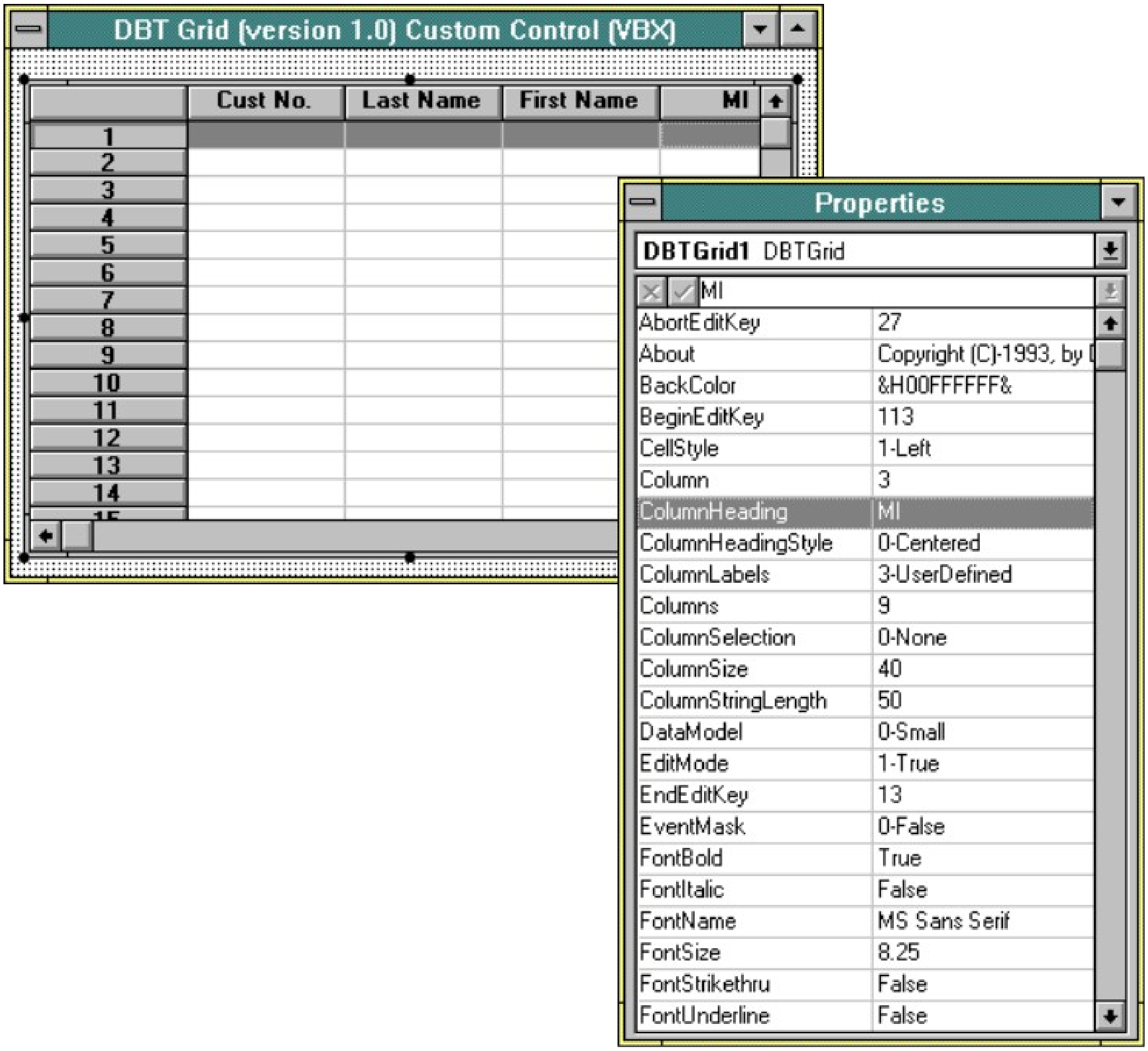
DBT Grid (version 1.0) Custom Control (VBX) Reference Guide



DBTGrid1 DBTGrid

MI

AbortEditKey	27
About	Copyright (C)-1993, by
BackColor	&H00FFFFFF&
BeginEditKey	113
CellStyle	1-Left
Column	3
ColumnHeading	MI
ColumnHeadingStyle	0-Centered
ColumnLabels	3-UserDefined
Columns	9
ColumnSelection	0-None
ColumnSize	40
ColumnStringLength	50
DataModel	0-Small
EditMode	1-True
EndEditKey	13
EventMask	0-False
FontBold	True
FontItalic	False
FontName	MS Sans Serif
FontSize	8.25
FontStrikethru	False
FontUnderline	False

Table of Contents

Introduction

<u>What is the DBT Grid?</u>

The DBT Grid (version 1.0) Custom Control (VBX) is an editable grid custom control for Microsoft Visual Basic (all versions) and Microsoft Visual C++ developers. The DBT Grid 1.0 provides the developer with a programmable object which is capable of presenting information in a tabular format (Rows and Columns). As of version 1.0 the DBT Grid has 76 properties, 19 Actions, and 19 events. This Reference Guide describes the DBT Grid's properties, Actions, and events. It is the official programmer's reference guide for DBT Grid based development.

Douglas A. Bebber
February 28, 1994

Trademarks

Visual Basic, Visual C++, and Windows are registered trademarks of the Microsoft Corporation.

The DBT Grid was written in the C programming language by Douglas A. Bebber. Address inquiries and bug reports (preferably Dr. Watson along with a listing of the suspected code) to:

Douglas A. Bebber

U.S. Postal Address:
2420 Briar Oak Circle
Sarasota, FL 34232

CompuServe ID: 72123,3661

Internet:

72123.3661@COMPUSERVE.COM

Technical Support

Registered developers may obtain free technical support by calling : (813) 378-3760 Monday - Friday, 8:00 A.M. - 6:00 P.M. EST. Also via EMail on CompuServe, ID: 72123,3661. Please have your serial number handy when calling for technical support.

24-hour BBS

Trader's Lounge BBS - DB Technologies Subboard at (813) 739-2952. At the main menu type "B DBT" to access the DB Technologies Subboard. The BBS provides developers with current Working Models, Tech Updates, news, and product support files.

How to Register

You may obtain a registered version of the DBT Grid Custom Control for $39.95 plus $5.00 S&H. Send check or money order to:

Douglas A. Bebber

2420 Briar Oak Circle
Sarasota, FL 34232
(813) 378-3760

(Please make all notes payable to: **Douglas A. Bebber).**

Credit Card Orders

You may now register the DBT Grid Custom Control from CompuServe. Type GO SWREG and select the Register menu option. (Search by title **DBTGRID** or by Author ID **72123,3661**.

DBT Grid

Description

The DBT Grid custom control provides an editable GRID control for your applications.

File Name

DBTGRID.VBX

Object Type

DBTGrid

Remarks

This control is useful for producing table objects or grid views of information. The DBT Grid is editable. A separate product is available for data-aware capabilities.

Note:

To use the DBT Grid you should copy the DBTGRID.VBX file into your WINDOWS/SYSTEM directory. The

DBTGRID.VBX file supports design time development only if the appropriate DBTGRID.LIC file can be located! Registered developers may not distribute the DBTGRID.LIC file with their applications. The DBTGRID.LIC is for registered developers only! Each DBTGRID.VBX file is shipped with a unique DBTGRID.LIC file which contains the encoded serial number of the VBX. A DBTGRID.VBX file will only support design time development if it locates a DBTGRID.LIC file which contains the correctly encoded serial number of the VBX file. During run-time the VBX file does not search for the DBTGRID.LIC file (so your completed application programs have no need for the LIC file).

THE DBTGRID.LIC FILE IS NEEDED ONLY FOR DEVELOPMENT SUPPORT!

DO NOT DISTRIBUTE IT WITH YOUR FINISHED APPLICATIONS!

We suggest that you place the DBTGRID.LIC file in your \WINDOWS\SYSTEM directory so the VBX can locate it in design-time development.

Properties

All of the properties, events, and methods for this control are listed below. Properties and events that apply only to this control, or require special consideration when used with it, are marked with an asterisk. They are documented below.

*AbortEditKey
*About
*Action
BackColor
*BeginEditKey
*CellBackColor
*CellForeColor
*CellSelectBackColor
*CellSelectForeColor
*CellStyle
*Column
*ColumnBackColor
*ColumnForeColor
*ColumnHeading
*ColumnHeadingStyle
*ColumnLabels
*Columns
*ColumnSelectBackColor
*ColumnSelectForeColor
*ColumnSelection
*ColumnSize
*ColumnStringLength
*DataModel
*DisableColumnEdit
*EditMode
*EndEditKey
*EventMask
*FixedColumn
FontBold
FontItalic
FontName
FontSize
FontStrikethru
FontUnderline
ForeColor
*GridWindowBorder
*HideColumn
Height
*Horizontals
Index
Left
*LeftVisibleColumn
*MultiSelectColumn
*MultiSelectRow
Name
*PreviousColumn
*PreviousRow
*Reaction
*Redraw
*ResizeColumn
*Row
*RowBackColor
*RowForeColor
*RowLabels
*RowNumber
*Rows
*RowSelectBackColor
*RowSelected
*RowSelectForeColor
*RowSelection
*SelectBackColor
*SelectedRow
*SelectForeColor
TabIndex
TabStop
Tag
*Text
Top
*TopVisibleRow
*Verticals
Visible
*VisibleColumns
*VisibleRows
Width

Events

*AbortEdit
*BeginEdit
*CellDoubleClicked
*CellFull
*CellValueChanged
*ColumnAdded
*ColumnDeleted
*ColumnMoved
*ColumnSized
GotFocus
*GridClick
*GridHScroll
*GridPositionChanged
*GridSelectChange
*GridVScroll
*KeyStroke
LostFocus
*MemoryError
*Resize

Methods

Move
Refresh
SetFocus

Properties

AbortEditKey

Description

Provides a user-definable key which cancels editing in a grid cell.

[form.]DBTGrid.AbortEditKey = [setting%]

Remarks

This property takes the decimal value for the Virtual Key and uses it for subsequent grid editing. The default value for this property is 27 (the Esc key). Any Virtual Key may be specified in this property however, the key assigned should not be a key used by the user in normal input operations. When the AbortEditKey is pressed by the user to cancel edit mode the text entered during that edit session is discarded and the previous cell contents are restored in the cell.

See Appendix A for a complete listing of available Virtual Key codes.

Data Type

Integer

Example

```
Sub Form_EditKeySettings()

        DBTGrid1.AbortEditKey = 114  ' AbortEditKey = <F3>
        DBTGrid1.BeginEditKey = 120  ' BeginEditKey = <F9>
        DBTGrid1.EndEditKey = 13     ' Stop editing grid cell on<Enter> (default)

End Sub
```

About

Description

Provides a Copyright notification to the user of the control.

Remarks

This property is read-only and is provided only as a copyright notification of the product.

Data Type

String

Action

Description

Provides a set of defined grid operations. This property is not available at design time and can only be written to during program execution.

[form.]DBTGrid.Action = [setting%]

Remarks

There are currently 19 DBT Grid Actions in this release (Version 1.0) of the DBT Grid. The available Actions are listed below:

1 - AbortEdit
2 - BeginEdit
3 - EndEdit
4 - DeleteColumn
5 - DeleteRow
6 - GetFirstSelectedColumn
7 - GetFirstSelectedRow
8 - GetNextSelectedColumn
9 - GetNextSelectedRow
10 - InsertColumn
11 - InsertRow
12 - MoveColumn
13 - MoveRow
14 - ResetCellColor
15 - ResetColumnColor
16 - ResetRowColor
17 - SetCellColor
18 - SetColumnColor
19 - SetRowColor

Note: You can use the DBTGRID.TXT file in your Visual Basic projects in order to use symbolic constants for DBT Grid Actions.

The result of an Action is reported in the Reaction property. Upon successfull completion of an Action the Reaction property is set to zero (0), if an error was encountered, a non-zero (error code) will be placed in the Reaction property (defined error codes are presented in the DBTGRID.TXT file).

1 - AbortEdit

Description

Aborts the editing of the current grid cell.

Remarks

This Action aborts the editing of the current grid cell. The data entered during the aborted edit session is discarded and the previous contents (if any) are restored in the grid cell. This Action is meant to be issued under program control (from code) to abort grid cell editing.

See Also

BeginEdit, and EndEdit.

Example

```
Sub AbortCellEdit()

        DBTGrid1.Action = AbortEdit  ' where the AbortEdit constant = 1

End Sub
```

2 - BeginEdit

Description
Initiates editing of the current grid cell.

Remarks
This Action initiates editing of the current grid cell.   This Action is meant to be issued under program control (from code) to initiate grid cell editing.

See Also
AbortEdit, and EndEdit.

Example

```
Sub BeginCellEdit()

        DBTGrid1.Action = BeginEdit  ' where the BeginEdit constant = 2

End Sub
```

3 - EndEdit

Description

Ends editing of the current grid cell.

Remarks

This Action ends editing of the current grid cell. This Action is meant to be issued under program control (from code) to end grid cell editing.

See Also

AbortEdit, and BeginEdit.

Example

```
Sub EndCellEdit()

        DBTGrid1.Action = EndEdit  ' where the BeginEdit constant = 3

End Sub
```

4 - DeleteColumn

Description

Deletes the current grid column.

Remarks

This Action deletes the current grid column (the column number specified in the Column property). Upon performing a DeleteColumn Action the column number specified in the Column property is deleted and all data residing in that column is deleted as well

Note: This Action can not be performed if the DataModel property setting is equal to 1-Large.

See Also

InsertColumn.

Example

```
Function DeleteGridColumn(ColumnNumber As Integer) As Integer

        DBTGrid1.Column = ColumnNumber          'Select the column to be deleted.
        DBTGrid1.Action = DeleteColumn  ' where the DeleteColumn constant = 4
        DeleteGridColumn = DBTGrid1.Reaction    ' report error (if any),
                                                        ' return of 0 reports success.
End Sub
```

5 - DeleteRow

Description

Deletes the current grid row.

Remarks

This Action deletes the current grid row (the row number specified in the Row property). Upon performing a DeleteRow Action the row number specified in the Row property is deleted and all data residing in that row is deleted as well

See Also

InsertRow.

Example

```
Function DeleteGridRow(RowNumber As Long) As Integer

        DBTGrid1.Row = RowNumber                'Select the row to be deleted.
        DBTGrid1.Action = DeleteRow             ' where the DeleteRow constant = 5
        DeleteGridRow = DBTGrid1.Reaction       ' report error (if any),
                                                ' return of 0 reports success.
End Sub
```

6 - GetFirstSelectedColumn

Description

Reports the first selected column in a grid configured for multi-column selection (the property ColumnSelection = 2).

Remarks

This Action determines the first selected column (should be used only when ColumnSelection property is set to 2 - MultiColumn). The Column number of the first selected column is placed in the MultiSelectColumn property. If there is no selected Column the value placed in the MultiSelectColumn property is negative (less than zero).

See Also

GetNextSelectedColumn.

Example

```
DBTGrid1.Action = GetFirstSelectedColumn            'Get the first selected column (if any).
If (DBTGrid1.MultiSelectColumn > -1) Then
            MsgBox "First Selected Column:" + Str$(DBTGrid1.MultiSelectColumn)
Else
            MsgBox "No Columns are currently selected!"
End If
```

7 - GetFirstSelectedRow

Description

Reports the first selected row in a grid configured for multi-row selection (the property RowSelection = 2).

Remarks

This Action determines the first selected row (should be used only when RowSelection property is set to 2 - MultiRow). The Row number of the first selected row is placed in the MultiSelectRow property. If there is no selected Row the value placed in the MultiSelectRow property is negative (less than zero).

See Also

GetNextSelectedRow.

Example

```
DBTGrid1.Action = GetFirstSelectedRow     'Get the first selected row (if any).
If (DBTGrid1.MultiSelectRow > -1) Then
            MsgBox "First Selected Row:" + Str$(DBTGrid1.MultiSelectRow)
Else
            MsgBox "No grid rows are currently selected!"
End If
```

8 - GetNextSelectedColumn

Description

Reports the next selected column in a grid configured for multi-column selection (the property ColumnSelection = 2).

Remarks

This Action determines the next selected column (should be used only when ColumnSelection property is set to 2 -

MultiColumn and after performing a GetFirstSelectedColumn Action). The Column number of the next selected column is placed in the MultiSelectColumn property. If there is no selected column the value placed in the MultiSelectColumn property is negative (less than zero).

See Also

GetFirstSelectedColumn.

Example

```
DBTGrid1.Action = GetFirstSelectedColumn          'Get the first selected column (if any).
If (DBTGrid1.MultiSelectColumn > -1) Then
        MsgBox "First Selected Column:" + Str$(DBTGrid1.MultiSelectColumn)

        Do While DBTGrid1.MultiSelectColumn > -1          'Get remaining selected Columns.
                DBTGrid1.Action = GetNextSelectedColumn
                If (DBTGrid1.MultiSelectColumn > -1) Then
                        MsgBox "Next Selected Column:" + Str$(DBTGrid1.MultiSelectColumn)
                Else
                        MsgBox "There are no more selected columns!"
                End If
        Loop

Else
        MsgBox "No Columns are currently selected!"
End If
```

9 - GetNextSelectedRow

Description

Reports the next selected row in a grid configured for multi-row selection (the property RowSelection = 2).

Remarks

This Action determines the next selected row (should be used only when RowSelection property is set to 2 - MultiRow and after performing a GetFirstSelectedRow Action). The row number of the next selected row is placed in the MultiSelectRow property. If there is no selected row the value placed in the MultiSelectRow property is negative (less than zero).

See Also

GetFirstSelectedRow.

Example

```
DBTGrid1.Action = GetFirstSelectedRow     'Get the first selected row (if any).
If (DBTGrid1.MultiSelectRow > -1) Then
        MsgBox "First Selected Row:" + Str$(DBTGrid1.MultiSelectRow)

        Do While DBTGrid1.MultiSelectRow > -1    'Get remaining selected Rows.
                DBTGrid1.Action = GetNextSelectedRow
                If (DBTGrid1.MultiSelectRow > -1) Then
                        MsgBox "Next Selected Row:" + Str$(DBTGrid1.MultiSelectRow)
                Else
                        MsgBox "There are no more selected rows!"
                End If
        Loop

Else
        MsgBox "No rows are currently selected!"
End If
```

10 - InsertColumn

Description

Inserts a grid column before the current grid column.

Remarks

This Action inserts a new column into the grid. The column is inserted immediately before the current grid column (the column number specified in the Column property). The inserted column is empty and is given a default size.

Note: This Action can not be performed if the DataModel property setting is equal to 1-Large.

See Also

DeleteColumn.

Example

```
Function InsertGridColumn(ColumnNumber As Integer) As Integer

        DBTGrid1.Column = ColumnNumber          'Select the column which the new
                                                'column will immediately precede.
        DBTGrid1.Action = InsertColumn   ' where the InsertColumn constant = 10
        InsertGridColumn = DBTGrid1.Reaction    ' report error (if any),
                                                        ' return of 0 reports success.
End Sub
```

11 - InsertRow

Description

Inserts a grid row before the current grid row.

Remarks

This Action inserts a new row into the grid. The row is inserted immediately before the current grid row (the row number specified in the Row property). The inserted row is empty.

See Also

DeleteRow.

Example

```
Function InsertGridRow(RowNumber As Long) As Integer

        DBTGrid1.Row = RowNumber                'Select the row which the new
                                                'row will immediately precede.
        DBTGrid1.Action = InsertRow             ' where the InsertRow constant = 11
        InsertGridRow = DBTGrid1.Reaction       ' report error (if any),
                                                ' return of 0 reports success.
End Sub
```

12 - MoveColumn

Description

Moves an existing grid column to another position within the grid.

Remarks

This Action moves the current grid column (the column number of which is specified in the Column property) to another position within the grid. The column number of the new position is specified in the ColumnNumber property. The example below shows how to move the third column (Column = 2) to the first column position (Column = 0) (remember that the grid's rows and columns are numbered with the first being 0 [Columns: 0 - 249, Rows: 0 - xxx...]):

Note: This Action can not be performed if the DataModel property setting is equal to 1-Large.

Example

```
DBTGrid1.Column = 2             ' We will move the third column to the front column position
                                ' in the grid. It will be the first grid column (Column=0).
DBTGrid1.ColumnNumber = 0       ' Here we specify the desired position of the column after the
                                ' move.
DBTGrid1.Action = 12            ' Move the Column.
If (DBTGrid1.Reaction <> 0 ) Then
        MsgBox "We encountered an error when moving the grid column!"
Else
        MsgBox "Column was moved successfully!"
End If
```

13 - MoveRow

Description

Moves an existing grid row to another position within the grid.

Remarks

This Action moves the current grid row (the row number of which is specified in the Row property) to another position within the grid. The row number of the new position is specified in the RowNumber property. The example below shows how to move the third row (Row = 2) to the first row position (Row = 0) (remember that the grid's rows and columns are numbered with the first being 0 [Columns: 0 - 249, Rows: 0 - xxx...]):

Example

```
DBTGrid1.Row = 2                ' We will move the third row to the top row position
                                ' in the grid. It will be the first grid row (Row=0).
DBTGrid1.RowNumber = 0          ' Here we specify the desired position of the row after the
                                ' move.
DBTGrid1.Action = 13            ' Move the row.

If (DBTGrid1.Reaction <> 0 ) Then
        MsgBox "We encountered an error when moving the grid row!"
Else
        MsgBox "Row was moved successfully!"
End If
```

14 - ResetCellColor

Description

Resets the color of a grid cell.

Remarks

This Action resets the color of the grid cell specified by the Row and Column properties. The color(s) of the specified cell is reset to the color(s) of the grid's row.

See Also

SetCellColor, CellBackColor, CellForeColor, CellSelectBackColor, and CellSelectForeColor.

Example

```
...
...
Reset the color of the very first grid cell (Row = 0, Column = 0).
DBTGrid1.Column = 0
DBTGrid1.Row = 0
DBTGrid1.Action = ResetCellColor         'Reset the cell color
If (DBTGrid1.Reaction = 0) Then          ' Perform error reporting (if any):
        MsgBox "Cell color reset ok!"
Else
        MsgBox "Error reseting cell color!"
End If
...
...
```

15 - ResetColumnColor

Description

Resets the color of a grid column.

Remarks

This Action resets the color of the grid column specified by the Column property. The color(s) of the specified column is reset to the color(s) of the grid itself.

See Also

SetColumnColor, ColumnBackColor, ColumnForeColor, ColumnSelectBackColor, and ColumnSelectForeColor.

Example

```
...
...
Reset the color of the very first grid column (Column = 0).
DBTGrid1.Column = 0
DBTGrid1.Action = ResetColumnColor       'Reset the column color
If (DBTGrid1.Reaction = 0) Then          ' Perform error reporting (if any):
        MsgBox "Column color reset ok!"
Else
```

```
        MsgBox "Error reseting column color!"
    End If
    ...
```

16 - ResetRowColor

Description

Resets the color of a grid row.

Remarks

This Action resets the color of the grid row specified by the Row property. The color(s) of the specified row is reset to the color(s) of the current column.

See Also

SetRowColor, RowBackColor, RowForeColor, RowSelectBackColor, and RowSelectForeColor.

Example

```
    'Reset the color of the very first grid row (Row = 0).
    DBTGrid1.Column = 0                 ' Reset to Color(s) of Column 0.
    DBTGrid1.Row = 0
    DBTGrid1.Action = ResetRowColor             'Reset the column color
    If (DBTGrid1.Reaction = 0) Then             ' Perform error reporting (if any):
        MsgBox "Row color reset ok!"
    Else
        MsgBox "Error reseting row color!"
    End If
```

17 - SetCellColor

Description

Sets the color(s) of a grid cell.

Remarks

This Action sets the color(s) of the grid cell specified by the Row and Column properties. The color(s) of the specified cell is set to the color(s) specified in the CellBackColor, CellForeColor, CellSelectBackColor, and CellSelectForeColor properties.

See Also

ResetCellColor, CellBackColor, CellForeColor, CellSelectBackColor, and CellSelectForeColor.

Example

```
    ' Set the color(s) of the very first cell in the grid (Row = 0, Column = 0).
    DBTGrid1.Row = 0
    DBTGrid1.Column = 0
    ' Now specify the colors:
    DBTGrid1.CellBackColor = &H00FFFFFF&                    ' White
    DBTGrid1.CellForeColor =  &H00000000&           ' Black
```

```
DBTGrid1.CellSelectBackColor = &H00000000&     ' Black
DBTGrid1.CellSelectForeColor = &H00FFFFFF&     ' White
'Now set the cell color.
DBTGrid1.Action = 17      ' 17 - SetCellColor.
```

18 - SetColumnColor

Description

Sets the color(s) of a grid column.

Remarks

This Action sets the color(s) of the grid column specified by the Column property. The color(s) of the specified column is set to the color(s) specified in the ColumnBackColor, ColumnForeColor, ColumnSelectBackColor, and ColumnSelectForeColor properties.

See Also

ResetColumnColor, ColumnBackColor, ColumnForeColor, ColumnSelectBackColor, and ColumnSelectForeColor.

Example

```
' Set the color(s) of the very first column in the grid (Column = 0).
DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.ColumnBackColor = &H00FFFFFF&                   ' White
DBTGrid1.ColumnForeColor =  &H00000000&                  ' Black
DBTGrid1.ColumnSelectBackColor = &H00000000&             ' Black
DBTGrid1.ColumnSelectForeColor = &H00FFFFFF&             ' White
'Now set the column color.
DBTGrid1.Action = 18      ' 18 - SetColumnColor.
```

19 -SetRowColor

Description

Sets the color(s) of a grid row.

Remarks

This Action sets the color(s) of the grid row specified by the Row property. The color(s) of the specified row is set to the color(s) specified in the RowBackColor, RowForeColor, RowSelectBackColor, and RowSelectForeColor properties.

See Also

ResetRowColor, RowBackColor, RowForeColor, RowSelectBackColor, and RowSelectForeColor.

Example

```
        ' Set the color(s) of the very first row in the grid (Row = 0).
        DBTGrid1.Row = 0
        ' Now specify the colors:
        DBTGrid1.RowBackColor = &H00FFFFFF&                ' White
        DBTGrid1.RowForeColor =  &H00000000&               ' Black
        DBTGrid1.RowSelectBackColor = &H00000000&    ' Black
        DBTGrid1.RowSelectForeColor = &H00FFFFFF&    ' White
        'Now set the row color.
        DBTGrid1.Action = 19      ' 19 - SetRowColor.
```

A collection of symbolic constants are defined for the DBT Grid Actions. The symbolic constants can be found in the DBTGRID.TXT file.

See Also

Reaction.

Data Type

Integer (Enumerated)

BeginEditKey

Description

Provides a user-definable key which initiates editing in a grid cell.

Usage

[form.]DBTGrid.BeginEditKey = [setting%]

Remarks

This property takes the decimal value for the Virtual Key and uses it for subsequent grid editing. The default value for this property is 113 (the F2 key). Any Virtual Key may be specified in this property however, the key assigned should not be a key used by the user in normal input operations.

See Appendix A for a complete listing of available Virtual Key codes.

Data Type

Integer

Example

```
Sub Form_EditKeySettings()

        DBTGrid1.BeginEditKey = 120  ' BeginEditKey = <F9>
        DBTGrid1.AbortEditKey = 114  ' AbortEditKey = <F3>
        DBTGrid1.EndEditKey = 13      ' Stop editing grid cell on<Enter> (default)

End Sub
```

CellBackColor

Description

Specifies the background color for a grid cell. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.CellBackColor = [setting&]

Remarks

A cells background color is specified by this property. The property setting is referenced during a SetCellColor Action only. A cells background color is not set when the CellBackColor property is set. Cell color21
s are set only by performing a SetCellColor Action.

Data Type

Long

Example

```
        ...
        ...
        ' Set the color(s) of the very first cell in the grid (Row = 0, Column = 0).

        DBTGrid1.Row = 0
        DBTGrid1.Column = 0
        ' Now specify the colors:
        DBTGrid1.CellBackColor = &H00FFFFFF&                    ' White
        DBTGrid1.CellForeColor =  &H00000000&            ' Black
        DBTGrid1.CellSelectBackColor = &H00000000&     ' Black
        DBTGrid1.CellSelectForeColor = &H00FFFFFF&     ' White
```

```
'Now set the cell color.
DBTGrid1.Action = 17      ' 17 - SetCellColor.

...
...
```

CellForeColor

Description

Specifies the foreground color for a grid cell. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.CellForeColor = [setting&]

Remarks

A cells foreground color is specified by this property. The property setting is referenced during a SetCellColor Action only! A cells foreground color is not set when the CellForeColor property is set. Cell colors are set only by performing a SetCellColor Action.

Data Type

Long

Example

```
...
...
' Set the color(s) of the very first cell in the grid (Row = 0, Column = 0).

DBTGrid1.Row = 0
```

```
DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.CellBackColor = &H00FFFFFF&                ' White
DBTGrid1.CellForeColor =  &H00000000&          ' Black
DBTGrid1.CellSelectBackColor = &H00000000&     ' Black
DBTGrid1.CellSelectForeColor = &H00FFFFFF&     ' White

'Now set the cell color.
DBTGrid1.Action = 17      ' 17 - SetCellColor.

...
...
```

CellSelectBackColor

Description

Specifies the background color for a grid cell when it is selected. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.CellSelectBackColor = [setting&]

Remarks

A cells selected background color is specified by this property. The property setting is referenced during a SetCellColor Action only! A cells selected background color is not set when the CellSelectBackColor property is set. Cell colors are set only by performing a SetCellColor Action.

Data Type

Long

Example

```
...
...
' Set the color(s) of the very first cell in the grid (Row = 0, Column = 0).

DBTGrid1.Row = 0
DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.CellBackColor = &H00FFFFFF&                 ' White
DBTGrid1.CellForeColor =  &H00000000&        ' Black
DBTGrid1.CellSelectBackColor = &H00000000&   ' Black
DBTGrid1.CellSelectForeColor = &H00FFFFFF&   ' White

'Now set the cell color.
DBTGrid1.Action = 17     ' 17 - SetCellColor.

...
...
```

CellSelectForeColor

Description

Specifies the foreground color for a grid cell when it is selected. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.CellSelectForeColor = [setting&]

Remarks

A cells selected foreground color is specified by this property. The property setting is referenced during a SetCellColor Action only! A cells selected foreground color is not set when the CellSelectForeColor property is set. Cell colors are set only by performing a SetCellColor Action.

Data Type

Long

Example

```
...
...
' Set the color(s) of the very first cell in the grid (Row = 0, Column = 0).

DBTGrid1.Row = 0
DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.CellBackColor = &H00FFFFFF&                    ' White
DBTGrid1.CellForeColor =  &H00000000&          ' Black
DBTGrid1.CellSelectBackColor = &H00000000&     ' Black
DBTGrid1.CellSelectForeColor = &H00FFFFFF&     ' White

'Now set the cell color.
DBTGrid1.Action = 17      ' 17 - SetCellColor.

...
...
```

CellStyle

Description

Determines how text in a grid cell is to be aligned. The assignment is on the current column and all text in that column will have this alignment.

Usage

[form.]DBTGrid.CellStyle = [setting%]

Remarks

The column is the current column specified by DBTGrid.Column.

The CellStyle property settings are:

Setting	Description
0	Center

1	(Default) Left
2	Right

Data Type

Integer (Enumerated)

Example

```
Sub Form_Load()
        ' Dimension the grid
        DBTGrid1.Columns = 9    ' 9 columns wide
        DBTGrid1.Rows = 30      ' by 30 rows deep
        DBTGrid1.Column = 0     ' Set the current column
        DBTGrid1.CellStyle = 0  ' Center all text in column #0
        DBTGrid1.Column = 1     ' Move to next column.
        DBTGrid1.CellStyle = 1  ' Left justify all text in column #1
End Sub
```

Column

Description

Sets or returns the current column in the grid.

Usage

[form.]DBTGrid.Column = [setting%]

Remarks

Movement between grid columns is performed through this property. When the user moves throughout the grid this property is automatically updated to reflect the current column in the grid. Columns are numbered in sequence with the first column being 0. The maximum number of columns available in this control is 250 (0 - 249).

Data Type

Integer

Example

```
Sub Form_Load()
            DBTGrid1.Column = 0      ' Set the current column
            DBTGrid1.Row = 0         'Set the current row.
            DBTGrid1.CellStyle = 0   ' Center all text in column #0
            DBTGrid1.Column = 1      ' Move to next column.
            DBTGrid1.CellStyle = 1   ' Left justify all text in column #1
End Sub
```

ColumnBackColor

Description

Specifies the background color for a grid column. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.ColumnBackColor = [setting&]

Remarks

A columns background color is specified by this property. The property setting is referenced during a SetColumnColor Action only. A columns background color is not set when the ColumnBackColor property is set. Column colors are set only by performing a SetColumnColor Action.

Data Type

Long

Example

```
            ' Set the color(s) of the very first column in the grid (Column = 0).
            DBTGrid1.Column = 0
            ' Now specify the colors:
            DBTGrid1.ColumnBackColor = &H00FFFFFF&                ' White
            DBTGrid1.ColumnForeColor =  &H00000000&               ' Black
            DBTGrid1.ColumnSelectBackColor = &H00000000&          ' Black
            DBTGrid1.ColumnSelectForeColor = &H00FFFFFF&          ' White
            'Now set the column color.
            DBTGrid1.Action = 18      ' 18 - SetColumnColor.
```

ColumnForeColor

Description

Specifies the foreground color for a grid column. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.ColumnForeColor = [setting&]

Remarks

A columns foreground color is specified by this property. The property setting is referenced during a SetColumnColor Action only! A columns foreground color is not set when the ColumnForeColor property is set. Column colors are set only by performing a SetColumnColor Action.

Data Type

Long

Example

```
' Set the color(s) of the very first column in the grid (Column = 0).

DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.ColumnBackColor = &H00FFFFFF&                  ' White
DBTGrid1.ColumnForeColor =  &H00000000&                 ' Black
DBTGrid1.ColumnSelectBackColor = &H00000000&            ' Black
DBTGrid1.ColumnSelectForeColor = &H00FFFFFF&            ' White

'Now set the column color.
DBTGrid1.Action = 18      ' 18 - SetColumnColor.
```

ColumnHeading

Description

Sets or returns the column heading string for the grid.

Usage

[form.]DBTGrid.ColumnHeading = [stringexpression$]

Remarks

Each column in the grid can have a ColumnHeading label. This label presents a description to the user concerning the information present in a grid column. The maximum size of a ColumnHeading string is 200 characters. Note that this property is only applicable when the ColumnLabels property is set to 3 (UserDefined). See the ColumnLabels description for more information.

Data Type

String

Example

```
Sub Form_Load()
        DBTGrid1.Column = 0     ' Set the current column
        DBTGrid1.Row = 0        'Set the current row.

                                      ' Set the column heading29
.
        DBTGrid1.ColumnHeading = "Customer No."

        DBTGrid1.CellStyle = 0  ' Center all text in column #0
        DBTGrid1.Column = 1     ' Move to next column.
        DBTGrid1.CellStyle = 1  ' Left justify all text in column #1
End Sub
```

ColumnHeadingStyle

Description

Sets or returns the column heading alignment for the grid.

Usage

[form.]DBTGrid.ColumnHeadingStyle = [setting%]

Remarks

Each column in the grid can have a column label (ColumnHeading string, column letters, or column numbers), this property is used to justify the column label in the grid column. The ColumnHeadingStyle property settings are:

Setting	Description
0	(Default) Center
1	Left
2	Right

Data Type

Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Column = 0      ' Set the current column
        DBTGrid1.Row = 0         'Set the current row.

        ' Set the column heading.
        DBTGrid1.ColumnHeading = "Customer No."
        ' Now center the heading in the grid
        DBTGrid1.ColumnHeadingStyle = 0

' Do column #1 now

        DBTGrid1.Column = 1      ' Set the current column

        ' Set the column heading.
        DBTGrid1.ColumnHeading = "Last Name"
        ' Now center the heading in the grid
        DBTGrid1.ColumnHeadingStyle = 0


End Sub
```

ColumnLabels

Description

Sets or returns the column labels for the grid.

Usage

[form.]DBTGrid.ColumnLabels = [setting%]

Remarks

Each column in the grid can have a column label. Grid columns however do not need to have column labels and you can remove them by setting the ColumnLabels property to 0 (None).
To label grid columns you have three choices:

Setting	Description
0	None (no column labels)
1	Letters (columns labeled with sequential letters)
2	Numbers (columns labeled with sequential numbers)
3	(Default) UserDefined (you label with ColumnHeading strings)

Data Type

Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.ColumnLabels = 3       ' User-defined column labels.
End Sub
```

ColumnNumber

Description

This property is used during MoveColumn Actions. This property is not available at design time.

Usage

[form.]DBTGrid.ColumnNumber = [setting%]

Remarks

The ColumnNumber property is used to specify the desired position of a column during a MoveColumn Action. This property is set before performing a MoveColumn Action (the MoveColumn Action internally references the value specified in this property).

See Also

Action (MoveColumn).

Data Type

Integer

Example

```
DBTGrid1.Column = 2             ' We will move the third column to the front column position
                                ' in the grid. It will be the first grid column (Column=0).
DBTGrid1.ColumnNumber = 0       ' Here we specify the desired position of the column after the
                                ' move.
DBTGrid1.Action = 12            ' Move the Column.
If (DBTGrid1.Reaction <> 0 ) Then
        MsgBox "We encountered an error when moving the grid column!"
Else
        MsgBox "Column was moved successfully!"
End If
```

Columns

Description

Sets or returns the number of columns in the grid.

Usage

[form.]DBTGrid.Columns = [setting%]

Columns are numbered in sequence with the first column being 0. The maximum number of columns available in this control is 250 (0 - 249).

Data Type

Integer

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.ColumnLabels = 3        ' User-defined column labels.
End Sub
```

ColumnSelection

Description

Sets or returns how grid columns can be selected by the user.

Usage

[form.]DBTGrid.ColumnSelection = [setting%]

Remarks

Columns can be made selectable for the user. This property determines if columns can be selected by the user, and if so, how columns can be selected. The ColumnSelection property settings are:

Setting	**Description**
0	None (columns can not be selected)
1	Single column selection supported
2	Multi-column selection supported

Data Type

Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.ColumnLabels = 3        ' User-defined column labels.
        DBTGrid1.ColumnSelection = 0     ' Do not allow columns to be selected.
End Sub
```

ColumnSelectBackColor

Description

Specifies the background color for a grid column when it is selected. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.ColumnSelectBackColor = [setting&]

Remarks

A columns selected background color is specified by this property. The property setting is referenced during a SetColumnColor Action only! A column's selected background color is not set when the ColumnSelectBackColor property is set. Column colors are set only by performing a SetColumnColor Action.

Data Type

Long

Example

```
...
...
' Set the color(s) of the very first column in the grid (Column = 0).

DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.ColumnBackColor = &H00FFFFFF&              ' White
DBTGrid1.ColumnForeColor =  &H00000000&             ' Black
DBTGrid1.ColumnSelectBackColor = &H00000000&        ' Black
DBTGrid1.ColumnSelectForeColor = &H00FFFFFF&        ' White

'Now set the column color.
DBTGrid1.Action = 18     ' 18 - SetColumnColor.

...
...
```

ColumnSelectForeColor

Description

Specifies the foreground color for a grid column when it is selected. This property is not available at design time and can only be written to during program execution.

Usage

[form.]DBTGrid.ColumnSelectForeColor = [setting&]

Remarks

A column's selected foreground color is specified by this property. The property setting is referenced during a SetColumnColor Action only! A column's selected foreground color is not set when the ColumnSelectForeColor property is set. Column colors are set only by performing a SetColumnColor Action.

Data Type

Long

Example

```
' Set the color(s) of the very first column in the grid (Column = 0).
DBTGrid1.Column = 0
' Now specify the colors:
DBTGrid1.ColumnBackColor = &H00FFFFFF&              ' White
DBTGrid1.ColumnForeColor =  &H00000000&             ' Black
DBTGrid1.ColumnSelectBackColor = &H00000000&        ' Black
DBTGrid1.ColumnSelectForeColor = &H00FFFFFF&        ' White
'Now set the column color.
DBTGrid1.Action = 18     ' 18 - SetColumnColor.
```

ColumnSize

Description

Sets or returns the width of a grid column.

Usage

[form.]DBTGrid.ColumnSize = [setting%]

Grid columns can be dynamically resized by the user as well as through program code. The number present in the ColumnSize property is in dialog units.

Data Type

Integer

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.Column = 0
        DBTGrid1.ColumnSize = 100
End Sub
```

ColumnStringLength

Description

Sets or returns the string length (in characters) of a grid column.

Usage

[form.]DBTGrid.ColumnStringLength = [setting%]

All data placed into the grid is of type string. Information is placed into the grid through the Text property. The grid columns accept string data via the Text property and all grid columns have a maximum string length. This string length is set through the ColumnStringLength property.

For example, if you have a Column set up to accept State information and you desire the user to enter the two letter state abbreviation, you will most likely set the column's ColumnStringLength property to 2. Once set, if the user tried to enter a third character, the grid would not accept it and would fire a CellFull event to handle the condition.

All grid columns are initially created with the ColumnStringLength set to 50 characters. The valid range for the ColumnStringLength property is 1 - 1000 characters.

Data Type

Integer

Example

```
Sub Form_Load()
        DBTGrid1.Column = 0     ' Set the current column
        DBTGrid1.Row = 0        'Set the current row.

        ' Set the column heading.
        DBTGrid1.ColumnHeading = "Customer No."
        ' Now center the heading in the grid
        DBTGrid1.ColumnHeadingStyle = 0
        ' Now set the max length of string in column #0 to 10 characters
        DBTGrid1.ColumnStringLength = 10

End Sub
```

DataModel

Description

Sets or returns the DataModel for the control.

Usage

[form.]DBTGrid.DataModel = [setting%]

All data placed into the grid requires system memory. The grid control allocates internal data memory based on the value placed into DataModel property. The DataModel property settings are:

Setting	Description
0	Small (the grid holds data in a single 64K segment. When the segment is full a MemoryError event is generated.)
1	Large (the grid holds data in multiple 64K segments. The total grid data is limited only by the amount of global memory available to the system. When no further global memory can be allocated a MemoryError event is generated.)

Note:The grid data size with DataModel set to 0 is approx 64Kb, with DataModel set to 1 is approx 64Mb.

Data Type

Integer (Enumerated)

Example

```
Sub Form_Load()
        'Set DataModel to Large (limited by global memory)
        DBTGrid1.DataModel = 1
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 2000    ' Set the number of rows.
        ' 9 columns x 50 bytes (ColumnStringLength) x 2000 Rows =  approx 879Kb
End Sub
```

DisableColumnEdit

Description

Controls the editing capability of a grid column.
(Not available at design-time).

Usage

[form.]DBTGrid.DisableColumnEdit = [setting%]

Remarks

A grid column can be set to allow or disallow editing of information by the user. The DisableColumnEdit property settings are:

Setting	**Description**
0	(Default) False (editing allowed)
1	True (editing not allowed)

The Column affected is the current column specified by the **Column** property.

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.EditMode = 1   ' Allow editing
        DBTGrid1.Column = 0     'Select Column 0
        DBTGrid1.DisableColumnEdit = 1 'Disallow editing of Column 0.
End Sub
```

EditMode

Description
Sets or returns the grid edit mode.

Usage
[form.]DBTGrid.EditMode = [setting%]

Remarks
The grid can be set to allow or disallow editing of information by the user. The EditMode property settings are:

Setting	**Description**
0	False (editing not allowed)
1	(Default) True (editing allowed)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()

        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.EditMode = 1   ' Allow editing
End Sub
```

EndEditKey

Description
Provides a user-definable key which normally terminates editing in a grid cell.

[form.]DBTGrid.EndEditKey = [setting%]

Remarks
This property takes the decimal value for the Virtual Key and uses it for subsequent grid editing. The default value for this property is 13 (the Enter key). Any Virtual Key may be specified in this property however, the key assigned should not be a key used by the user in normal input operations (data user's store in grid cells - this is an Edit Control key).

See Appendix A for a complete listing of available Virtual Key codes.

Data Type
Integer

Example

```
Sub Form_EditKeySettings()

        DBTGrid1.EndEditKey = 13      ' Stop editing grid cell on<Enter> (default)
        DBTGrid1.AbortEditKey = 114  ' AbortEditKey = <F3>
        DBTGrid1.BeginEditKey = 120  ' BeginEditKey = <F9>

End Sub
```

EventMask

Description
Sets or returns the grid event mask.
(Not available at design-time).

Usage
[form.]DBTGrid.EventMask = [setting%]

Remarks
The grid supported events can be turned on and off at any point by the programmer. The EventMask property settings are:

Setting	Description
0	(Default) False (events are allowed or are on)
1	True (events are masked off)

Note: The EventMask property only has control over the GridPositionChanged event.

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()

        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.EditMode = 1   ' Allow editing
        ' Shut off DBTGrid events
        DBTGrid1.EventMask = 1
                ...
                ...
        " Turn DBTGrid event support back on
        DBTGrid1.EventMask = 0
End Sub
```

FixedColumn

Description
This property allows grid columns to be fixed columns.
(Not available at design-time).

Usage
[form.]DBTGrid.FixedColumn = [setting%]

Remarks
This property is used to mark a grid column (column affected is the current column specified in the **Column** property) as fixed. Fixed columns are displayed towards the left of the grid and do not scroll horizontally. Fixed Columns do scroll vertically. You can specify as many fixed columns as you wish. When multiple fixed columns exist they are displayed left-to-right in numeric order.

For example if a grid has 5 columns (**Columns = 5**) and columns 2 and 4 are set as fixed columns (**FixedColumns = 1**), then the display order of the grid columns would be 2,4,0,1,3 (left-to-right).

Setting	**Description**
0	(Default) False (Column is not fixed)
1	True (Column is fixed)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()

            DBTGrid1.Columns = 9                        ' Set the number of columns.
            DBTGrid1.Rows = 30                          ' Set the number of rows.
            DBTGrid1.Column = 0                         'Make Column #0 a fixed Column.
            DBTGrid1.FixedColumn = 1                    'Set as Fixed.
End Sub
```

GridWindowBorder

Description
This property allows the main grid window to be (or not to be) resized by the user at run-time.

Usage
[form.]DBTGrid.GridWindowBorder = [setting%]

Remarks
This property determines how the main grid window appears to the user at run-time. It will either allow the user to resize the main grid window or not. The GridWindowBorder property settings are:

Setting	Description
0	Non-Resize (User may not resize the main grid window at run-time)
1	(Default) Resize (User may resize the main grid window)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()

            DBTGrid1.Columns = 9                        ' Set the number of columns.
            DBTGrid1.Rows = 30                          ' Set the number of rows.
            DBTGrid1.GridWindowBorder = 1               'Allow the user to resize main grid window.
End Sub
```

HideColumn

Description
This property allows grid columns to be made visible or invisible.
(Not available at design-time.)

Usage
[form.]DBTGrid.HideColumn = [setting%]

Remarks
Any grid column can be made visible or invisible using this property. This property is used in conjunction with the

ColumnNumber property. The HideColumn property setting works on the Column specified in the **ColumnNumber** property. The HideColumn property settings are:

Setting	**Description**
0	(Default) False (Column is visible)
1	True (Column is hidden from view)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()

        DBTGrid1.Columns = 9              ' Set the number of columns.
        DBTGrid1.Rows = 30                ' Set the number of rows.
        DBTGrid1.ColumnNumber = 1         'We will hide Column # 1.
        DBTGrid1.HideColumn = 1                   'Hide it!
                ...
                ...
        DBTGrid1.ColumnNumber = 1         'Lets make Column #1 visible again.
        DBTGrid1.HideColumn = 0                   'Show it!
End Sub
```

Horizontals

Description
Sets or returns the grids horizontal line support between grid rows.

Usage
[form.]DBTGrid.Horizontals = [setting%]

Remarks
The grid can have horizontal lines drawn to separate rows. The horizontal lines which separate rows can be turned on or off through the Horizontals property. The Horizontals property settings are:

Setting	**Description**
0	False (horizontal lines are off)
1	(Default) True (horizontal lines are on)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()

        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.Horizontals = 0 ' Turn off horizontal lines between rows.

End Sub
```

LeftVisibleColumn

Description
Controls the left-most visible column in the grid. This property is not available at design time.

Usage
[form.]DBTGrid.LeftVisibleColumn = [setting%]

Remarks
This property sets or gets the left-most visible column in the grid. It is used primarily to determine what portion of the grid structure is currently visible to the user. This property is used in conjunction with the TopVisibleRow, VisibleColumns, and VisibleRows property settings to determine exactly what portions of the entire grid structure are VISIBLE to the user. Using this property you can also specify what column is the left-most visible column from the user's perspective.

Data Type
Integer

Example

```
'Find out what portions of the entire grid structure is currently visible to the user.
'Declare some variables to hold the VISIBLE information.
Dim NVisibleColumns As Integer
Dim NVisibleRows As Long
Dim LeftMostColumn As Integer
Dim TopRowVisible As Long

NVisibleColumns = DBTGrid1.VisibleColumns        'Number of ENTIRE columns visible to the user.
NVisibleRows = DBTGrid1.VisibleRows              'Number of ENTIRE rows visible to the user.
LeftMostColumn = DBTGrid1.LeftVisibleColumn      'Left-most column visible to the user.
TopRowVisible = DBTGrid1.TopVisibleRow           'Upper-most row visible to the user.

'Now we know that:
'The user has row number TopVisibleRow as the uppermost visible row in the grid.
'There are NVisibleRows vertically visible starting with TopVisibleRow  at the top.
'There are NVisibleColumns visible with LeftMostColumn being the one furthest to
'the left.

'Note that PARTIALLY visible rows and columns are not reported in these numbers!
```

MultiSelectColumn

Description
This property is used to report selected columns when performing the GetFirstSelectedColumn and the GetNextSelectedColumn Actions. This property is not available at design time and is read-only during program execution.

Usage
[form.]DBTGrid.MultiSelectColumn = [setting%]

Remarks
This property is used in conjunction with the GetFirstSelectedColumn and the GetNextSelectedColumn Actions. When those actions are successfully performed, the MultiSelectColumn property holds the column number sought via the Actions. A negative value in this property indicates an invalid column number (no columns selected).

See Also
Action (GetFirstSelectedColumn and GetNextSelectedColumn).

Data Type
Integer

Example

```
DBTGrid1.Action = GetFirstSelectedColumn          'Get the first selected column (if any).
If (DBTGrid1.MultiSelectColumn > -1) Then
        MsgBox "First Selected Column:" + Str$(DBTGrid1.MultiSelectColumn)

        Do While DBTGrid1.MultiSelectColumn > -1          'Get remaining selected Columns.
                DBTGrid1.Action = GetNextSelectedColumn
                If (DBTGrid1.MultiSelectColumn > -1) Then
                        MsgBox "Next Selected Column:" + Str$(DBTGrid1.MultiSelectColumn)
                Else
                        MsgBox "There are no more selected columns!"
                End If
        Loop

Else
        MsgBox "No Columns are currently selected!"
End If
```

MultiSelectRow

Description

This property is used to report selected rows when performing the GetFirstSelectedRow and the GetNextSelectedRow Actions. This property is not available at design time and is read-only during program execution.

Usage

[form.]DBTGrid.MultiSelectRow = [setting&]

Remarks

This property is used in conjunction with the GetFirstSelectedRow and the GetNextSelectedRow Actions. When those actions are successfully performed, the MultiSelectRow property holds the row number sought via the Actions. A negative value in this property indicates an invalid row number (no rows selected).

See Also

Action (GetFirstSelectedRow and GetNextSelectedRow).

Data Type

Long

Example

```
DBTGrid1.Action = GetFirstSelectedRow     'Get the first selected row (if any).
If (DBTGrid1.MultiSelectRow > -1) Then
        MsgBox "First Selected Row:" + Str$(DBTGrid1.MultiSelectRow)

        Do While DBTGrid1.MultiSelectRow > -1    'Get remaining selected Rows.
                DBTGrid1.Action = GetNextSelectedRow
                If (DBTGrid1.MultiSelectRow > -1) Then
                        MsgBox "Next Selected Row:" + Str$(DBTGrid1.MultiSelectRow)
                Else
                        MsgBox "There are no more selected rows!"
                End If
        Loop

Else
        MsgBox "No Rows are currently selected!"
End If
```

PreviousColumn

Description

Reports the previous grid column position after a grid cell position change.
(Not available at design-time).

Usage

[form.]DBTGrid.PreviousColumn = [setting%]

Remarks
This property is used to determine the previous location (column location) of the grid cursor after a **GridPositionChanged** event has occured. Use this property to determine what column you were in prior to receiving the **GridPositionChanged** event .

Data Type
Integer

Example
MsgBox "Previous Column:" + Str$(DBTGrid1.PreviousColumn)

PreviousRow

Description
Reports the previous grid row position after a grid cell position change.
(Not available at design-time).

Usage
[form.]DBTGrid.PreviousRow = [setting&]

Remarks
This property is used to determine the previous location (row location) of the grid cursor after a **GridPositionChanged** event has occured. Use this property to determine what row you were in prior to receiving the **GridPositionChanged** event .

Data Type
Long

Example
MsgBox "Previous Row:" + Str$(DBTGrid1.PreviousRow)

Reaction

Description
Reports the status of an Action. This property is not available at design time and is read-only during program execution.

Usage
[form.]DBTGrid.Reaction = [setting%]

Remarks

This property is used to indicate the success or failure of an Action. A zero (0) is present in this property when the previously performed Action was successfully carried out, non-zero (error code) if an error was encountered. The valid error codes are present in the DBTGRID.TXT file.

Data Type

Integer

Example

```
DBTGrid1.Action = 2        'BeginEdit

If (DBTGrid1.Reaction = 0) Then
        MsgBox "Success!"
Else
        MsgBox :Failure!"
End If
```

Redraw

(Not available at design-time).

Description

This property controls the redraw state of the grid.

Usage

[form.]DBTGrid.Reaction = [setting%]

Remarks

When Redraw is set to False (0) updates to the grid are not displayed until redraw is set to True (1). The Redraw property settings are:

Setting	Description
0	False (Redraw off)
1	(Default) True (Redraw on)

Data Type

Integer (Enumerated)

Example

```
        DBTGrid1.Redraw = 0              'Turn off grid redraw.
                ...
        DBTGrid1.Redraw = 1              'Turn redraw back on.
```

ResizeColumn

Description

Sets a grid column as resizable or non-resizable.
(Not available at design-time).

Usage
[form.]DBTGrid.ResizeColumn = [setting%]

Remarks
Grid columns can be either resized by the user at run-time (**ResizeColumn = 1**) or non-resizeable (**ResizeColumn = 0**). The affected grid column is the current column (specified in the **Column** property). The ResizeColumn settings are:

Setting	Description
0	False (Can not resize column)
1	(Default) True (column is resizeable)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Column = 0             ' Set the current column
        DBTGrid1.Row = 0                'Set the current row.
        DBTGrid1.ResizeColumn = 0       'Do not allow user to resize Column #0.
End Sub
```

Row

Description
Sets or returns the current row in the grid.

Usage
[form.]DBTGrid.Row = [setting&]

Remarks
Movement between grid rows is performed through this property. When the user moves throughout the grid this property is automatically updated to reflect the current row in the grid. Rows are numbered in sequence with the first row being 0. The maximum number of rows available in this control is limited only by memory (see the DataModel property for memory specific information.) You can theoretically have more than 2 billion rows of data with the DBT Grid custom control.

Data Type
long

Example

```
Sub Form_Load()
        DBTGrid1.Column = 0     ' Set the current column
        DBTGrid1.Row = 0        'Set the current row.
        DBTGrid1.CellStyle = 0  ' Center all text in column #0
        DBTGrid1.Row = 1        ' Move to next row.
End Sub
```

RowBackColor

Description
Specifies the background color for a grid row. This property is not available at design time and can only be written to during program execution.

Usage
[form.]DBTGrid.RowBackColor = [setting&]

Remarks
A row's background color is specified by this property. The property setting is referenced during a SetRowColor Action only. A row's background color is not set when the RowBackColor property is set. Row colors are set only by performing a SetRowColor Action.

See Also
Action (ResetRowColor and SetRowColor).

Data Type
Long

Example

```
' Set the color(s) of the very first row in the grid (Row = 0).
DBTGrid1.Row = 0
' Now specify the colors:
DBTGrid1.RowBackColor = &H00FFFFFF&                    ' White
DBTGrid1.RowForeColor =  &H00000000&                   ' Black
DBTGrid1.RowSelectBackColor = &H00000000&    ' Black
DBTGrid1.RowSelectForeColor = &H00FFFFFF&    ' White
'Now set the row color.
DBTGrid1.Action = 19      ' 19 - SetRowColor.
```

RowForeColor

Description
Specifies the foreground color for a grid row. This property is not available at design time and can only be written to during program execution.

Usage
[form.]DBTGrid.RowForeColor = [setting&]

Remarks
A row's foreground color is specified by this property. The property setting is referenced during a SetRowColor Action only! A row's foreground color is not set when the RowForeColor property is set. Row colors are set only by performing a SetRowColor Action.

See Also
Action (ResetRowColor and SetRowColor).

Data Type
Long

Example

```
        ' Set the color(s) of the very first row in the grid (Row = 0).
        DBTGrid1.Row = 0
        ' Now specify the colors:
        DBTGrid1.RowBackColor = &H00FFFFFF&                 ' White
        DBTGrid1.RowForeColor =  &H00000000&                ' Black
        DBTGrid1.RowSelectBackColor = &H00000000&    ' Black
        DBTGrid1.RowSelectForeColor = &H00FFFFFF&    ' White
        'Now set the row color.
        DBTGrid1.Action = 19      ' 19 - SetRowColor.
```

RowLabels

Description
Sets or returns the row labels for the grid.

Usage
[form.]DBTGrid.RowLabels = [setting%]

Remarks
Rows can have labels which are controlled via the RowLabels property. The RowLabels property settings are:

Setting	**Description**
0	None (no row labels)
1	RowButtons (blank row buttons on each row)
2	(Default) RowNumbers (row buttons labeled with sequential numbers).

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.ColumnLabels = 3        ' User-defined column labels.
        DBTGrid1.RowLabels = 1 ' Row buttons only
End Sub
```

RowNumber

Description
This property is used during the MoveRow Action. This property is not available at design time.

Usage
[form.]DBTGrid.RowNumber = [setting&]

Remarks
The RowNumber property is used to specify the desired position of a row during a MoveRow Action. This property is set before performing a MoveRow Action (the MoveRow Action internally references the value specified in this property).

See Also
Action (MoveRow.

Data Type
Long

Example

```
DBTGrid1.Row = 2               ' We will move the third row to the top row position
                               ' in the grid. It will be the first grid row (Row=0).
DBTGrid1.RowNumber = 0         ' Here we specify the desired position of the row after the
                               ' move.
DBTGrid1.Action = 13           ' Move the Row.

If (DBTGrid1.Reaction <> 0 ) Then
        MsgBox "We encountered an error when moving the grid row!"
Else
        MsgBox "Row was moved successfully!"
End If
```

Rows

Description
Sets or returns the number of rows in the grid.

Usage
[form.]DBTGrid.Rows = [setting&]

Remarks
Rows are numbered in sequence with the first row being 0. The maximum number of rows available in this control is limited only by memory (see the DataModel property for memory specific information.) You can theoretically have more than 2 billion rows of data with the DBT Grid custom control.

Data Type
long

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
End Sub
```

RowSelectBackColor

Description
Specifies the background color for a grid row when it is selected. This property is not available at design time and can only be written to during program execution.

Usage
[form.]DBTGrid.RowSelectBackColor = [setting&]

Remarks
A row's selected background color is specified by this property. The property setting is referenced during a SetRowColor Action only! A row's selected background color is not set when the RowSelectBackColor property is set. Row colors are set only by performing a SetRowColor Action.

Data Type
Long

Example

```
        ' Set the color(s) of the very first row in the grid (Row = 0).
        DBTGrid1.Row = 0
        ' Now specify the colors:
        DBTGrid1.RowBackColor = &H00FFFFFF&                 ' White
        DBTGrid1.RowForeColor =  &H00000000&                ' Black
        DBTGrid1.RowSelectBackColor = &H00000000&    ' Black
        DBTGrid1.RowSelectForeColor = &H00FFFFFF&    ' White
        'Now set the row color.
        DBTGrid1.Action = 19      ' 19 - SetRowColor.
```

RowSelectForeColor

Description
Specifies the foreground color for a grid row when it is selected. This property is not available at design time and can only be written to during program execution.

Usage
[form.]DBTGrid.RowSelectForeColor = [setting&]

Remarks
A row's selected foreground color is specified by this property. The property setting is referenced during a SetRowColor Action only! A rows selected foreground color is not set when the RowSelectForeColor property is set. Row colors are set only by performing a SetRowColor Action.

Data Type
Long

Example

```
        ' Set the color(s) of the very first row in the grid (Row = 0).
        DBTGrid1.Row = 0
        ' Now specify the colors:
        DBTGrid1.RowBackColor = &H00FFFFFF&                ' White
        DBTGrid1.RowForeColor =  &H00000000&               ' Black
        DBTGrid1.RowSelectBackColor = &H00000000&   ' Black
        DBTGrid1.RowSelectForeColor = &H00FFFFFF&   ' White
        'Now set the row color.
        DBTGrid1.Action = 19      ' 19 - SetRowColor.
```

RowSelection

Description
Sets or returns how grid rowss can be selected by the user.

Usage
[form.]DBTGrid.RowSelection = [setting%]

Remarks
Rows can be made selectable for the user. This property determines if rows can be selected by the user, and if so, how rows can be selected. The RowSelection property settings are:

Setting	Description
0	None (rows can not be selected)
1	Single row selection supported
2	Multi-row selection supported (not available in Working Model!)

Data Type
Integer (Enumerated)

Example

```
Sub Form_Load()
        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
                                              DBTGrid1.Colum53
nLabels = 3          ' User-defined column labels.
        DBTGrid1.RowSelection = 0          ' Do not allow columns to be selected.
End Sub
```

SelectBackColor

Description
Specifies the selected background color for the grid when has selected rows and columns.

Usage
[form.]DBTGrid.SelectBackColor = [setting&]

Remarks
Sets the selected background color for the grid when it displays selections.

See Also
SelectForeColor.

Data Type
Long

Example

```
    ...
    ...
    DBTGrid1.SelectBackColor = &H00000000&        ' Black
```

SelectForeColor

Description
Specifies the selected foreground color for a grid when it has selected rows and columns.

Usage
[form.]DBTGrid.SelectForeColor = [setting&]

Remarks
Sets the selected foreground color for the grid when it displays selections.

See Also
SelectBackColor.

Data Type
Long

Example

```
    ...
    ...
    DBTGrid1.SelectForeColor = &H00000000&        ' Black
    ...
    ...
```

SelectedColumn

Description
Specifies the currently selected grid column (applicable only when ColumnSelection=1 (Single-Column selection). This property is not available at design time and can only be read during program execution.

Usage
[form.]DBTGrid.SelectedColumn = [setting%]

Remarks
This property is only applicable when the DBTGrid is set up for Single Column Selection (ColumnSelection = 1). In this mode, the SelectedColumn property is automatically maintained to report the current selected column number. If no column is currently selected this property contains a negative value to signify that no column is currently selected.

See Also
ColumnSelection.

Data Type
Integer

Example

```
'If (DBTGrid1.SelectedColumn  > -1) Then
        MsgBox "Column number " + Str$(DBTGrid1.SelectedColumn) + " is currently selected"
Else
        MsgBox "No column is currently selected!"
End If
```

SelectedRow

Description
Specifies the currently selected grid row (applicable only when RowSelection=1 (Single-Row selection). This property is not available at design time and can only be read during program execution.

Usage
[form.]DBTGrid.SelectedRow = [setting&]

Remarks
This property is only applicable when the DBTGrid is set up for Single Row Selection (RowSelection = 1). In this mode, the SelectedRow property is automatically maintained to report the current selected row number. If no row is currently selected this property contains a negative value to signify that no row is currently selected.

See Also
RowSelection.

Data Type
Long

Example

```
'If (DBTGrid1.SelectedRow  > -1) Then
        MsgBox "Row number " + Str$(DBTGrid1.SelectedRow) + " is currently selected"
Else
        MsgBox "No row is currently selected!"
End If
```

Text

Description
Sets or returns the text of a grid cell.

Usage
[form.]DBTGrid.Text = [stringexpression$]

All data placed into the grid is of type string. Information is placed into the grid through the Text property. The grid columns accept string data via the Text property and all grid columns have a maximum string length. This string length is set through the ColumnStringLength property.

For example, if you have a Column set up to accept State information and you desire the user to enter the two letter state abbreviation, you will most likely set the column's ColumnStringLength property to 2. Once set, if the user tried to enter a third character, the grid would not accept it and would fire a CellFull event to handle the condition.

All grid columns are initially created with the ColumnStringLength set to 50 characters. The valid range for the ColumnStringLength property is 1 - 1000 characters.

Data Type
String

Example

```
Sub Form_Load()
        DBTGrid1.Column = 0     ' Set the current column
        DBTGrid1.Row = 0        'Set the current row.

        ' Set the column heading.
        DBTGrid1.ColumnHeading = "Customer No."
        ' Now center the heading in the grid
        DBTGrid1.ColumnHeadingStyle = 0
        ' Now set the max length of string in column #0 to 10 characters
        DBTGrid1.ColumnStringLength = 10
        ' Put text in the grid cell
        DBTGrid1.Text = "123456789"


End Sub
```

TopVisibleRow

Description
Controls the top-most visible row in the grid. This property is not available at design time.

Usage
[form.]DBTGrid.TopVisibleRow = [setting&]

Remarks
This property sets or gets the top-most visible row in the grid. It is used primarily to determine what portion of the grid structure is currently visible to the user. This property is used in conjunction with the LeftVisibleColumn, VisibleColumns, and VisibleRows property settings to determine exactly what portions of the entire grid structure are VISIBLE to the user. Using this property you can also specify what row is the top-most visible row from the user's perspective.

Data Type
Long

Example

```
'Find out what portions of the entire grid structure is currently visible to the user.

'Declare some variables to hold the VISIBLE information.
Dim NVisibleColumns As Integer
Dim NVisibleRows As Long
Dim LeftMostColumn As Integer
Dim TopRowVisible As Long

NVisibleColumns = DBTGrid1.VisibleColumns       'Number of ENTIRE columns visible to the user.
NVisibleRows = DBTGrid1.VisibleRows             'Number of ENTIRE rows visible to the user.
LeftMostColumn = DBTGrid1.LeftVisibleColumn     'Left-most column visible to the user.
TopRowVisible = DBTGrid1.TopVisibleRow          'Upper-most row visible to the user.

'Now we know that:
'
'The user has row number TopVisibleRow as the uppermost visible row in the grid.
'There are NVisibleRows vertically visible starting with TopVisibleRow  at the top.
'There are NVisibleColumns visible with LeftMostColumn being the one furthest to
'the left.

'Note that PARTIALLY visible rows and columns are not reported in these numbers!
```

Verticals

Description

Sets or returns the grids vertical line support between grid columns.

Usage

[form.]DBTGrid.Verticals = [setting%]

Remarks

The grid can have vertical lines drawn to separate columns. The vertical lines which separate columns can be turned on or off through the Verticals property. The Verticals property settings are:

Setting	**Description**
0	False (vertical lines are off)
1	(Default) True (vertical lines are on)

Data Type

Integer (Enumerated)

Example

```
Sub Form_Load()

        DBTGrid1.Columns = 9    ' Set the number of columns.
        DBTGrid1.Rows = 30      ' Set the number of rows.
        DBTGrid1.Horizontals = 0 ' Turn off horizontal lines between rows.
        DBTGrid1.Verticals = 0  ' Turn off vertical lines between columns.

End Sub
```

VisibleColumns

Description
Reports the number of entirely visible columns in the grid. This property is not available at design time.

Usage
[form.]DBTGrid.VisibleColumns= [setting%]

Remarks
This property gets the number of entirely visible columns in the grid. It is used primarily to determine what portion of the grid structure is currently visible to the user. This property is used in conjunction with the LeftVisibleColumn, TopVisibleRow, and VisibleRows property settings to determine exactly what portions of the entire grid structure are VISIBLE to the user. Note that partially displayed columns are not reported.

Data Type
Integer

Example

```
'Find out what portions of the entire grid structure is currently visible to the user.

'Declare some variables to hold the VISIBLE information.
Dim NVisibleColumns As Integer
Dim NVisibleRows As Long
Dim LeftMostColumn As Integer
Dim TopRowVisible As Long

NVisibleColumns = DBTGrid1.VisibleColumns       'Number of ENTIRE columns visible to the user.
NVisibleRows = DBTGrid1.VisibleRows             'Number of ENTIRE rows visible to the user.
LeftMostColumn = DBTGrid1.LeftVisibleColumn     'Left-most column visible to the user.
TopRowVisible = DBTGrid1.TopVisibleRow          'Upper-most row visible to the user.

'Now we know that:
'
'The user has row number TopVisibleRow as the uppermost visible row in the grid.
'There are NVisibleRows vertically visible starting with TopVisibleRow  at the top.
                'There are NVisibleColumns visible with LeftMostColumn being the one furthest to58

'the left.

'Note that PARTIALLY visible rows and columns are not reported in these numbers!
```

VisibleRows

Description
Reports the number of entirely visible rows in the grid. This property is not available at design time.

Usage
[form.]DBTGrid.VisibleRows = [setting&]

Remarks
This property gets the number of entirely visible rows in the grid. It is used primarily to determine what portion of the grid structure is currently visible to the user. This property is used in conjunction with the LeftVisibleColumn,

TopVisibleRow, and VisibleColumns property settings to determine exactly what portions of the entire grid structure are VISIBLE to the user. Note that partially displayed rows are not reported.

Data Type
Long

Example

```
'Find out what portions of the entire grid structure is currently visible to the user.
'Declare some variables to hold the VISIBLE information.
Dim NVisibleColumns As Integer
Dim NVisibleRows As Long
Dim LeftMostColumn As Integer
Dim TopRowVisible As Long
NVisibleColumns = DBTGrid1.VisibleColumns        'Number of ENTIRE columns visible to the user.
NVisibleRows = DBTGrid1.VisibleRows              'Number of ENTIRE rows visible to the user.
LeftMostColumn = DBTGrid1.LeftVisibleColumn      'Left-most column visible to the user.
TopRowVisible = DBTGrid1.TopVisibleRow           'Upper-most row visible to the user.
'Now we know that:
'The user has row number TopVisibleRow as the uppermost visible row in the grid.
'There are NVisibleRows vertically visible starting with TopVisibleRow  at the top.
'There are NVisibleColumns visible with LeftMostColumn being the one furthest to
'the left.

'Note that PARTIALLY visible rows and columns are not reported in these numbers!
```

Events

AbortEdit

Description
Occurs when either the user aborts an edit session or when the program performs an AbortEdit Action.

Syntax
Sub DBTGrid1_AbortEdit (Column As Integer, Row As Long)

Remarks
This event occurs every time a grid cell edit session is aborted. The cell that was being edited is identified by the Column and Row arguments.

BeginEdit

Description
Occurs when either the user initiates edit mode in a grid cell or when the program performs a BeginEdit Action.

Syntax
Sub DBTGrid1_BeginEdit (Column As Integer, Row As Long)

Remarks
This event occurs every time a grid cell enters edit mode. The cell being edited is identified by the Column and Row arguments.

CellDoubleClicked

Description
Occurs when a grid cell is double-clicked with the mouse.

Syntax
Sub DBTGrid1_CellDoubleClicked (Column As Integer, Row As Long)

Remarks
This event occurs every time a grid cell is double-clicked with a mouse. The cell double clicked is identified by the Column and Row arguments.

CellFull

Description

Occurs when a grid cell's text reaches the maximum value specified by the ColumnStringLength property or when the column needs to be resized.

Syntax

Sub DBTGrid1_CellFull (Column As Integer, Row As Long)

Remarks

This event notifies the program when the grid cell's limits have been met and appropriate action must be taken. The cell generating the event is identified by the arguments.

CellValueChanged

Description

Occurs when a grid cell's text has been changed.

Syntax

Sub DBTGrid1_CellValueChanged (Column As Integer, Row As Long)

Remarks

This event notifies the program when a grid cell's text has been changed. The cell generating the event is identified by the arguments.

ColumnAdded

Description

Occurs anytime a new column is added to the grid.

Syntax

Sub DBTGrid1_ColumnAdded (Column As Integer)

Remarks

This event occurs every time a column is added to the grid. The Column added is identified by the Column argument.

ColumnDeleted

Description

Occurs anytime a column is deleted.

Syntax

Sub DBTGrid1_ColumnDeleted (Column As Integer)

Remarks

This event occurs every time a column is deleted. The Column deleted is identified by the Column argument.

ColumnMoved

Description

Occurs anytime a column is moved within the grid.

Syntax

Sub DBTGrid1_ColumnMoved (Column As Integer)

Remarks

This event occurs every time a column is moved. The Column moved is identified by the Column argument.

ColumnSized

Description

Occurs anytime a column is sized within the grid.

Syntax

Sub DBTGrid1_ColumnSized (Column As Integer)

Remarks

This event occurs every time a column is sized. The Column sized is identified by the Column argument.

GridClick

Description

Occurs anytime the user clicks the mouse (left button) inside the grid control.

Syntax

Sub DBTGrid1_GridClick (GridClickPosition As Integer)

Remarks

This event occurs every time a user clicks the mouse (left button) inside the grid. The location of the mouse click is provided by the GridClickPosition argument as follows:

GridClickPosition	Meaning (location)
0	Elsewhere in the grid VBX client area.
1	On a column heading.
2	On a row label.
3	On a grid vertical line.
4	On a grid cell.

GridHScroll

Description

Occurs when the grid has been horizontally scrolled.

Syntax

Sub DBTGrid1_GridHScroll ()

Remarks

This event notifies the program when the grid has been horizontally scrolled.

GridPositionChanged

Description

Occurs when the current grid cell position changes.

Syntax

Sub DBTGrid1_GridPositionChanged (Column As Integer, Row As Long)

Remarks

This event is generated every time the current grid's cell position is changed. When the user navigates through the grid this event fires every time a new cell is entered. The new grid cell position is identified by the arguments (Column, and Row).

GridSelectChange

Description

Occurs everytime a user either selects or unselects a row or column.

Syntax

Sub DBTGrid1_GridSelectChange ()

Remarks

This event is generated every time the user either selects or unselects a row or column.

GridVScroll

Description

Occurs when the grid has been vertically scrolled.

Syntax

Sub DBTGrid1_GridVScroll ()

Remarks

This event notifies the program when the grid has been vertically scrolled.

Keystroke

Description
Occurs when the user types a character or presses a key in the grid.

Syntax
Sub DBTGrid1_KeyStroke (Column As Integer, Row As Long, KeyCode As Integer, KeyExtra As Long)

Remarks
This event notifies the program when a keystroke is detected in a grid cell. The row, column, and the key pressed are identified by the function arguments. The **KeyExtra** parameter is a 32-bit long defined as:

Bit	**Value**
0-15 (low-order word)	Repeat count (holding down key)
16-23 (low byte of high-order word)	Scan code (OEM-dependent value)
24	Extended key (1 if extended key)
25-26	Not used
27-28	Used internally by Windows
29	Context code (1 if Alt key held down)
30	Previous key state (1 if down, 0 if up) before event.
31	Transition state (1 if key released, 0 if key pressed).

MemoryError

Description

Occurs when the grid no longer has memory space for row data.

Syntax

Sub DBTGrid1_MemoryError ()

Remarks

This event notifies the program when the grid control no longer has memory space for row data. (See the DataModel property information for memory related limits.)

Resize

Description

Occurs everytime the DBT Grid is resized.

Syntax

Sub DBTGrid1_Resize()

Remarks

This event notifies the program when the grid control has been resized.

Appendix A - Keyboard Information

Virtual Keys

The following is a list of Virtual keys (defined for Windows) that may be assigned to the DBT Grid Edit Keys:

Virtual Key	Key Value (Decimal)
VK_LBUTTON	1
VK_RBUTTON	2
VK_CANCEL	3
VK_MBUTTON	4
VK_BACK	8
VK_TAB	9
VK_CLEAR	12
VK_RETURN	13 (Default EndEditKey)
VK_SHIFT	16
VK_CONTROL	17
VK_MENU	18
VK_PAUSE	19
VK_CAPITAL	20
VK_ESCAPE	27 (Default AbortEditKey)
VK_SPACE	32
VK_PRIOR	33
VK_NEXT	34
VK_END	35
VK_HOME	36
VK_LEFT	37
VK_UP	38
VK_RIGHT	39
VK_DOWN	40
VK_SELECT	41
VK_PRINT	42
VK_EXECUTE	43
VK_SNAPSHOT	44
VK_INSERT	45
VK_DELETE	46
VK_HELP	47
VK_NUMPAD0	96
VK_NUMPAD1	97
VK_NUMPAD2	98
VK_NUMPAD3	99
VK_NUMPAD4	100
VK_NUMPAD5	101
VK_NUMPAD6	102
VK_NUMPAD7	103
VK_NUMPAD8	104
VK_NUMPAD9	105
VK_MULTIPLY	106
VK_ADD	107
VK_SEPARATOR	108
VK_SUBTRACT	109
VK_DECIMAL	110
VK_DIVIDE	111
VK_F1	112
VK_F2	113 (Default BeginEditKey)
VK_F3	114

VK_F4	115
VK_F5	116
VK_F6	117
VK_F7	118
VK_F8	119
VK_F9	120
VK_F10	121
VK_F11	122
VK_F12	123
VK_F13	124
VK_F14	125
VK_F15	126
VK_F16	127
VK_F17	128
VK_F18	129
VK_F19	130
VK_F20	131
VK_F21	132
VK_F22	133
VK_F23	134
VK_F24	135
VK_NUMLOCK	144
VK_SCROLL	145

VK_A thru VK_Z are the same as their ASCII equivalents: 'A' thru 'Z'
VK_0 thru VK_9 are the same as their ASCII equivalents: '0' thru '0'

Keyboard Templates

The keystrokes available for the DBT Grid (Working Model) are:

Key	**Description**
Up arrow	Moves up one cell
Down arrow	Moves down one cell
Right arrow	Moves right one cell
Left arrow	Moves left one cell
PgDn	Scroll down
PgUp	Scroll up
Home	Move to first cell in row
End	Move to last cell in row
Ctrl-Up	Move to top visible row
Ctrl-Down	Move to bottom visible row
Ctrl-Right	Move to right-most visible column
Ctrl-Left	Move to left-most visible column
Ctrl-PgUp	Scroll left
Ctrl-PgDn	Scroll right
Ctrl-Home	Move to top left grid cell
Ctrl-End	Move to bottom right grid cell

F2	Edit current grid cell
Enter	End cell editing
Esc	Abort cell editing

(defined for Windows)