

(ColumnLayout)

Replaced by the **(Custom)** property. In the Visual Basic "Properties" window, clicking the (Custom) property launches the Grid Editor.

AllowAdditions

Replaced by **AllowAddNew** property. Parameters and usage remain the same.

AllowDeletions

Replaced by the **AllowDelete** property. Parameters and usage remain the same.

AllowUpdates

Replaced by the **AllowUpdate** property. Parameters and usage remain the same.

AutoSize

In Version 1.0:

AutoSize was an enumerated type that set or returned the whether the control was automatically sized to its picture or the picture was sized to the button.

Setting	Description
0	(Default) No AutoSizing takes place.
1	Adjust Picture Size to Button. This will stretch or shrink the picture to fit the size of the button. This option has no effect if the picture is an icon or if there is a Caption specified for the control.
2	Adjust Button Size to Picture. This will resize the button to exactly fit the size of the Picture. This option has no effect if there is a Caption specified for the control.

In Version 2.0:

Autosize is a boolean that determines whether the control should automatically be sized to fit the picture specified in the Picture property.

Setting	Description
True	Control is sized to the size of the picture assigned in the Picture property.
False	(Default) The control will not automatically resize to a picture.

BevelWidth

In Version 1.0:

The valid range for this property was 0 to 30.

In Version 2.0:

The valid range for this property is 0 to 10 with a default of 1.

Bookmark

Consult the documentation of your host environment for detailed information on using bookmarks.

The nature of bookmarks has changed quite significantly from version 3.0 to 4.0 of Visual Basic. Bookmarks are now stored as byte arrays, rather than as strings. DataWidgets 2.0 adheres to all the new conventions for using bookmarks.

In Data Widgets 1.0, the **Bookmark** property returned the bookmark of the currently selected record in the DataGrid, while the **BookmarkCurrent** property was used to return the current record for other controls. In DataWidgets 2.0, **Bookmark** is an object that represents a specific position in a recordset.

BorderWidth

In Version 1.0:

The valid range for this property was 0 to 30

In Version 2.0:

The valid range for this property is 0 to 10 with a default value of 3.

BtnClicked

Replaced by the **BtnClick** event. Description of use remains the same.

In Version 1.0:

The syntax was *Sub control_BtnClicked ([Index As Integer])* where *Index* uniquely identified the control if it was in a control array.

In Version 2.0:

The syntax is *Sub control_BtnClick ()* with the event only firing when the column's *Style* property is set to 1 (Edit Button) or 4 (Button) and the button is clicked.

CaptionAlignment

In Version 1.0:

Determined the location of the caption area of the control. Version 1.0 allowed a setting of 3 which indicated no caption.

In Version 2.0:

- § For the SSDBData and SSDBOptSet controls, determines how the caption will be aligned on each button
- § For the Column object, adjusts the alignment of the column caption.
- § For all others, determines how the caption will be aligned on the control.

The setting of 3 (No Caption) no longer exists. Setting **Caption** = "" accomplishes the same.

CaptionType

No longer exists. Use **CaptionAlignment**.

Case

In Version 1.0:

Applied to the Edit portion of the DataCombo.

In Version 2.0:

Applies to the **Column** object. Settings remain the same.

CellUpdated

Replaced by the **AfterColUpdate** event.

Click

Description remains the same.

In Version 1.0:

Syntax was *Sub control_Clicked ([Index As Integer,] WhereClicked As Integer)*

In Version 2.0:

Syntax is *Sub control_Click (nButtonIndex As Integer)* where *nbuttonindex* is an integer value indicated the area of the control being pointed to.

See the **Click Event** in the Data Widgets 2.0 Help file for a list of the possible values of *nButtonIndex* for each control.

CloseUp

Replaced by the **ComboCloseUp** event.

In Version 1.0:

Syntax was *Sub control_CloseUp ([Index As Integer])* where *Index* uniquely identified the control if it is in a control array.

In Version 2.0:

Syntax is *Sub control_ComboCloseUp ()*

ColAlignment

Replaced by the **Alignment** property which applies to the **Column** object. Settings remain the same.

In Version 1.0:

An example of usage would be:

```
mygrid.colalignment(2) = 1
```

In Version 2.0:

An example of usage would be:

```
mygrid.columns(2).alignment = 1
```


ColAllowEnterKey

This property does not exist in Version 2.0.

Pressing the ENTER key moves from cell to cell. Pressing CTRL-ENTER forces a line break when editing the cell.

ColAllowInput

Replaced by the **Locked** property. Usage remains the same.

Syntax

object. **Locked**[= *boolean*]

Setting	Description
True	User input into the column is allowed.
False	(Default) User input into the column is not allowed.

ColAllowNulls

This property does not exist in Version 2.0.

ColBackColor

This property does not exist in Version 2.0. The introduction of objects allows you to access the **BackColor** property directly from the column object.

In Version 1.0:

The syntax for this property was:

```
SSDataGrid1.ColBackColor(n) = color
```

In Version 2.0:

The syntax for this property is:

```
SSDBGrid1.Columns(n).BackColor = color
```

ColBtn

This property has been replaced by the **Style** property.

In Version 1.0:

The syntax was

object. **ColBtn**(n) = {True/False}

In Version 2.0:

The **Style** property allows you to set the column to a range of styles.

Setting	Description
0	(Default) Edit Mode
1	Edit Button
2	Check Box
3	Combo Box
4	Button

The following code accomplishes the same display as shown above:

```
SSDBGrid1.Columns(n).Style = setting
```

ColBtnPicture

Replaced by the **PictureButton** property.

In Version 1.0:

The syntax was

object. ColBtnPicture(n) = picturename

In Version 2.0:

The syntax is

object. PictureButton = picturename

Version 2.0 allows the picture to be set for the entire grid, not per column.

ColCase

Replaced by the **Case** property.

In Version 1.0:

The syntax was

object. **ColCase**(n) = x

In Version 2.0:

The syntax is

object. **Columns**(n) = x

ColChanged

This property does not exist in Version 2.0.

ColDropDownHwnd

This property has been replaced by **DropDownHwnd**.

In Version 1.0:

An example of usage would be:

```
SSDataGrid1.ColDropDownHwnd(0) = SSDropDown1.Hwnd
```

In Version 2.0:

An example of usage would be:

```
SSDBGrid1.Columns(0).DropDownHwnd = SSDBDropDown1.Hwnd
```

ColFieldLen

Replaced by the FieldLen property.

In Version 1.0:

An example of usage would be:

```
SSDataGrid1.ColFieldLen(0)=2
```

In Version 2.0:

An example of usage is:

```
SSDBGrid1.Columns(0).FieldLen=2
```

ColFieldName

This property does not exist in Version 2.0.

Each column has a DataField and a caption. Use the **Caption** to reference the column in the **Columns** collection.

ColFieldNum

This property does not exist in Version 2.0.

In Version 1.0:

This property was used to indicate the column's position in the grid.

In Version 2.0:

Use the column's Position property to find the column's current position in the grid.

ColFieldType

Replaced by the **DataType** property.

In Version 1.0:

An example of usage would be:

```
SSDataGrid.ColFieldType(0) = 2
```

In Version 2.0:

An example of usage would be:

```
SSDBGrid1.Columns(0).DataType = 17
```

ColForeColor

This property does not exist in Version 2.0. The introduction of objects allows you to access the **ForeColor** property directly from the column object.

In Version 1.0:

The syntax for this property was:

object.ColForeColor(n) = color

In Version 2.0:

The syntax for this property is:

object.Columns(n).ForeColor = color

ColFormat

This property has been replaced by the **NumberFormat** property.

In Version 1.0:

The syntax for this property was:

object. **ColFormat**(n) = *format*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**NumberFormat** = *format*

The NumberFormat property makes use of Visual Basic's Format function.

ColHeading

This has been replaced by the **Caption** property on the **Column** object.

In Version 1.0:

The syntax for this property was:

object. **ColHeading**(n) = *text*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**Caption** = *text*

ColHeadingAlignment

Replaced by the **CaptionAlignment** property.

In Version 1.0:

The syntax for this property was:

object. **ColHeadingAlignment** = *setting*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**CaptionAlignment** = *setting*

ColHeadingLines

Replaced by the **Headlines** property.

In Version 1.0:

The syntax for this property was:

object. **ColHeadingLines** = *value*

In Version 2.0:

The syntax for this property is:

object. **Headlines** = *value*

ColPtX

Replaced by the **Left** property of the **Column** object.

In Version 1.0:

The syntax for this property was:

object. **ColPtX**(*n*) = *x-coordinate*

In Version 2.0:

The syntax for this property is:

object. Columns(*n*).**Left** = *x-coordinate*

ColResized

Replaced by the **ColResize** event.

In Version 1.0:

Syntax was:

Sub *control_ColResized* (*[Index As Integer,*] *ResizedCol As Integer*)

In Version 2.0:

Syntax is:

Sub *control_ColResize* (*[ColIndex As Integer]* [*Cancel As Integer*])

where *ColIndex* specifies the column being resized and *Cancel* specifies whether or not the operation occurs.

ColText

Replaced by the **Text** property of the **Column** object.

In Version 1.0:

The syntax for this property was:

object. **ColText**(n) = *text*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**Text** = *text*

ColTextScrollBar

Replaced by the **VertScrollBar** property of the **Column** object.

In Version 1.0:

The syntax for this property was:

object. **ColTextScrollBar** = {True/False}

In Version 2.0:

The syntax for this property is:

object. Columns(n).**VertScrollBar** = {True/False}

Version 2.0 allows you to selectively specify whether each column will have a scrollbar.

ColUpdateable

Replaced by the **Locked** property of the **Column** object. Functionality remains the same.

In Version 1.0:

Syntax was:

object. **ColUpdateable** = {True/False}

In Version 2.0:

Syntax is:

object. Columns(n).**Locked** = {True/False}

ColVisible

Replaced by the **Visible** property of the **Column** object.

In Version 1.0:

The syntax for this property was:

object. **ColVisible**(n) = {True/False}

In Version 2.0:

The syntax for this property is:

object. Columns(n).**Visible** = {True/False}

ColWidth

Replaced by the **Width** property of the **Column** object.

In Version 1.0:

The syntax for this property was:

object.ColWidth(n) = width

In Version 2.0:

The syntax for this property is:

object.Columns(n).Width = width



Version 1.0 to 2.0 Compatability Reference

Due to the transition from VBX based controls to OCX based controls, there have been many changes in the Properties, Events, and Methods implemented in Data Widgets Version 2.0. This help file is a guide to help ease the transition with a listing of all Properties, Events, and Methods that existed in Version 1. Clicking on an item will display information on how it was implemented in Version 2.

[Properties](#) [Version 1.0](#)
[Events](#) [Properties/Methods](#)
[Metods](#)

[Properties](#) [Version 1.0 Events](#)
[Events](#)
[Metods](#)



[Data Widgets 2.0 Online](#)
[Help](#)

Data Widgets 1.0 Events

BtnClicked

Click

CellUpdated

CloseUp

ColResized

DeleteBegin

DeleteEnd

FirstRowAdd

FixedColsChanged

InitColumnProps

LeftColChange

PositionList

RowColChange

RowLoaded

RowsResized

SelChange

TopRowChange

UpdateBegin

UpdateEnd

UpdateError

ValidateList

Data Widgets 1.0 Properties/Methods

Listed below are the properties and methods that existed in Version 1.0 of Data Widgets. Clicking on each property/method will provide information on how it is implemented in Version 2.0 of Data Widgets. Items that have hypertext jumps (indicated by being underlined) have changed since Version 1. Items that do not have hypertext jumps are the same in Version 2.

(About)	DatabaseAction	PictureButtons
(ColumnLayout)	<u>DataSourceHwnd</u>	PictureCaption
<u>AllowAdditions</u>	DataSourceList	<u>PictureCaptionAlign</u>
<u>AllowDeletions</u>	<u>DataSourceListHwnd</u>	Redraw
AllowInput	<u>DefBtnPicture</u>	<u>RoundedCorners</u>
AllowNull	DefColWidth	<u>Row</u>
<u>AllowUpdates</u>	DelayInitial	<u>RowCellBackColor</u>
<u>AutoSize</u>	DelaySubsequent	<u>RowCellForeColor</u>
BevelInner	<u>DeleteSelected</u>	<u>RowCellBold</u>
BevelOuter	<u>DividerStyle</u>	<u>RowCellItalic</u>
<u>BevelWidth</u>	<u>DividerType</u>	<u>RowCellUnderline</u>
<u>Bookmark</u>	<u>DropDownPtX</u>	<u>RowCellStrikeThru</u>
<u>BorderWidth</u>	<u>DropDownPtY</u>	RowChanged
ButtonSize	DroppedDown	RowHeight
<u>CaptionAlignment</u>	<u>EvalRowBookmark</u>	<u>RowLabels</u>
<u>CaptionType</u>	<u>EvalRowNumber</u>	Rows
<u>Case</u>	<u>EvalRowIsSelected</u>	<u>SavedBookmark</u>
Col	<u>FixedCols</u>	<u>ScrollBars</u>
<u>ColAlignment</u>	Font3D	<u>SelBookmark</u>
<u>ColAllowEnterKey</u>	<u>HdgBackColor</u>	SelectByCell
<u>ColAllowInput</u>	<u>HdgCellForeColor</u>	<u>SelCount</u>
<u>ColAllowNulls</u>	<u>HdgCellBold</u>	<u>SelectionTypeCol</u>
<u>ColBackColor</u>	<u>HdgCellItalic</u>	<u>SelectionTypeRow</u>
<u>ColBtn</u>	<u>HdgCellUnderline</u>	<u>SelRow</u>
<u>ColBtnPicture</u>	<u>HdgCellStrikeThru</u>	<u>SelStartBomark</u>
<u>ColCase</u>	<u>HdgForeColor</u>	<u>SelEndBookmark</u>
<u>ColChanged</u>	<u>HdgStyle</u>	<u>SelStartCol</u>
<u>ColDropDownHwnd</u>	HwndEdit	<u>SelEndCol</u>
<u>ColFieldLen</u>	<u>IsCurrentCellValid</u>	<u>SelStartRow</u>
<u>ColFieldName</u>	IsItemInList	<u>SelEndRow</u>
<u>ColFieldNum</u>	LeftCol	ShadowColor
<u>ColFieldType</u>	ListAutoPosition	<u>ShowBookmarkButton</u>
<u>ColForeColor</u>	ListAutoValidate	<u>ShowPageButton</u>
<u>ColFormat</u>	<u>ListLinkedCol</u>	<u>ShowFirstLastButton</u>

<u>ColHeading</u>	<u>ListSequence</u>	<u>ShowPrevNextButton</u>
<u>ColHeadingAlignment</u>	<u>ListStyle</u>	<u>TextFieldType</u>
<u>ColHeadingLines</u>	ListWidth	TextFormat
<u>ColPtX</u>	ListWidthAutoSize	<u>TopRow</u>
Cols	MaxDropDownItems	UseExactRowCount
<u>ColText</u>	<u>MaxRowsEachScroll</u>	<u>VariableColWidth</u>
<u>ColTextScrollBar</u>	MinDropDownItems	<u>VariableRowHeight</u>
<u>ColUpdateable</u>	OptionValue	<u>VariableFixedCols</u>
<u>ColVisible</u>	<u>Outline</u>	VisibleRows
<u>ColWidth</u>	<u>PageValue</u>	

DataSourceHwnd

This property does not exist in Version 2.0 because Visual Basic 4 allows Data Widgets to bind across forms.

DataSourceListHwnd

This property does not exist in Version 2.0 because Visual Basic 4 allows Data Widgets to bind across forms.

DefBtnPicture

Replaced by the **PictureButton** property. Functionality remains the same.

DeleteBegin

Replaced by the **BeforeDelete** event.

In Version 1.0:

Syntax was:

```
Sub control_DeleteBegin ([Index As Integer,] RtnCancel As Integer,  
RtnDispPromptMsg As Integer)
```

In Version 2.0:

Syntax is:

```
Sub control_BeforeDelete (Cancel As Integer, DispPromptMsg As Integer)
```

where *Cancel* is an integer expression that specifies whether the operation occurs and *DispPromptMsg* is an integer expression specifying whether a dialog is displayed asking the user to confirm the deletion.

DeleteEnd

Replaced by the **AfterDelete** event.

In Version 1.0:

Syntax was:

Sub *control_DeleteEnd* (*[Index As Integer]*)

In Version 2.0:

Syntax is:

Sub *control_AfterDelete* (*RtnDispErrMsg As Integer*)

where *RtnDispErrMsg* is an integer expression that indicates if an error message box should be displayed.

DeleteSelected

This property does not exist in Version 2.0 since it is no longer needed with the addition of the **BeforeDelete** and **AfterDelete** events.

DividerStyle

In Version 1.0:

Setting	Description
0	None. No divider lines will be displayed in the control.
1	Black line.
2	Dark gray line.
3	Raised. The dividers will appear to be raised from the control, giving a 3-D look.
4	Inset. The dividers will appear to be inset into the control, giving a 3-D look.

In Version 2.0:

Settings	Description
0	Black line
1	(Default) Dark gray line
2	Raised
3	Inset
4	ForeColor

DividerType

In Version 1.0:

Setting	Description
0	None. No divider lines will be displayed in the control.
1	Vertical only. Vertical lines will be displayed between each column.
2	Horizontal only. Horizontal lines will be displayed between each row.
3	Both. Both horizontal and vertical lines will be drawn between each row and column in the control.

In Version 2.0:

Settings	Description
0	None
1	Horizontal
2	Vertical
3	(Default) Both

DropDownPtX

This property does not exist in Version 2.0.

In Version 2.0:

Use the **ActiveCell.Left** and **ActiveCell.Width** properties to return this information.

DropDownPtY

This property does not exist in Version 2.0.

In Version 2.0:

Use the **ActiveCell.Top** and **ActiveCell.Height** properties to return this information.

EvalRowBookmark

This property has been replaced by the **RowBookmark** method.

EvalRowsSelected

This property does not exist in Version 2.0.

In Version 2.0:

You can get the row from the **SelBookmark** collection.

EvalRowNumber

This property does not exist in Version 2.0.

In Version 2.0:

The use of bookmarks eliminates the use of row numbers. You can access the data in a row by getting the bookmark for that row and using the **CellText** and **CellValue** properties of the **Column** object.

FirstRowAdd

This property does not exist in Version 2.0 since Visual Basic 4.0 now supports the addition of a row to an empty record set for custom controls.

FixedCols

This property has been eliminated due to the introduction of the Splitter bar.

Refer to the **SplitterPos** and **SplitterVisible** properties for more information on how to work with fixed columns.

FixedColsChanged

Replaced by the **SplitterMove** event.

In Version 1.0:

Syntax was:

Sub *control*_FixedCols()

In Version 2.0:

Syntax is:

Sub *control*_SplitterMove ([*Cancel* **As Integer**])

where *Cancel* is an integer expression that specifies whether or not the operation occurs.

HdgBackColor

This property does not exist in Version 2.0. The introduction of objects allows you to access the **BackColor** property directly from the column object.

In Version 1.0:

The syntax for this property was:

object. **HdgBackColor** = *color*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**HeadBackColor** = *color*

HdgCellBold

This property does not exist in Version 2.0. The **HeadFont** object allows you to specify the font attributes directly from the object.

In Version 1.0:

The syntax for this property was:

object. **HdgCellBold** = {True/False}

In Version 2.0:

The syntax for this property is:

object. **HeadFont.Bold** = {True/False}

HdgCellForeColor

This property does not exist in Version 2.0. The introduction of objects allows you to access this property directly from the column object.

In Version 1.0:

The syntax for this property was:

object. **HdgCellForeColor** = *color*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**ForeColor** = *color*

HdgCellItalic

This property does not exist in Version 2.0. The **HeadFont** object allows you to specify the font attributes directly from the object.

In Version 1.0:

The syntax for this property was:

object. **HdgCellItalic** = {True/False}

In Version 2.0:

The syntax for this property is:

object. **HeadFont**.Italic = {True/False}

HdgCellStrikeThru

This property does not exist in Version 2.0. The **HeadFont** object allows you to specify the font attributes directly from the object.

In Version 1.0:

The syntax for this property was:

object. **HdgCellStrikeThru** = {True/False}

In Version 2.0:

The syntax for this property is:

object. **HeadFont**.Strikethrough = {True/False}

HdgCellUnderline

This property does not exist in Version 2.0. The **HeadFont** object allows you to specify the font attributes directly from the object.

In Version 1.0:

The syntax for this property was:

object. **HdgCellUnderline** = {True/False}

In Version 2.0:

The syntax for this property is:

object. **HeadFont**.Underline = {True/False}

HdgForeColor

This property does not exist in Version 2.0. The introduction of objects allows you to access this property directly from the **Column** object.

In Version 1.0:

The syntax for this property was:

object. **HdgForeColor** = *color*

In Version 2.0:

The syntax for this property is:

object. Columns(n).**HeadForeColor** = *color*

HdgStyle

Replaced by the **HeadFont3D** property.

In Version 1.0:

Setting	Description
0	2D - (BackColor used)
1	(Default) 3D - (BackColor ignored)

In Version 2.0:

Settings	Description
0	(Default) None. Text is displayed flat (not three-dimensional).
1	Raised with light shading. Caption appears as if it is raised slightly off the screen.
2	Raised with heavy shading. Caption appears even more raised.
3	Inset with light shading. Caption appears as if it is inset slightly into the screen.
4	Inset with heavy shading. Caption appears even more inset.

InitColumnProps

In Version 1.0:

Syntax was:

Sub *control*_InitColumnProps ([*Index As Integer*])

In Version 2.0:

Syntax is:

Sub *control*_InitColumnProps ()

IsCurrentCellValid

Replaced by the **IsCellValid** method.

In Version 1.0:

Syntax was:

object. **IsCurrentCellValid**

In Version 2.0:

Syntax is:

object. Column(n).**IsCellValid**

IsCellValid can be used in the **BeforeColUpdate** event to make sure that the text/value that was entered can be converted to the `DataType` of the column. It can also be used anywhere to validate an entire row prior to an application's call to update. This would be particularly helpful in the `LostFocus` event.

LeftColChange

This property does not exist in Version 2.0.

In Version 2.0:

This is now handled by the **Scroll** event.

ListLinkedCol

This property does not exist in Version 2.0.

In Version 2.0:

Refer to the **DatasourceList** and **DataFieldList** properties.

ListSequence

This property does not exist in Version 2.0.

In Version 2.0:

This property is no longer needed as this has been optimized by the data control.

ListStyle

This property does not exist in Version 2.0.

In Version 2.0:

Use the **RowSelectionType** property to set the selection type.

Use a combination of **DividerStyle** and **BevelColor** to achieve a 3D look.

MaxRowsEachScroll

This property does not exist in Version 2.0.

In Version 2.0:

MaxRowsEachScroll is no longer needed as all scrolling has been optimized in the data control. This property was in Version 1.0 to deal with performance issues in the VBX data control.

Outline

This property has been replaced by **BorderStyle**.

In Version 1.0:

Setting	Description
True	(Default) A thin black outline border (one pixel wide) will be drawn around the control.
False	No outline is drawn.

In Version 2.0:

Setting	Description
0	No border will be displayed.
1	(Default) A fixed single border will be displayed.

PageValue

In Version 1.0:

The maximum value for this property was 1000.

In Version 2.0:

SSDBCommand:

Valid range is 2-1000. The default value is 20.

SSDBData:

Valid range is 1-300. The default value is 20.

PictureCaptionAlign

In Version 1.0:

Setting	Description
0	(Default) Left align. The picture will be aligned to the left of the caption area.
1	Right align. The picture will be aligned to the right of the caption area.
2	Centered. The picture will be centered in the caption area.

In Version 2.0:

Setting	Description
0	Left - Top
1	Left - Middle
2	Left - Bottom
3	Right - Top
4	(Default) Right - Middle
5	Right - Bottom
6	Center - Top
7	Center - Middle
8	Center - Bottom
9	Left of Caption
10	Right of Caption
11	Above Caption
12	Below Caption
13	Fit to Caption
14	Tile

PositionList

In Version 1.0:

Syntax was:

```
Sub control_PositionList ([Index As Integer,] Text As String)
```

In Version 2.0:

Syntax is:

```
Sub control_PositionList (Text As String)
```

where *Text* is the text to match in the list of values.

RoundedCorners

In Version 1.0:

Setting	Description
----------------	--------------------

True	(Default) The control will be drawn with rounded corners.
------	---

False	The control will be drawn with square corners.
-------	--

In Version 2.0:

Setting	Description
----------------	--------------------

0	(False). Control will be displayed with square corners.
---	---

1	(True) Control will be displayed with rounded corners.
---	--

Row

In Version 1.0:

Returned or set the current row number in the list.

In Version 2.0:

Sets or returns the current display row. Display row means that only those rows currently in view can be selected.

To bring record into view, you have to set current bookmark. The **Row** property will only affect a row currently in display.

RowCellBackColor

The introduction of StyleSets in Version 2.0 has eliminated this property.

To accomplish the setting of the background color for the specified column for the current row:

```
SSDBGrid1.StyleSet("stylesetname").BackColor = vbRed  
SSDBGrid1.Columns(0).CellStyleSet "stylesetname", SSDBGrid1.Row
```

The above code sets the CellStyleSet to *stylesetname* for Column 0 on the current row (as indicated by SSDBGrid1.Row).

If performed in the **RowLoaded** event, the SSDBGrid1.Row parameter is not needed.

RowCellBold

The introduction of StyleSets in Version 2.0 has eliminated this property.

To accomplish the setting of the bold property for the specified column for the current row:

```
SSDBGrid1.StyleSet("stylesetname").Bold = True  
SSDBGrid1.Columns(0).CellStyleSet "stylesetname", SSDBGrid1.Row
```

The above code sets the CellStyleSet to *stylesetname* for Column 0 on the current row (as indicated by SSDBGrid1.Row)

If performed in the **RowLoaded** event, the SSDBGrid1.Row parameter is not needed.

RowCellForeColor

The introduction of StyleSets in Version 2.0 has eliminated this property.

To accomplish the setting of the foreground color for the specified column for the current row:

```
SSDBGrid1.StyleSet("stylesetname").ForeColor = vbBlue  
SSDBGrid1.Columns(0).CellStyleSet "stylesetname", SSDBGrid1.Row
```

The above code sets the CellStyleSet to *stylesetname* for Column 0 on the current row (as indicated by SSDBGrid1.Row)

If performed in the **RowLoaded** event, the SSDBGrid1.Row parameter is not needed.

RowCellItalic

The introduction of StyleSets in Version 2.0 has eliminated this property.

To accomplish the setting of the italic property for the specified column for the current row:

```
SSDBGrid1.StyleSet("stylesetname").Italic = True  
SSDBGrid1.Columns(0).CellStyleSet "stylesetname", SSDBGrid1.Row
```

The above code sets the CellStyleSet to *stylesetname* for Column 0 on the current row (as indicated by SSDBGrid1.Row)

If performed in the **RowLoaded** event, the SSDBGrid1.Row parameter is not needed.

RowCellStrikeThru

The introduction of StyleSets in Version 2.0 has eliminated this property.

To accomplish the setting of the strikethrough property for the specified column for the current row:

```
SSDBGrid1.StyleSet("stylesetname").Strikethru = True  
SSDBGrid1.Columns(0).CellStyleSet "stylesetname", SSDBGrid1.Row
```

The above code sets the CellStyleSet to *stylesetname* for Column 0 on the current row (as indicated by SSDBGrid1.Row)

If performed in the **RowLoaded** event, the SSDBGrid1.Row parameter is not needed.

RowCellUnderline

The introduction of StyleSets in Version 2.0 has eliminated this property.

To accomplish the setting of the underline property for the specified column for the current row:

```
SSDBGrid1.StyleSet("stylesetname").Underline = True  
SSDBGrid1.Columns(0).CellStyleSet "stylesetname", SSDBGrid1.Row
```

The above code sets the CellStyleSet to *stylesetname* for Column 0 on the current row (as indicated by SSDBGrid1.Row)

If performed in the **RowLoaded** event, the SSDBGrid1.Row parameter is not needed.

RowColChange

In Version 1.0:

Syntax was:

```
Sub control_RowColChange ([Index As Integer,] LastRow As Long, LastCol As Integer)
```

In Version 2.0:

Syntax is:

```
Sub control_RowColChange ([LastRow As Variant] [LastCol As Integer])
```


RowLabels

This property has been replaced with **RecordSelectors**. Usage and settings remain the same.

RowLoaded

In Version 1.0:

Syntax was:

Sub *control_RowLoaded* ([*Index* **As Integer**,] *Bookmark* **As String**, *RowNum* **As Long**)

In Version 2.0:

Syntax is

Sub *control_RowLoaded* (*Bookmark* **As Variant**)

where *Bookmark* is a variant specifying the bookmark of the row that has been loaded.

RowsResized

In Version 1.0:

Syntax was:

Sub *control*_RowsResized ([*Index As Integer*])

In Version 2.0:

Syntax is:

Sub *control*_RowResize (*Cancel As Integer*)

where *Cancel* is an integer expression that specifies whether the operation occurs.

SavedBookmark

In Version 1.0:

Applied to **SSVData**, **SSHData**, **SSDataCommand**

In Version 2.0:

Applies to **SSDBCommand**

ScrollBars

In Version 1.0:

Setting	Description
0	None. No scroll bars are displayed.
1	Horizontal. Only a horizontal scroll bar is displayed.
2	Vertical. Only a vertical scroll bar is displayed.
3	Both. Both horizontal and vertical scroll bars are displayed.

In Version 2.0:

Setting	Description
0	None
1	Horizontal only
2	Vertical only
3	Both
4	(Default) Automatically determined by SSDBGrid

SelBookmark

In Version 1.0:

Contains a collection of bookmarks of all selected rows.

Usage: *object*. **SelBookmark**(index)

In Version 2.0:

SelBookmark has become the **SelBookmarks** Collection.

The **SelBookmarks** collection represents a set of selected bookmark objects. Bookmarks are added to this collection whenever a user selects a row in the grid. If Multiselect is True, then each row selected will be added to the collection in the order in which the selection occurred. When a row is unselected, that row is then removed from the **SelBookmarks** collection.

SelChange

In Version 1.0:

Syntax was:

Sub *control*_SelChange ([*Index* **As Integer**])

In Version 2.0:

Syntax is:

Sub *control*_SelChange (*NumSelected* **As Long**, *SelType* **As Long**, *Cancel* **As Integer**)

where *NumSelected* indicates the number of rows or columns selected. *SelType* indicates the type of selection (0=Group, 1=Column, 2=Row) and *Cancel* determines whether the selection reverts to its position before the event occurred.

SelCount

This property does not exist in Version 2.0.

In Version 2.0:

Use **SelBookmarks.Count**.

SelEndBookmark

This property does not exist in Version 2.0.

In Version 2.0:

This is the last bookmark in the **SelBookmarks** collection.

SelEndCol

This property does not exist in Version 2.0.

In Version 2.0:

You can set the column's **Selected** property to True.

SelEndRow

This property does not exist in Version 2.0.

In Version 2.0:

You can add the row to the **SelBookmarks** collection.

SelRow

This property does not exist in Version 2.0.

SelfStartBookmark

This property does not exist in Version 2.0.

In Version 2.0:

This is the first bookmark in the **SelfBookmarks** collection.

SelStartCol

This property does not exist in Version 2.0.

In Version 2.0:

You can set the column's **Selected** property to True.

SelStartRow

This property does not exist in Version 2.0.

In Version 2.0:

You can add the row to the **SelBookmarks** collection.

SelectionTypeCol

Note This property is called **SelectTypeCol** in version 2.0

In Version 1.0:

Setting	Description
0	None. Column/row selection is disabled.
1	Single. Column/row selection is enabled but only one column/row can be selected at any time.
2	(Default) Multiple. The user can select a range of columns/rows by holding down the SHIFT key and selecting a range of columns/rows.
3	Tag List. Non-contiguous rows can be tagged selected.

In Version 2.0:

Setting	Description
0	None
1	(Default) Single Select
2	Multi Select, Individual selection only
3	Multi-Select, Range selection allowed

SelectionMode

Note This property is called **SelectTypeRow** in version 2.0

In Version 1.0:

Setting	Description
0	None. Column/row selection is disabled.
1	Single. Column/row selection is enabled but only one column/row can be selected at any time.
2	(Default) Multiple. The user can select a range of columns/rows by holding down the SHIFT key and selecting a range of columns/rows.
3	Tag List. Non-contiguous rows can be tagged selected.

In Version 2.0:

Setting	Description
0	None
1	(Default) Single Select
2	Multi Select, Individual selection only
3	Multi-Select, Range selection allowed

ShadowColor

Replaced by the **BevelColor** series of properties.

ShowBookmarkButton

Replaced by the **ShowBookmarkButtons** property. Functionality and settings remain the same.

ShowFirstLastButton

Replaced by the **ShowFirstLastButtons** property. Functionality and settings remain the same.

ShowPageButton

Replaced by the **ShowPageButtons** property. Functionality and settings remain the same.

ShowPrevNextButton

Replaced by the **ShowPrevNextButtons** property. Functionality and settings remain the same.

TextFieldType

This property does not exist in Version 2.0.

In Version 2.0:

Use the **DataType** property of the **Column** object.

TopRow

Replaced by the **FirstRow** property.

In Version 1.0:

TopRow was set with or returned a row number.

In Version 2.0:

FirstRow uses a bookmark.

TopRowChange

This property does not exist in Version 2.0.

In Version 2.0:

This is now handled by the **Scroll** event.

UpdateBegin

Replaced by **BeforeUpdate**.

In Version 1.0:

Syntax was:

```
Sub control_UpdateBegin ([Index As Integer,] Bookmark As String, RtnCancel As Integer)
```

In Version 2.0:

Syntax is:

```
Sub control_BeforeUpdate ([Cancel As Integer])
```

where *Cancel* is an integer expression that specifies whether the operation occurs.

UpdateEnd

Replaced by **AfterUpdate**.

In Version 1.0:

Syntax was:

Sub *control_UpdateEnd* (*[Index As Integer]*)

In Version 2.0:

Syntax is:

Sub *control_AfterUpdate* (*RtnDispErrMsg As Integer*)

where *RtnDispErrMsg* is an integer expression that indicates if an error message box should be displayed.

UpdateError

Does not exist in Version 2.0.

ValidateList

In Version 1.0:

Syntax was:

```
Sub control_ValidateList ([Index As Integer,] Text As Integer, RtnPassed As Integer,  
RtnRestore As Integer)
```

In Version 2.0:

Syntax is:

```
Sub control_ValidateList (Text As String, RtnPassed As Integer)
```

where Text is the text to validate and RtnPassed is set to notify the control that the data is valid or not. The RtnPassed parameter defaults to False.

VariableColWidth

Replaced with **AllowColumnSizing**. Functionality and settings remain the same.

VariableFixedCols

The **SplitterVisible** determines whether a Splitter bar is visible or not, which allows the fixed columns to be changed.

VariableRowHeight

Replaced with **AllowRowSizing**. Functionality and settings remain the same.

