmetros

<nTopo>, <nEsquerda>, <nBase>, e <nDireita> definem as coordenadas da região a rolada. Valores de linha e coluna podem variar entre 0, 0 e VTMAXROW(), VTMAXCOL().

<nLinhas> define a quantidade de linhas a serem roladas. Um valor maior do que zero para cima a quantidade especificada de linhas.

Um valor menor do que zero rola para baixo a quantidade especificada de linhas. Um v de zero apaga a rea especificada.

Retorna

VTSCROLL() sempre retorna NIL.

Funções para VT100

Exemplo

```
VTSCROLL(0,0, VTMAXROW(),VTMAXCOL,1)
```

VTLastKey()

Tipo: TELNET VT100

Retorna o valor VTINKEY() da última tecla extraída do buffer de teclado

Sintaxe

VTLASTKEY() --> nCodInkey

Retorna

VTLASTKEY() retorna um número de -39 a 386 que identifica o valor VTINKEY() última tecla extraída do buffer de teclado.

Descrição

VTLASTKEY() , uma função de tratamento de teclado que informa o valor VTINKEY() última tecla capturada do buffer de teclado pela função VTINKEY(), ou por um estado de espera como VTREAD, VTPAUSE, VTACHoice(),VTABROWSE ou VTDBBROWSE. VTLASTKEY() retorna seu valor corrente até que outra tecla seja capturada do buffer de teclado.

Exemplo

```
#include 'apvt100.ch'
aTela := VTSave(0,0,4,10)
cVar1 := cVar2 := cVar3 := SPACE(10)
@ 1, 1 VTSAY " Um ." VTGET cVar1 VALID !EMPTY(cVar1)
@ 2, 1 VTSAY " Dois:" VTGET cVar2 WHEN RTRIM(cVar1) != "Um"
@ 3, 1 VTSAY " Tres:" VTGET cVar3 VALID !EMPTY(cVar3)
VTREAD
VTRestore(0,0,4,10,aTela)
If VTLastKey() == 27
  Return .f.
Endif
```

VTSetKey()

Tipo: TELNET VT100

Atribui um bloco de ação a uma tecla

Sintaxe

VTSETKEY(<nCodInkey>, [<bAção>]) --> bAcaoCorrente

Parâmetros

<nCodInkey> , o valor VTINKEY() da tecla a ser associada ou questionada.

<bAção> especifica o bloco de código a ser automaticamente executado sempre que a tecla especificada for pressionada durante um estado de espera.

Retorna

VTSETKEY() retorna o bloco de ação correntemente associado a tecla especificada, ou caso a tecla especificada não esteja associada a um bloco.

Exemplo

```
#include 'apvt100.ch'
bKeyAnt := VTSetKey(65, {|| teste()})
CCodigo := space(6)
@ 1, 1 VTGET cCodigo
VTREAD
VTSetKey(65, bKeyAnt)

RETURN

FUNCTION Teste()
  @ 2, 1 VTSay 'TESTE'
RETURN NIL
```

VTKeyboard()

Tipo: TELNET VT100

Coloca uma cadeia de caracteres (string) no buffer de teclado

Sintaxe

VTKEYBOARD(<cCodigoTecla>)

Parâmetros

<cCodigoTecla> , o conjunto de caracteres a ser colocado no buffer de teclado.

Exemplo

```
#include 'apvt100.ch'
bKeyAnt := VTSetKey(65, {|| teste()})
CCodigo := space(6)
@ 1, 1 VTGET cCodigo
VTREAD
VTSetKey(65, bKeyAnt)

RETURN

FUNCTION Teste()
  @ 2, 1 VTSay 'TESTE'
  VTKeyboard(chr(27))
RETURN NIL
```

VTRow()

Tipo: TELNET VT100

Retorna a posição de linha do cursor na tela

Sintaxe

VTROW() --> nLinha

Funções para VT100

Retorna

VTROW() retorna a posição de linha do cursor na forma de um valor numérico inteiro. A faixa do valor de retorno varia entre zero e VTMAXROW().

Exemplo

```
@ 0,0 VTSay "Teste"  
@ VTRow()+1 ,0 VTSay "Teste2"
```

VTCOL()

Tipo: TELNET VT100

Retorna a posição de coluna do cursor na tela

Sintaxe

VTCOL() --> nCol

Retorna

VTCOL() retorna um valor numérico inteiro. A faixa do valor de retorno é de zero a VTMAXCOL().

Exemplo

```
@ 1, 1 VTSAY "Cliente: " + TRIM(Cliente)  
@ VTROW(), VTCOL() + 1 VTSAY Status
```

VTInkey()

Tipo: TELNET VT100

Extrai um caractere do buffer de teclado

Sintaxe

VTINKEY([<nSegundos>]) --> nCodInkey

Parâmetros

<nSegundos> especifica a quantidade de segundos que VTINKEY() deve esperar por uma tecla. O valor pode ser especificado em incrementos do tamanho de até um décimo de segundo. Se for especificado zero, o programa aguarda até que uma tecla seja pressionada. Caso <nSegundos> seja omitido, VTINKEY() não espera por uma tecla.

Retorna

VTINKEY() retorna um valor numérico inteiro de -39 até 386, que identifica a tecla extraída do buffer de teclado. Caso o buffer de teclado esteja vazio, VTINKEY() retorna zero.

Exemplo

```
While .t.  
  IF VtInkey(1) == 27 // correspondente a tela ESC
```

Funções para VT100

```
    exit
  Endlf
End
```

VTMaxCol()

Tipo: TELNET VT100

Determina a coluna máxima visível na tela

Sintaxe

VTMAXCOL() --> nColuna

Retorna

VTMAXCOL() retorna o número da coluna visível mais a direita para fins de exibição.

Exemplo

```
@ 1, int(VTMaxCOL()/2) VTSAY "**"
```

VTMaxRow()

Tipo: TELNET VT100

Determina a máxima linha visível na tela

Sintaxe

VTMAXROW() --> nLinha

Retorna

VTMAXROW() retorna o número da última linha visível para fins de exibição.

Descrição

VTMAXROW() , uma função de tratamento de tela que pode ser utilizada para determinar a máxima linha visível da tela. Números de linha e coluna começam com zero em Clipper.

Exemplo

A seguinte função definida pelo usuário, TamTela(), utiliza VTMAXROW() e VTMAXCOL() para retornar um vetor que contém o tamanho da tela corrente:

```
FUNCTION TamTela
  RETURN { VTMAXROW(), VTMAXCOL() }
```

VTBeep()

Tipo: TELNET VT100

Emite um beep

Sintaxe

VTBEEP([<nQtde>]) --> NIL

- 1 -

Funções para VT100

Parâmetros

<nQtde> especifica a quantidade de beep que será emitido, Caso <nQtde> seja omitido VTBEEP() emitira um beep.

Retorna

VTBEEP() retorna NIL

Exemplo

```
VTBEEP(3)
```

VTReverso()

Tipo: TELNET VT100

Ativa ou desativa o modo reverso da tela.

Sintaxe

VTREVERSO([<IRev>]) --> IReverso

Parâmetros

<IRev> Se verdadeiro ativa, falso desativa o modo de tela. Caso <IRev> seja omitido VTReverso() retorna o modo atual.

Retorna

VTREVERSO() retorna o modo atual, verdadeiro que está em reverso, falso não está em reverso.

Exemplo

```
#include 'apvt100.ch'  
  
IReverso:= VTReverso(.t.)  
@ 0,0 VTSay "Teste 1"  
VTReverso(IReverso)  
@ 1,0 VTSay "Teste 2"
```

VTClearBuffer()

Tipo: TELNET VT100

Limpa o buffer de teclado

Sintaxe

VTCLEARBUFFER() --> NIL

Retorna

VTCLEARBUFFER() retorna NIL

- 1 -

Funções para VT100

Exemplo

VTCLEARBUFFER()

VTAlert()

Tipo: TELNET VT100

Mostra uma mensagem na tela

Sintaxe

VTALERT(<cMsg>,[<cCaption>],[<ICenter>],[<nSleep>],[<nBeep>]) (nTecla

Parâmetros

<cMsg> Mensagem a ser exibida.

<cCaption> Titulo da mensagem

<ICenter> Se verdadeiro centraliza a mensagem conforme o VTsetSize.

<nSleep> especifica a quantidade de tempo em milésimo de segundos em a mensagem permanecer na tela, Caso omito, aguardara que seja digitado Enter ou ESC.

<nBeep> Número de beep's a serem emitidos.

Retorna

VTALERT () retorna o código da tecla digitada.

Exemplo

VTAlert("Produto não cadastrado","Aviso",.t.,4000)

VTYesNo()

Tipo: TELNET VT100

Mostra uma mensagem a espera de uma confirmação na tela.

Sintaxe

VTYESNO (<cMsg>,[<cCaption>],[<ICenter>]) (IConfrime

Parâmetros

<cMsg> Mensagem a ser exibida.

<cCaption> Titulo da mensagem

<ICenter> Se verdadeiro centraliza a mensagem conforme o VTsetSize.

Retorna

VTYESNO () retorna o verdadeiro caso tenha confirmado.

- 1 -

Funções para VT100

Exemplo

```
IF ! VTYesNo('Confirma a alteracao','Atencao ',.T.)
  Return .F.
ENDIF
```

VTChoice()

Tipo: TELNET VT100

Executa um menu pop-up

Sintaxe

VTCHOICE(<nTopo>, <nEsquerda>, <nBase>, <nDireita>,
 <acltensMenu>, [<alltensSelecionaveis>],
 [<cFuncaoUsuario>],[<nItemInicial>], [INaoBranco] , [<IMsg>],
 [<nLinhaJanela>],[<IScroll>]) --> nPosicao

Parâmetros

<nTopo>, <nEsquerda> e <nBase>, <nDireita> são as coordenadas do canto superior esquerdo e canto inferior direito da janela. Valores de linha podem variar entre zero e VTMAXROW(), e valores de coluna podem variar entre zero e VTMAXCOL().

<acltensMenu> é um vetor que contém as cadeias de caracteres que serão exibidas como sendo os itens de menu. Cada item de menu será mais tarde identificado através de posição numérica neste vetor.

<alltensSelecionaveis> é um vetor paralelo de valores lógicos, diretamente relacionado a <acltensMenu> que especifica os itens de menu que poderão ser selecionados. Os elementos podem ser valores lógicos ou cadeias de caracteres. Caso o elemento seja uma cadeia de caracteres, ele é avaliado como uma expressão macro que deverá retornar um tipo de valor lógico. Em ambos os casos, um valor de falso (.F.) significa que o item de menu correspondente não está disponível, e um valor de verdadeiro (.T.) significa que está disponível.

<cFuncaoUsuario> é o nome de uma função definida pelo usuário que é executada quando

uma tecla não reconhecível for pressionada. O nome da função é especificado como uma expressão caractere sem parênteses ou argumentos. Note que o comportamento VTACHOICE() é afetado pela presença deste argumento. Consulte o texto abaixo para maiores informações.

<nItemInicial> é a posição ocupada no vetor de <acltensMenu> pelo item que aparecer em destaque quando o menu for exibido pela primeira vez. Caso você especifique um item de menu que não esteja disponível, ou caso você não use argumento algum, o item que aparecer em destaque será o primeiro item selecionável do vetor.

<INaoBranco> Se for verdadeiro as opções do menu será montado conforme o tamanho da opção desconsiderando os espaços em branco à direita e esquerda. Caso seja negativo a opção do menu será montado conforme a dimensão da tela do VTACHOICE definida em <nEsquerda> e <nDireita>.

<IMsg> Conteúdo tem que ser NIL, parâmetro reservado para implementação futura.

- 1 -

Funções para VT100

<nLinhaJanela> É o número da linha da janela na qual o item de menu inicial aparecerá.

Retorna

VTACHOICE() retorna a posição numérica ocupada pelo item de menu selecionado no vetor de <acltensMenu>. Se o processo de seleção for interrompido, VTACHOICE() retorna zero.

Função de usuário: Da mesma forma que as demais funções de interface com o usuário, VTACHOICE() aceita uma função de usuário. A função de usuário é especificada quando você deseja aninhar as invocações da função VTACHOICE() para criar menus hierárquicos ou redes de teclas.

Modos de VTACHOICE()

- 0-Inativo
- 1-Tentativa de passar início da lista
- 2-Tentativa de passar final da lista
- 3-Normal
- 4-tens não selecionados

Após a função de usuário ter executado as operações apropriadas ao modo VTACHOICE() ela deve retornar um valor que solicite ao VTACHOICE() executar uma operação entre o seguinte conjunto de ações:

Valores de Retorno da Função de Controle de VTACHOICE()

- 0-Aborta seleção
- 1-Executa seleção
- 2-Continua VTACHOICE()
- 3-Vai para o próximo item cuja primeira letra for a tecla pressionada

Exemplo

```
acMenuItens := {"Um", "Dois", "-----", "Tres"}
alSelectableItens := {.T., .T., .F., .T.}
nPosition := VTACHoice(0, 0, 7, 19, acMenuItens, alSelectableItens, "TESTECTRL" )
```

```
Function testectrl(modos,nElem,nElemW)
If modos == 1
    VtAlert('Top')
Elseif Modos == 2
    VtAlert('Bottom')
Else
    If VTLastkey() == 27
        VtAlert('sair')
        VTBeep(3)
        return 0
    elseif VTLastkey() == 13
        VtAlert('ok')
        VtBeep(1)
        return 1
    Endif
Endif
Return 2
```

- 1 -

Funções para VT100

VTABrowse()

Tipo: TELNET VT100

Monta um browse com referencia a um array.

Sintaxe

```
VTABROWSE(<nTopo>, <nEsquerda>, <nBase>, <nDireita>, <aCab>, [<altens>], [<aSize>],
           [<cFuncaoUsuario>],[<nItemInicial>]) --> nPosicao
```

Parâmetros

<nTopo>, <nEsquerda> e <nBase>, <nDireita> são as coordenadas do canto superior esquerdo e canto inferior direito da janela. Valores de linha podem variar entre zero e VTMAXROW(), e valores de coluna podem variar entre zero e VTMAXCOL().

<aCab>, é um vetor que contem os títulos das colunas
 <altens>, é um vetor que contem os dados a serem mostrados
 <aSize>, é um vetor que contem o tamanho de cada coluna

<cFuncaoUsuario> é o nome de uma função definida pelo usuário que é executada quando uma tecla não reconhecível for pressionada. O nome da função é especificado como uma expressão caractere sem parênteses ou argumentos. Note que o comportamento VTABROWSE() é afetado pela presença deste argumento. Consulte o texto abaixo para

maiores informações.

<nItemInicial> é a posição ocupada no vetor de <altens > pelo item que aparecer destaque quando o menu for exibido pela primeira vez. Caso você especifique um item de menu que não esteja disponível, ou caso você não use argumento algum, o item que aparecer destaque será o primeiro item selecionável do vetor.

Retorna

VTABROWSE() retorna a posição numérica ocupada pelo item de menu selecionado no vetor de <altens>. Se o processo de seleção for interrompido, VTABROWSE() retorna zero.

Função de usuário: Utilizada da mesma forma que VTACHOICE.

Modos de VTABROWSE()

- 0-Inativo
- 1-Tentativa de passar início da lista
- 2-Tentativa de passar final da lista
- 3-Normal
- 4-Itens não selecionados

Após a função de usuário ter executado as operações apropriadas ao modo de VTABROWSE() ela deve retornar um valor que solicite ao VTABROWSE() executar uma operação entre o seguinte conjunto de ações:

Valores de Retorno da Função de Controle de VTABROWSE()

- 1 -

Funções para VT100

- 0-Aborta seleção
- 1-Executa seleção
- 2-Continua VTABROWSE()
- 3-Vai para o próximo item cuja primeira letra for a tecla pressionada

Exemplo

```
#INCLUDE 'APVT100.CH'
VTClear
acab :={"Codigo","Codigo","Descricao","UM"}
aSize := {10,4,20,10}
nPos := 12
altens :={"1010 ",10,"DESCRICAO1","UN "};;
        {"2010 ",20,"DESCRICAO2","CX "};;
        {"2020 ",30,"DESCRICAO3","CX "};;
        {"2010 ",40,"DESCRICAO4","CX "};;
        {"2020 ",50,"DESCRICAO5","CX "};;
        {"3010 ",60,"DESCRICAO6","CX "};;
        {"3020 ",70,"DESCRICAO7","CX "};;
        {"3030 ",80,"DESCRICAO7","CX "};;
```

```

{"3040 ",90,"DESCRICA07","CX "},;
{"2010 ",40,"DESCRICA04","CX "},;
{"2020 ",50,"DESCRICA05","CX "},;
{"3010 ",60,"DESCRICA06","CX "},;
{"3020 ",70,"DESCRICA07","CX "},;
{"3030 ",80,"DESCRICA07","CX "},;
{"3050 ",100,"DESCRICA07","CX "}}
npos := VTaBrowse(0,0,7,15,aCab,altens,aSize,'testectrl',npos)
.
.
.

```

```

Function testectrl(modos,nElem,nElemW)
If modos == 1
    VtAlert('Top')
Elseif Modos == 2
    VtAlert('Bottom')
Else
    If VTLastkey() == 27
        VtAlert('sair')
        VTBeep(3)
        return 0
    elseif VTLastkey() == 13
        VtAlert('ok')
        VtBeep(1)
        return 1
    Endif
Endif
Return 2

```

- 1 -

Funções para VT100

VTDBBrowse()

Tipo: TELNET VT100

Monta um browse com referencia a uma tabela

Sintaxe

VTDBBROWSE(<nTopo>, <nEsquerda>, <nBase>, <nDireita>,<cAlias>, <aCab>, [<aField
[<aSize>], [<cFuncaoUsuario>],[<cTop>],[<cBottom>]] --> nRecno

Parâmetros

<nTopo>, <nEsquerda> e <nBase>, <nDireita> são as coordenadas do canto superior esquerdo e canto inferior direito da janela. Valores de li podem variar entre zero e VTMAXROW(), e valores de coluna podem variar entre zero e VTMAXCOL().

<cAlias>, é uma string com alias da tabela

<aCab>, é um vetor que contem os títulos das colunas
 <aFields>, é um vetor que contem os campos do alias
 <aSize>, é um vetor que contem o tamanho de cada coluna

<cFuncaoUsuario> é o nome de uma função definida pelo usuário que é executada quando uma tecla não reconhecível for pressionada. O nome da função é especificado como uma expressão caractere sem parênteses ou argumentos. Note que o comportamento VTDBBROWSE () é afetado pela presença deste argumento. Consulte o texto abaixo para maiores informações.

<cTop> string com a condição de validação de top
 <cBottom> string com a condição de validação de Bottom

Retorna

VTDBBROWSE () retorna o recno() Se o processo de seleção for interrompido VTDBBROWSE () retorna zero.

Função de usuário: Utilizada da mesma forma que VTACHOICE e VTaBROWSE.

Modos de VTDBBROWSE ()

- 0-Inativo
- 1-Tentativa de passar início da lista
- 2-Tentativa de passar final da lista
- 3-Normal
- 4-Itens não selecionados

Após a função de usuário ter executado as operações apropriadas ao modo VTDBBROWSE () ela deve retornar um valor que solicite ao VTDBBROWSE () executar uma operação entre o seguinte conjunto de ações:

Valores de Retorno da Função de Controle de VTDBBROWSE ()

- 0-Aborta seleção
- 1-Executa seleção
- 2-Continua VTDBBROWSE ()
- 3-Vai para o próximo item cuja primeira letra for a tecla pressionada

Exemplo

- 1 -

Funções para VT100

VtClear

```
aFields := {"B1_COD","B1_DESC","B1_UM","B1_PICM"}
aSize := {16,20,10,15}
aHeader := {'COD','DESCRICAO ','UM','% ICM'}
```

```
sb1->(dbseek(xfilial()+'00000000000001'))
nRecno := VTDBBrowse(0,0,7,15,"SB1",aHeader,aFields,aSize,'testectrl';
                    "xfilial('SB1')+00000000000001";
                    "xfilial('SB1')+00000000000002")
```

vtclear()


```

Local nValor      := 0
Local aOpcoes
Local aItens, acab, aSize
Local aFields, aHeader

VTsetSize(2,20)

While .T.
  VTClear()

  cCodigo := Space(6)
  dData   := CtoD("")
  nValor  := 0
  cSenha  := Space(6)

  VTClearBuffer()
  While .T.
    cOpcao:= "1"
    VTClear()
    @ 0,0 VTSAY "1.RF 2.MT44 3.MT16"
    @ 1,0 VTSAY "Selecione: " VTGET cOpcao pict "9"
    VTRead

    If VTLastKey() == 27
      Exit
    EndIF

    If cOpcao == "1"
      VTsetSize(8,20)
      Exit
    ElseIf cOpcao == "2"
      VTsetSize(2,40)
      Exit
    ElseIf cOpcao == "3"
      VTsetSize(2,20)
      Exit
    Else
      Vtclear()
      @ 0,0 VtSay "Opcao invalida          "
      VTInkey(1000)
    EndIf

  EndDo

  @ 00,00 VTSay PadC("Demo de VT100", If(VTModelo()=="MT16" .or.
VTModelo()=="RF",19,39))
  @ 01,00 VTSay "Modelo:" + VTModelo()

```

- 1 -

Funções para VT100

```

// Se o tecla pressionada for <ESC>
If VTLastKey() == 27

```

```

Exit
EndIF
    VTInkey(0)
    VTClear()

    VTClearBuffer()

@ 00,00 VTSay "Codigo: "
@ 00,08 VTGet cCodigo Pict "@!" Valid ValidCod(cCodigo) When WhenCod()
    VTRead
        // Se o tecla pressionada for <ESC>
If VTLastKey() == 27
    Loop
EndIF

    @ 01,00 VTSay "Data: "
@ 01,07 VTGet dData Pict "99/99/99" Valid ValidData(dData) When
WhenData()
    VTRead
        // Se o tecla pressionada for <ESC>
If VTLastKey() == 27
    Loop
EndIF

    VTClear()
    VTClearBuffer()

@ 00,00 VTSay "Valor: "
@ 00,07 VTGet nValor Pict "@E 999.99" When WhenValor()
VTRead

    If TerEsc()
    Loop
EndIF

@ 01,00 VTSay "Senha: "
@ 01,07 VTGet cSenha PASSWORD /*Pict "*****"*/ Valid (VTAlert(cSenha,'Se
Digitada',.T.,1500),.T.) When WhenSenha()
    VTRead
        // Se o tecla pressionada for <ESC>
If VTLastKey() == 27
    Loop
EndIF

    VTClear()
    VTAlert("Entrando no Achoice....", "[~]", .T., 1500)
    VTClear()

VTClearBuffer()
aOpcoes := { "Opcao 1", ;
             "Opcao 2", ;
             "Opcao 3", ;
             "Opcao 4", ;
             "Opcao 5"}

```

```

npos      :=
VTaChoice(0,0,VTMaxRow(),VTMaxCol(),aOpcoes,, 'U_VldAchVT',npos)

VTAlert("Entrando no aBrowser... aguarde...","[-]",.T.,1500)
aItens :={"1010 ",10, "DESCRICAO1", "UN "},;
        {"2010 ",20, "DESCRICAO2", "CX "},;
        {"2020 ",30, "DESCRICAO3", "CX "},;
        {"2010 ",40, "DESCRICAO4", "CX "},;
        {"2020 ",50, "DESCRICAO5", "CX "},;
        {"3010 ",60, "DESCRICAO6", "CX "},;
        {"3020 ",70, "DESCRICAO7", "CX "},;
        {"3030 ",80, "DESCRICAO7", "CX "},;
        {"3040 ",90, "DESCRICAO7", "CX "},;
        {"2010 ",100, "DESCRICAO4", "CX "},;
        {"2010 ",110, "DESCRICAO4", "CX "},;
        {"2020 ",120, "DESCRICAO5", "CX "},;
        {"3010 ",130, "DESCRICAO6", "CX "},;
        {"3020 ",140, "DESCRICAO7", "CX "},;
        {"3030 ",150, "DESCRICAO7", "CX "},;
        {"3050 ",160, "DESCRICAO7", "CX "}

acab :={"Codigo", "Cod          ", "Descricao
", "UM"}
aSize := {10,4,20,10}
nPos := 1
npos := VTaBrowse(,,,aCab,aItens,aSize,,nPos)

VTAlert("Entrando no DBBrowser... aguarde...","[-]",.T.,2000)
aFields := {"B1_COD", "B1_DESC", "B1_UM", "B1_PICM"}
aSize := {16,20,10,15}
aHeader := {'COD', 'DESCRICAO      ', 'UM', "% ICM"}
sb1->(dbseek(xfilial()))
npos := VTDBBrowse(,,, "SB1", aHeader, aFields, aSize)

VTClearBuffer()
If VTYesNo("Deseja finalizar?","Pergunta")
Exit
EndIf

If VTLastKey() == 27
Loop
EndIF

VTClearBuffer()

EndDo

Return .T.

Static Function ValidCod(cCodigo)
Local aTela
VTSAVE SCREEN TO aTela
VTAlert("Total de bytes:"+ AllTrim(Str(Len(AllTrim(cCodigo))))), ;
        "Validando:",.T., 2000)

```

Funções para VT100

```

    VTClear()
        VTRestore Screen FROM aTela
Return .T.

Static Function WhenCod()
Local aTela
    VTSAVE SCREEN TO aTela
    VTAlert("Exemplo de Get"+chr(13)+chr(10)+"com Caracter", "[
]ATENCAO", .T.,2000)
    VTClear()
    VTRestore Screen FROM aTela
Return .T.

Static Function WhenData()
Local aTela
    VTSAVE SCREEN TO aTela
    VTAlert("Exemplo de Get"+chr(13)+chr(10)+"com Data", "[
]ATENCAO", .T.,2000)
    VTClear()
    VTRestore Screen FROM aTela
Return .T.

Static Function WhenValor()
Local aTela
    VTSAVE SCREEN TO aTela
    VTAlert("Exemplo de Get"+chr(13)+chr(10)+"com Numerico", "[
]ATENCAO", .T.,2000)
    VTClear()
    VTRestore Screen FROM aTela
Return .T.

Static Function WhenSenha()
Local aTela
    VTSAVE SCREEN TO aTela
    VTAlert("Exemplo de Get"+chr(13)+chr(10)+"com Senha", "[
]ATENCAO", .T.,2000)
    VTClear()
    VTRestore Screen FROM aTela
Return .T.

Static Function ValidData(dData)
Local lRet := .T.
    If Empty(dData)
        lRet := .F.
        VTAlert("Data Invalida", "Atencao", .T.,2000)
    EndIf
Return lRet

User Function VldAchVT(modo, nElem, nElemW)

If modo == 1
    VTAlert('Inicio do Achoice', '-', .T.,1000)
Elseif Modo == 2

```

Funções para VT100

```
    VTAlert('Fim do Achoice', '-', .T., 1000)
Else
    If VTLastkey() == 27
        VTAlert('Saindo do Achoice', '-', .T., 1000)
        VTBeep(1)
        return 0
    elseif VTLastkey() == 13
        VTAlert('Tecla <ENTER> precionada', '-', .T., 1000)
        VTBeep(1)
        return 1
    Endif

EndIf

Return 2
```

