### @...CHECKBUTTON \/ DEFINE CHECKBUTTON

Creates a CheckButton control

**Standard Syntax (xBase Style):**

```
@ <nRow> ,<nCol>
CHECKBUTTON<ControlName>
[ OF | PARENT <ParentWindowName> ]
CAPTION <cCaption> | PICTURE <cPictureName>
[ WIDTH <nWidth>] [ HEIGHT <nHeight> ]
[ VALUE <lValue> ]
[ FONT <cFontName> SIZE <nFontSize> ]
[ BOLD ] [ ITALIC ] [ UNDERLINE ] [ STRIKEOUT ]
[ TOOLTIP <cToolTipText> ]
[ ON GOTFOCUS <OnGotFocusProcedur> | <bBlock> ]
[ ON CHANGE <OnChangeProcedure> | <bBlock> ]
[ ON LOSTFOCUS <OnLostFocusProcedure> | <bBlock> ]
[ ON ENTER <OnEnterProcedure> | <bBlock> ]
[ HELPID <nHelpId> ]
[ INVISIBLE ]
[ NOTABSTOP ]
[ NOTRANSPARENT ]
```

**Alternate Syntax:**

```
DEFINE CHECKBUTTON <Controlname>
   PARENT <ParentWindowName>
   ROW <nValue>
   COL <nValue>
   CAPTION <cValue>
   WIDTH <nValue>
   HEIGHT <nValue>
   FONTNAME <cValue>
   FONTSIZE <nValue>
   FONTBOLD <lValue>
   FONTITALIC <lValue>
   FONTUNDERLINE <lValue>
   FONTSTRIKEOUT <lValue>
   TOOLTIP <cValue>
   ONGOTFOCUS <ActionProcedure>
   ONLOSTFOCUS <ActionProcedure>
   ONCHANGE <ActionProcedure>
   ON ENTER <OnEnterProcedure>
   TABSTOP <lValue>
   HELPID <nValue>
   VISIBLE <lValue>
   TRANSPARENT <lValue>
END CHECKBUTTON
```

**Properties:**

- Value
- Enabled
- Visible
- Row
- Col
- Width
- Height
- Caption
- FontName
- FontSize
- FontBold
- FontItalic
- FontUnderline
- FontStrikeout
- ToolTip
- Picture
- Name (R)
- Parent (D)
- HelpId (D)
- TabStop (D)

D: Available at control definition only R: Read-Only

**Events:**

- OnGotFocus
- OnChange
- OnLostFocus
- OnEnter

**Methods:**

- Show
- Hide
- SetFocus
- Release

**Note:** Transparence in picture checkbuttons requires 256 or less color depth bitmaps.

### @...COMBOBOX \/ DEFINE COMBOBOX

Creates a ComboBox control

**Standard Syntax (xBase Style):**

```
@ <nRow> ,<nCol>
COMBOBOX <ControlName>
[ OF | PARENT <ParentWindowName> ]
[ ITEMS <caItems> ]
[ ITEMSOURCE <ItemSourceField1> [ , <ItemSourceField2> ] ]
[ VALUE <nValue> ]
[ VALUESOURCE <ValueSourceField> ]
[ DISPLAYEDIT ]
[ WIDTH <nWodth> ]
[ HEIGHT <nHeight>]
[ FONT <cFontName> SIZE <nFontSize> ]
[ BOLD ] [ ITALIC ] [ UNDERLINE ] [ STRIKEOUT ]
[ TOOLTIP <cToolTipText> ]
[ ON GOTFOCUS <OnGotFocusProcedur> | <bBlock> ]
[ ON CHANGE <OnChangeProcedure> | <bBlock> ]
[ ON LOSTFOCUS <OnLostFocusProcedure> | <bBlock> ]
[ ON ENTER <OnEnterProcedure> | <bBlock> ]
[ ON DISPLAYCHANGE <OnDisplayChangeProcedure> | <bBlock> ]
[ ON DROPDOWN <OnDropDownProcedure> | <bBlock> ]
[ ON CLOSEUP <OnCloseUpProcedure> | <bBlock> ]
[ NOTABSTOP ]
[ HELPID <nHelpId> ]
[ BREAK ]
[ GRIPPERTEXT <cGripperText> ]
[ INVISIBLE ]
[ SORT ]
[ IMAGE <acImageNames> ]
[ DROPPEDWIDTH <nDroppedWidth> ]
[ ON CANCEL <OnCancelProcedure> | <bBlock> ]
[ NOTRANSPARENT ]
```

**Alternate Syntax:**

```
DEFINE COMBOBOX <Controlname>
   PARENT <ParentWindowName>
   ROW <nValue>
   COL <nValue>
   WIDTH <nValue>
   HEIGHT <nValue>
   FONTNAME <cValue>
   FONTSIZE <nValue>
   FONTBOLD <lValue>
   FONTITALIC <lValue>
   FONTUNDERLINE <lValue>
   FONTSTRIKEOUT <lValue>
   TOOLTIP <cValue>
   ONGOTFOCUS <ActionProcedure>
   ONLOSTFOCUS <ActionProcedure>
   ONCHANGE <ActionProcedure>
   TABSTOP <lValue>
   HELPID <nValue>
   VISIBLE <lValue>
   ITEMS <caItems>
   ITEMSOURCE <ItemSourceField1> [ , <ItemSourceField2> ]
   VALUE <nValue>
   VALUESOURCE <ValueSourceField>
   DISPLAYEDIT <lValue>
   BREAK <lvalue>
   GRIPPERTEXT <cGripperText>
   SORT <lValue>
   IMAGE <acImageNames>
   DROPPEDWIDTH <nDroppedWidth>
   ONCANCEL <OnCancelProcedure>
   TRANSPARENT <lValue>
END COMBOBOX
```

**Properties:**

- Value
- Enabled
- Visible
- Item ( nItemIndex )
- Items (D)
- ItemCount
- Row
- Col
- Width
- Height
- FontName
- FontSize

- FontBold
- FontItalic
- FontUnderline
- FontStrikeout
- ToolTip
- DisplayValue
- Name (R)
- ItemSource (D)
- Parent (D)
- HelpId (D)
- Sort (D)
- TabStop (D)
- DisplayEdit (D)
- Break (D)
- GripperText (D)
- Image (D)
- DroppedWidth (D)
- ValueSource (D)

D: Available at control definition only R: Read-Only

### Events:

- OnGotFocus
- OnChange
- OnLostFocus
- OnDisplayChange
- OnEnter
- OnDropDown
- OnCloseUp
- OnCancel

### Methods:

- Show
- Hide
- AddItem ( cItemText )
- DeleteItem ( nItemIndex )
- DeleteAllItems
- SetFocus
- Release

**Hints:**

- In a ComboBox the 'Height' clause refers to the total height (considering extended list height)
- When used in control definition, ITEM property must be a character array.
- When ITEMSOURCE property is set to a fieldname, 'Value' property uses the physical record number, as in browse.
- If you set the VALUESOURCE property to a fieldname, its content is returned instead the physical record number.
- 'DroppedWidth' property is used to set the dropdown list in a combobox control. 'DroppedWidth' can't be less that ComboBox width.
- OnDropDown Event will be executed when the user attempt to open dropdown list

**Image Support:**

- 'Image' Property specify a character array containing image file names or resource names.
- When you add an item, must specify the image array index number (zero based) and the text associated with it.
- When adding items at startup you must to use a two dimensional array.
- This array must have one row for each combo item and two columns.
- The first column must contain the image index and the second the text for the item.
- When using the additem or Item properties you must use a single array containing two elements. The first, the image index item and the second, the text for the item.
- When you retrieve the item, using the 'item' property, it will return a two elements array containing the image index and the text of the item.
- When 'Image' and 'ItemSource' properties are used simultaneously, 'ItemSource' must be specified as a list containing two field names.
- The first, the image index for the items, the second, the item text.
- 'Sort' and 'Image' can't be used simultaneously.

### DEFINE CONTEXT MENU

Creates a Context Menu definition

**Standard Syntax (xBase Style):**

```
DEFINE CONTEXT MENU OF <ParentWindowName>
 MENUITEM <cItemCaption>
 ACTION <ActionProvedureName> | <bBlock>
 [ NAME <MenuItemName>]
 [ IMAGE <cImageName> ]
 [ CHECKED ]
 [ NOTRANSPARENT ]
 [ TOOLTIP <cToolTipText> ]

...
...
[ SEPARATOR ]
...
...
END MENU
```

**Alternate Syntax:**

```
DEFINE CONTEXTMENU PARENT <ParentWindowName>
   MENUITEM <cItemCaption>
   ONCLICK <ActionProvedure>
   [ NAME <MenuItemName>]
   [ IMAGE <cImageName> ]
   [ CHECKED <lValue> ]
   [ TRANSPARENT <lValue> ]
   [ TOOLTIP <cToolTipText> ]
 ...
 ...
   [ SEPARATOR ]
 ...
 ...
 END MENU
```

**ContextMenu Properties:**

- Parent (R)

R: Read-Only

**MenuItem Properties:**

- Name (R)
- Checked

- Enabled
- Image
- ToolTip

R: Read-Only

### MenuItem Events:

- OnClick

You can **DEFINE\RELEASE** Menu at runtime:

```
RELEASE CONTEXT MENU OF FormName
RELEASE CONTEXTMENU OF FormName
ReleaseContextMenu ( cFormName )
IsContextMenuDefined ( cFormName ) --> lBoolean
```

### DEFINE CONTROL CONTEXT MENU

Add a Context Menu to a Control

**Standard Syntax (xBase Style):**

```
DEFINE CONTROL CONTEXT MENU <ControlName> [ OF <ParentWindowName> ]
MENUITEM <cItemCaption>
ACTION <ActionProvedureName> | <bBlock>
[ NAME <MenuItemName>]
[ IMAGE <cImageName> ]
[ CHECKED ]
[ NOTRANSPARENT ]
[ TOOLTIP <cToolTipText> ]
 ...
 ...
[ SEPARATOR ]
 ...
 ...
END MENU
```

**Alternate Syntax:**

```
DEFINE CONTROL CONTEXTMENU <ControlName> [ PARENT <ParentWindowName> ]
   MENUITEM <cItemCaption>
   ONCLICK <ActionProvedure>
   [ NAME <MenuItemName>]
   [ IMAGE <cImageName> ]
   [ CHECKED <lValue> ]
   [ TRANSPARENT <lValue> ]
   [ TOOLTIP <cToolTipText> ]
 ...
 ...
   [ SEPARATOR ]
 ...
 ...
END MENU
```

**Control ContextMenu Properties:**

- Parent (R)

  R: Read-Only

**MenuItem Properties:**

- Name (R)
- Checked
- Enabled

- Image
- ToolTip

R: Read-Only

### MenuItem Events:

- OnClick

You can **DEFINE\/RELEASE** Menu at runtime:

```
RELEASE CONTROL CONTEXT MENU cControlName OF | PARENT cParentName
RELEASE CONTROL CONTEXTMENU cControlName OF | PARENT cParentName
ReleaseControlContextMenu ( cControlName, cParentForm )
IsControlContextMenuDefined ( cControlName, cParentForm ) --> Return lBoolean
```

### Set On\/Off Control Context Menu:

```
SET CONTROL CONTEXTMENU [ ON | OFF ]
SET CONTROL CONTEXT MENU [ ON | OFF ]
```

## @...DATEPICKER \/ DEFINE DATEPICKER

Creates a DatePicker control

### Standard Syntax (xBase Style):

```
@ <nRow> ,<nCol>
DATEPICKER<ControlName>
[ OF | PARENT <cParentWindowName> ]
[ VALUE <dValue> ]
[ FIELD <FieldName> ]
[ WIDTH <nWidth> ]
[ HEIGHT <nHeight> ]
[ FONT <cFontName> SIZE <nFontSize> ]
[ BOLD ] [ ITALIC ] [ UNDERLINE ] [ STRIKEOUT ]
[ TOOLTIP <cToolTipText> ]
[ SHOWNONE ]
[ UPDOWN ]
[ RIGHTALIGN ]
[ ON GOTFOCUS <OnGotFocusProcedur> | <bBlock> ]
[ ON CHANGE <OnChangeProcedure> | <bBlock> ]
[ ON LOSTFOCUS <OnLostFocusProcedure> | <bBlock> ]
[ HELPID <nHelpId> ]
[ ON ENTER <OnEnterProcedure> | <bBlock> ]
[ INVISIBLE ]
[ NOTABSTOP ]
[ FORMAT <cDateFormat> ]
```

### Alternate Syntax:

```
DEFINE DATEPICKER <ControlName>
   PARENT <ParentWindowName>
   ROW <nValue>
   COL <nValue>
   WIDTH <nValue>
   HEIGHT <nValue>
   FONTNAME <cValue>
   FONTSIZE <nValue>
   FONTBOLD <lValue>
   FONTITALIC <lValue>
   FONTUNDERLINE <lValue>
   FONTSTRIKEOUT <lValue>
   TOOLTIP <cValue>
   ONGOTFOCUS <ActionProcedure>
   ONLOSTFOCUS <ActionProcedure>
   ONCHANGE <ActionProcedure>
   TABSTOP <lValue>
   HELPID <nValue>
   VISIBLE <lValue>
   VALUE <dValue>
   SHOWNONE <lValue>
   UPDOWN <lValue>
   RIGHTALIGN <lValue>
   ONENTER <ActionProcedure>
   FORMAT <cDateFormat>
END DATEPICKER
```

**Properties:**

- Value
- Enabled
- Visible
- Row
- Col
- Width
- Height
- FontName
- FontSize
- FontBold
- FontItalic
- FontUnderline
- FontStrikeout
- ToolTip
- Name (R)
- Field (D)
- Parent (D)

- ShowNone (D)
- UpDown (D)
- RightAlign (D)
- HelpId (D)
- TabStop (D)

D: Available at control definition only R: Read-Only

### Events:

- OnGotFocus
- OnChange
- OnLostFocus

### Methods:

- Show
- Hide
- SetFocus
- Release
- Save
- Refresh

Note: description of the **cDateFormat** string

```
"d" The one- or two-digit day.
"dd" The two-digit day. Single-digit day values are preceded by a zero.
"ddd" The three-character weekday abbreviation.
"dddd" The full weekday name.
"M" The one- or two-digit month number.
"MM" The two-digit month number. Single-digit values are preceded by a zero.
"MMM" The three-character month abbreviation.
"MMMM" The full month name.
"yy" The last two digits of the year (that is, 1996 would be displayed as "96").
"yyyy" The full year (that is, 1996 would be displayed as "1996").
```

**Example**:

```
"d.MM.yyyy" will display date as 9.02.2012
"dd.MM.yyyy" will display date as 09.02.2012
"dd/MM/yyyy" will display date as 16/02/2012
"dd.MMM.yyyy" will display date as 16. FEB. 2012
```

- To make the information more readable, you can add body text to the format string by enclosing it in single quotes.
- Spaces and punctuation marks do not need to be quoted. Nonformat characters that are

not delimited by single quotes will result in unpredictable display by the DATEPICKER control.

- For example, to display the current date with the format "'Today is: 04:22:31 Tuesday Mar 23, 1996", the format string is "'Today is: 'hh':'m':'s dddd MMM dd', 'yyyy".
- To include a single quote in your body text, use two consecutive single quotes.
- For example, "'Don''t forget' MMM dd',' yyyy" produces output that looks like:
- Do not forget **Mar 23, 1996**. It is not necessary to use quotes with the comma, so "'Don''t forget' MMM dd, yyyy" is also valid, and produces the same output.