

## **Contents**

The following Help Topics are available:

[General Overview](#)

[TIncCombo Component](#)

[TDBIncCombo Component](#)

[TDBIncLookupCombo Component](#)

[TDBIncSearchCombo Component](#)

[Support and Upgrades](#)

[Ordering Information](#)

[Return Policy](#)

[License Agreement](#)

[History](#)

**Copyright 1996 Softouch Development Inc. All rights reserved.**

## General Overview

## TIncCombo Component

[Properties](#)

[Methods](#)

[Events](#)

### Unit

IncCombo

### Description

The TIncCombo component is the most basic of the Combo Collection. This component is equivalent to Delphi's standard combo box with the exception of its incremental properties.

In addition to these properties, methods, and events, this component also has the properties, methods, and events that apply to all windowed controls.

## Properties

### ▶ Run-Time only

<a href="#"><u>AutoDropDown</u></a>	<a href="#"><u>DropDownOptions</u></a>	<a href="#"><u>Name</u></a>	<a href="#"><u>Style</u></a>
<a href="#"><u>AutoSize</u></a>	<a href="#"><u>Enabled</u></a>	<a href="#"><u>ParentColor</u></a>	<a href="#"><u>TabOrder</u></a>
<a href="#"><u>BorderStyle</u></a>	<a href="#"><u>Font</u></a>	<a href="#"><u>ParentCtl3D</u></a>	<a href="#"><u>TabStop</u></a>
<a href="#"><u>ButtonCursor</u></a>	▶ <a href="#"><u>Found</u></a>	<a href="#"><u>ParentFont</u></a>	<a href="#"><u>Tag</u></a>
<a href="#"><u>CharCase</u></a>	<a href="#"><u>Height</u></a>	<a href="#"><u>ParentShowHint</u></a>	<a href="#"><u>Top</u></a>
<a href="#"><u>Color</u></a>	<a href="#"><u>HelpContext</u></a>	<a href="#"><u>PopupMenu</u></a>	<a href="#"><u>Visible</u></a>
▶ <a href="#"><u>ComboButtonVisible</u></a>	<a href="#"><u>Hint</u></a>	<a href="#"><u>ReadOnly</u></a>	▶ <a href="#"><u>VisibleRows</u></a>
<a href="#"><u>Ctl3D</u></a>	▶ <a href="#"><u>ItemIndex</u></a>	▶ <a href="#"><u>SearchResult</u></a>	<a href="#"><u>Width</u></a>
<a href="#"><u>Cursor</u></a>	<a href="#"><u>Items</u></a>	▶ <a href="#"><u>SearchText</u></a>	
<a href="#"><u>DragCursor</u></a>	<a href="#"><u>Left</u></a>	<a href="#"><u>ShowComboButton</u></a>	
<a href="#"><u>DragMode</u></a>	<a href="#"><u>MaxLength</u></a>	<a href="#"><u>ShowHint</u></a>	

## Methods

<a href="#"><u>CanDropDown</u></a>	<a href="#"><u>Destroy</u></a>
<a href="#"><u>CanModify</u></a>	<a href="#"><u>DropDown</u></a>
<a href="#"><u>CloseUp</u></a>	<a href="#"><u>DropDownList</u></a>
<a href="#"><u>Create</u></a>	

## Events

<a href="#"><u>OnChange</u></a>	<a href="#"><u>OnDragDrop</u></a>	<a href="#"><u>OnEnter</u></a>	<a href="#"><u>OnKeyUp</u></a>
<a href="#"><u>OnClick</u></a>	<a href="#"><u>OnDragOver</u></a>	<a href="#"><u>OnExit</u></a>	<a href="#"><u>OnMouseDown</u></a>
<a href="#"><u>OnCloseUp</u></a>	<a href="#"><u>OnDropDown</u></a>	<a href="#"><u>OnKeyDown</u></a>	<a href="#"><u>OnMouseUp</u></a>
<a href="#"><u>OnDbClick</u></a>	<a href="#"><u>OnEndDrag</u></a>	<a href="#"><u>OnKeyPress</u></a>	<a href="#"><u>OnNewItem</u></a>

## License Agreement

Grant of License for evaluators of this software product. Evaluator is granted the right to evaluate the shareware version of this software product and to distribute the unmodified shareware files to others, at no cost, for evaluation purposes only.

Grant of License for purchased software products. Purchaser is granted the right to create and distribute executable files using this software product. Purchaser may not distribute any version of this software product whether it be the original software product or a modified version thereof except as provided by above.

EVALUATOR OR PURCHASER ASSUMES ALL RESPONSIBILITIES AND RISKS ASSOCIATED WITH THE USE OF THE THIS SOFTWARE PRODUCT. THE AUTHORS OF THIS SOFTWARE PRODUCT ASSUME NO RESPONSIBILITY OR LIABILITY BEYOND THE COST OF THE COMPONENT FOR ANY ACTIONS ASSOCIATED WITH THIS SOFTWARE PRODUCT.

IF YOU DO NOT AGREE WITH ANY OF THE TERMS LISTED, PLEASE RETURN THE SOFTWARE PRODUCT IMMEDIATELY FOR A REFUND.

## **Support and Upgrades**

Registration of the Incremental Combo Collection entitles the purchaser to 30 days of free support via CompuServe and 6 months of free upgrades. Beyond the initial 30 day period, submission of bugs are always welcome and we will do are best to respond back to them in a reasonable amount of time.

## Ordering Information

To register a copy of the Increment Combo Collection using Compuserve, GO SWREG and search for IncCombo.

You can mail us at  
Softouch Development Inc.  
12320 S.W. 128 Street  
Tigard, Oregon 97223  
CIS: 102565,2117

## **Return Policy**

As always, our components come with a 30 day money back guarantee.

## AutoDropDown Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property AutoDropDown: TAutoDropDown;
```

### Description

AutoDropDown lets you determine at what point the drop down list will automatically drop down.

## TAutoDropDown Type

### Unit

IncBase

### Declaration

#### type

```
TAutoDropDown = (adNever, adFocused, adChanged);
```

### Description

The TAutoDropDown type defines the possible values for the [AutoDropDown](#) property of the [TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components, as described in the following table:

Value	Description
adNever	The drop down list will never automatically drop down.
AdFocused	The drop down list will drop down when the control gets focus.
AdChanged	The drop down list will drop down when the text is changed.

## ButtonCursor Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property ButtonCursor: TCursor;
```

### Description

The ButtonCursor property is the image used when the mouse passes over the controls combo button.

## ComboButtonVisible Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** ComboButtonVisible: Boolean;

### Description

Run-time and read only, the ComboButtonVisible property is used to determine if the combo button is actually visible at any given time.

## SearchResult Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property SearchResult: String;
```

### Description

Run-time and read only, the SearchResult property is the actually text value found within the dataset or list.

## SearchText Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property SearchText: String;
```

### Description

Run-time and read only, the SearchText property is the text value used for searching.

## ShowComboButton Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** ShowComboButton: [TShowComboButton](#);

### Description

The ShowComboButton property is used to determine when the combo button will be visible.

## TShowComboButton Type

### Unit

IncBase

### Declaration

#### type

```
TShowComboButton = (scNever, scAlways, scFocused);
```

### Description

The TShowComboButton type defines the possible values for the [ShowComboButton](#) property of the [TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components, as described in the following table:

Value	Description
scNever	The combo button will never be visible.
scAlways	The combo button will always be visible.
scFocused	The combo button will be visible when the control gets focus.

## Style Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** Style: [TComboStyle](#);

### Description

The Style property is used to determine what type of combo box this control is and how it will function.

## VisibleRows Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** VisibleRows: Integer;

### Description

Run-time and read only, the VisibleRows property is used to determine the number of visible rows within the drop down list. To change the number of rows that appear within the drop down list, set the [MaxRows](#) property within the [DropDownOptions](#) property for this control. The only time [MaxRows](#) will not equal VisibleRows is when the drop down list is not visible or the items within the drop down list are less than [MaxRows](#). If the list is not visible, VisibleRows will equal 0.

## TComboStyle Type

### Unit

IncBase

### Declaration

#### type

```
TComboStyle = (csIncSearch, csIncSearchEdit);
```

### Description

The TComboStyle type defines the possible values for the [Style](#) property of the [TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components, as described in the following table:

Value	Description
csIncSearch	Only values within the drop down list are valid values.
CsIncSearchEdit	New values can be entered.

## TDBIncCombo Component

[Properties](#)

[Methods](#)

[Events](#)

### Unit

IncDBCmb

### Description

The TDBIncCombo component is a Combo Box with data-aware capabilities. It uses a standard single column drop down list whos items can be changed through the Items property. This component is equivalent to Delphis standard data-aware combo box with the exception of its incremental properties.

In addition to these properties, methods, and events, this component also has the properties, methods, and events that apply to all windowed controls.

## Properties

### ▶ Run-Time only

<a href="#">AutoDropDown</a>	<a href="#">DragCursor</a>	<a href="#">Left</a>	<a href="#">ShowComboButton</a>
<a href="#">AutoSize</a>	<a href="#">DragMode</a>	<a href="#">MaxLength</a>	<a href="#">ShowHint</a>
<a href="#">BorderStyle</a>	<a href="#">DropDownOptions</a>	<a href="#">Name</a>	<a href="#">Style</a>
<a href="#">ButtonCursor</a>	<a href="#">Enabled</a>	<a href="#">ParentColor</a>	<a href="#">TabOrder</a>
<a href="#">CharCase</a>	<a href="#">Font</a>	<a href="#">ParentCtl3D</a>	<a href="#">TabStop</a>
<a href="#">Color</a>	<a href="#">Font</a>	<a href="#">ParentFont</a>	<a href="#">Tag</a>
▶ <a href="#">ComboButtonVisible</a>	<a href="#">Height</a>	<a href="#">ParentShowHint</a>	<a href="#">Top</a>
<a href="#">Ctl3D</a>	<a href="#">HelpContext</a>	<a href="#">PopupMenu</a>	<a href="#">Visible</a>
<a href="#">Cursor</a>	<a href="#">Hint</a>	<a href="#">ReadOnly</a>	▶ <a href="#">VisibleRows</a>
<a href="#">DataField</a>	▶ <a href="#">ItemIndex</a>	▶ <a href="#">SearchResult</a>	<a href="#">Width</a>
<a href="#">DataSource</a>	<a href="#">Items</a>	▶ <a href="#">SearchText</a>	

## ItemIndex Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#) components

### Declaration

```
property ItemIndex: Integer;
```

### Description

Run-time only. The value of the ItemIndex property is the ordinal number of the selected item in the control's [items](#) property. If no item is selected, the value is -1. To select an item at run time, set the value of ItemIndex to the index of the item in the list you want selected, with 0 being the first item in the drop down list.

## Items Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#) components

### Declaration

**property** Items: [TStrings](#);

### Description

The Items property contains the strings that appear in the drop down list box. Because Items is an object of type [TStrings](#), you can add, delete, insert, and move items using the Add, Delete, Insert, Exchange, and Move methods of the [TStrings](#) object.

The [ItemIndex](#) property determines which item is selected, if any.

## Found Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** Found: Boolean;

### Description

The Found property is used to determine if the actual text for the combo box exist in the drop down list.

## CanDropDown Method

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
function CanDropDown: Boolean;
```

### Description

The CanDropDown method determines whether this control can drop down its list box.

## CanModify Method

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
function CanModify: Boolean;
```

### Description

The CanModify method determines whether this controls text value can be edited.

## CloseUp Method

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
procedure CloseUp;
```

### Description

The CloseUp method causes the drop down list for this control to close up and hide.

## DropDown Method

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
procedure DropDown(State: TDropDownState);
```

### Description

The DropDown method causes the drop down list for this control to drop down and become visible.

## DropDownList Method

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
function DropDownList: TIncGridWinControl;
```

### Description

The DropDownList method returns a pointer to the actual drop down list. For [TIncCombo](#) and [TDBIncCombo](#), this would be a TCustomListBox. For [TDBIncLookupCombo](#) and [TDBIncSearchCombo](#), this would be a TCustomDBGrid.

## TIncGridWinControl Type

### Unit

IncBase

### Declaration

#### type

```
TIncGridWinControl = class(TWinControl)
```

### Description

The TIncGridWinControl type defines an object declaration of type TWinControl with the exception of exposing the Color and Font property. This type is used to define the return value for the [DropDownList](#) method.

## OnNewItem Event

### Applies to

[TIncCombo](#), [TDBIncCombo](#) components

### Declaration

**property** OnNewItem: [TNewItemEvent](#);

### Description

The OnNewItem event gets called whenever a new item is typed into the combo box that does not exist in the list. It is up to the user, within this event, to add the new item to the Items property or abort from this event.

If for any reason, you decide not to update the items property, you must abort from this event. If you do not have this event assigned, and a new item needs to be added, the control will abort the addition automatically and revert the text back to its original value.

## TNewItemEvent Type

### Unit

IncCombo

### Declaration

#### type

TNewItemEvent = procedure(Sender: TCustomIncListBoxCombo) of object;

### Description

The TNewItemEvent type defines the event type for when a new items is type into a combo box for either a [TIncCombo](#) or a [TDBIncCombo](#) component.

## DropDownOptions Property

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

For [TIncCombo](#), [TDBIncCombo](#)

property DropDownOptions: [TListBoxOptions](#);

For [TDBIncLookupCombo](#), [TDBIncSearchCombo](#)

property DropDownOptions: [TDBListBoxOptions](#)

### Description

The DropDownOptions property is an object that defines the attributes of the drop down list box.

## TListBoxOptions Object

[Properties](#)

### Unit

IncCombo

### Description

The TListBoxOptions object defines the attributes of the drop down list box for a TIncCombo or a TDBIncCombo [DropDownOptions](#) property

## Properties

<a href="#"><u>AlignHorizontally</u></a>	<a href="#"><u>MaxRows</u></a>
<a href="#"><u>AlignVertically</u></a>	<a href="#"><u>ParentColor</u></a>
<a href="#"><u>Color</u></a>	<a href="#"><u>ParentCtl3D</u></a>
<a href="#"><u>Ctl3D</u></a>	<a href="#"><u>ParentFont</u></a>
<a href="#"><u>Cursor</u></a>	<a href="#"><u>Sorted</u></a>
<a href="#"><u>Font</u></a>	<a href="#"><u>Width</u></a>

## AlignHorizontally Property

### Applies to

[TListBoxOptions](#) object for [TIncCombo](#), [TDBIncCombo](#) components and the [TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** AlignHorizontally: [TDropDownAlignHorizontally](#);

### Description

The AlignHorizontally property defines where the drop down list drops down horizontally in respect to its edit box.

## TDropDownAlignHorizontally Type

### Unit

IncBase

### Declaration

#### type

```
TDropDownAlignHorizontally = (daLeft,daRight);
```

### Description

The TDropDownAlignHorizontally type defines the possible values for the [AlignHorizontally](#) property of the [TListBoxOptions](#) object for [TIncCombo](#), [TDBIncCombo](#) components, and the [TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components, as described in the following table:

Value	Description
daLeft	Align drop down list to the left side of the combo boxes edit field.
daRight	Align drop down list to the right side of the combo boxes edit field.

## AlignVertically Property

### Applies to

[TListBoxOptions](#) object for [TIncCombo](#), [TDBIncCombo](#) components and the [TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** AlignVertically: [TDropDownAlignVertically](#);

### Description

The AlignVertically property defines where the drop down list drops down vertically in respect to its edit box

## **TDropDownAlignVertically Type**

### **Unit**

IncBase

### **Declaration**

#### **type**

```
TDropDownAlignVertically = (daTop,daBottom);
```

### **Description**

The TDropDownAlignVertically type defines the possible values for the [AlignVertically](#) property of the [TListBoxOptions](#) object for [TIncCombo](#), [TDBIncCombo](#) components, and the [TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components, as described in the following table:

<b>Value</b>	<b>Description</b>
daTop	Drop down list will appear above the combo boxes edit field.
daBottom	Drop down list will drop down below the combo boxes edit field.

## MaxRows Property

### Applies to

[TListBoxOptions](#) object for [TIncCombo](#), [TDBIncCombo](#) components and the [TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property MaxRows: Integer;
```

### Description

The MaxRows property defines how many rows can be displayed within the drop down list. If the items within the drop down list are less than the MaxRows then the rows displayed will not equal MaxRows. To get the actual row count of the drop down list, use the [VisibleRows](#) property.

## DataField Property

### Applies to

[TDBIncCombo](#), [TDBIncLookupCombo](#) components

### Declaration

**property** DataField: String;

### Description

For [TDBIncCombo](#), the DataField property identifies the field from which the data-aware control displays data.

For [TDBIncLookupComb](#), the DataField property identifies the field from which the data-aware control uses to link to the [LookupField](#) property. The DataField in this instance is not the display value. The display value is determined by the [LookupDisplay](#) property.

The [DataSource](#) property determines where the component obtains the data.

## DataSource Property

### Applies to

[TDBIncCombo](#), [TDBIncLookupCombo](#) components

### Declaration

```
property DataSource: TDataSource;
```

### Description

The DataSource property determines where the component obtains the data. Specify the data source component that identifies the dataset the data is found in. This field works in conjunction with [DataField](#).

## TDBIncLookupCombo Component

[Properties](#)

[Methods](#)

[Events](#)

### Unit

IncDBLup

### Description

The TDBIncLookupCombo component is a combo box with data-aware capabilities that builds its lookup list from another dataset. This component is equivalent to Delphi's standard data-aware lookup combo box with the exception of its incremental properties.

In addition to these properties, methods, and events, this component also has the properties, methods, and events that apply to all windowed controls.

## Properties

### ▶ Run-Time only

<a href="#">AutoDropDown</a>	<a href="#">DragCursor</a>	<a href="#">LookupDisplay</a>	<a href="#">ShowComboButton</a>
<a href="#">AutoSelect</a>	<a href="#">DragMode</a>	<a href="#">LookupField</a>	<a href="#">ShowHint</a>
<a href="#">AutoSize</a>	<a href="#">DropDownOptions</a>	<a href="#">LookupIndex</a>	<a href="#">Style</a>
<a href="#">AutoSync</a>	<a href="#">Enabled</a>	<a href="#">LookupSource</a>	<a href="#">TabOrder</a>
<a href="#">BorderStyle</a>	<a href="#">ExtendedSearch</a>	<a href="#">MaxLength</a>	<a href="#">TabStop</a>
<a href="#">ButtonCursor</a>	<a href="#">Font</a>	<a href="#">Name</a>	<a href="#">Tag</a>
<a href="#">CharCase</a>	▶ <a href="#">Found</a>	<a href="#">ParentColor</a>	<a href="#">Top</a>
<a href="#">Color</a>	<a href="#">Height</a>	<a href="#">ParentCtl3D</a>	▶ <a href="#">Value</a>
<a href="#">ComboButtonVisible</a>	<a href="#">HelpContext</a>	<a href="#">ParentFont</a>	<a href="#">Visible</a>
▶ <a href="#">Ctl3D</a>	<a href="#">Hint</a>	<a href="#">ParentShowHint</a>	▶ <a href="#">VisibleRows</a>
<a href="#">Cursor</a>	▶ <a href="#">ItemIndex</a>	<a href="#">PopupMenu</a>	<a href="#">Width</a>
<a href="#">DataField</a>	<a href="#">Items</a>	<a href="#">ReadOnly</a>	
<a href="#">DataSource</a>	<a href="#">Left</a>	▶ <a href="#">SearchResult</a>	
▶ <a href="#">DisplayDBFld</a>	▶ <a href="#">LookupDBFld</a>	▶ <a href="#">SearchText</a>	

## AutoSync Property

### Applies to

[TDBIncLookupCombo](#) component

### Declaration

```
property AutoSync: Boolean;
```

### Description

The AutoSync property is used to determine if external navigation of the [LookupSource](#) dataset will cause changes to the controls values. The only time this property will have any effect is if the control has focus.

## BufferedLookupSource Property

### Applies to

[TDBIncSearchCombo](#) component

### Declaration

```
property BufferedLookupSource: TDataSource;
```

### Description

Run-time and read only, the BufferedLookupSource is a pointer to the internal LookupSource used by this control. The BufferedLookupSource is the actual datasource used for the drop down list. This BufferedLookupSource is needed so that when the drop down list is scrolled, the [LookupSource](#) dataset position does not change.

## DisplayDBFId Property

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** DisplayDBFId: TField;

### Description

Run-time and read only, the DisplayDBFId property is a pointer to the TField that is used as the display field for the edit box. This field is located within the [BufferedLookupSource](#). The fields name is always the first field within [LookupDisplay](#) property.

## ExtendedSearch Property

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** ExtendedSearch: Boolean;

### Description

The ExtendedSearch property, if set to True, will cause all searches to be treated as numbers and padded to right with zeros until the length of the field has been exhausted or a match is found.

## LookupDBFId Property

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property LookupDBFId: TField;
```

### Description

Run-time and read only, the LookupDBFId property is a pointer to the TField that is used as the Lookup Field for the edit box. This field is located within the [BufferedLookupSource](#). The field's name is always equal to the [LookupField](#) or the first field within [LookupDisplay](#) property for [TDBIncSearchCombo](#).

## LookupDisplay Property

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property LookupDisplay: String;
```

### Description

The LookupDisplay property determines which field in the lookup table displays in the database lookup combo box. Before you specify a LookupDisplay field, link the two datasets using the [LookupField](#) property.

You can choose to display multiple fields from the lookup dataset. Each field appears in a separate column. To specify more than one field to display, separate each field name with a semicolon. For example, this line of code displays three columns in the drop-down list of a database lookup combo box. Column 1 is the name of the company, column 2 is the city where the company is located, and column 3 is the country.

```
DBLookupCombo1.LookupDisplay := 'Company;City;Country';
```

The actual edit field for the combo box will display Company and this will be the value that will get incrementally searched one when the user types in text.

Within Delphi 16Bit, [LookupSource](#) dataset must also contain an index that starts with this field.

## LookupField Property

### Applies to

[TDBInclLookupCombo](#) component

### Declaration

**property** LookupField: String;

### Description

The LookupField property links the [LookupSource](#) dataset to the [DataSource](#) dataset you are working with.

Although the name of the field specified as the LookupField does not have to be the same as the name of the field specified as the [DataField](#), the two fields must contain the same values. For example, the LookupField value can be CustomerNumber and the [DataField](#) value can be CustNo, as long as both fields use the same number to identify a particular customer.

Within Delphi 16Bit, [LookupSource](#) dataset must also contain an index that starts with this field.

After you specify a LookupField field, you can choose which field you prefer to display in the control with the [LookupDisplay](#) property.

## LookupSource Property

### Applies to

[TDBInclLookupCombo](#), [TDBInclSearchCombo](#) components

### Declaration

**property** LookupSource: TDataSource;

### Description

The LookupSource of a database lookup combo box is the data source component (TDataSource) that identifies the dataset you want the control to use to "look up" the information you want displayed in the control.

In [TDBInclLookupCombo](#) any navigational changes to LookupSource can be reflected in the values of the control by setting [AutoSync](#) to True.

In [TDBInclSearchCombo](#) this is the datasource used to navigate the table. Any navigational changes to LookupSource will be reflected in the values of the control.

Within Delphi 16Bit, the LookupSource dataset must be a TTable unless you assign events to do custom searching. Those events are either [OnCustomSearch](#) or [OnCustomSearchRec](#) event and if the control is a [TDBInclLookupCombo](#) then the [OnCustomLocate](#) event must be assigned. In Delphi 32Bit, the LookupSource dataset can be either TTable or TQuery.

## Value Property

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** Value: String;

### Description

Run-time and read only.

For [TDBIncLookupCombo](#) the Value property is equal to the actual value for [DataField](#). This value can also be obtained using the [LookupDBFld](#) property.

For [TDBIncSearchCombo](#) the Value property is equal to the actual value of the first field of [LookupDisplay](#). This value can also be obtained using the [DisplayDBFld](#) property.

## Events

<a href="#"><u>OnChange</u></a>	<a href="#"><u>OnCustomSearchRec</u></a>	<a href="#"><u>OnEndDrag</u></a>	<a href="#"><u>OnKeyUp</u></a>
<a href="#"><u>OnClick</u></a>	<a href="#"><u>OnDbClick</u></a>	<a href="#"><u>OnEnter</u></a>	<a href="#"><u>OnMouseDown</u></a>
<a href="#"><u>OnCloseUp</u></a>	<a href="#"><u>OnDragDrop</u></a>	<a href="#"><u>OnExit</u></a>	<a href="#"><u>OnMouseUp</u></a>
<a href="#"><u>OnCustomLocate</u></a>	<a href="#"><u>OnDragOver</u></a>	<a href="#"><u>OnKeyDown</u></a>	<a href="#"><u>OnNewRecord</u></a>
<a href="#"><u>OnCustomSearch</u></a>	<a href="#"><u>OnDropDown</u></a>	<a href="#"><u>OnKeyPress</u></a>	

## OnCustomSearch Event

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

property OnCustomSearch: [TSearchNotifyEvent](#);

### Description

The OnCustomSearch event, when assigned, gets called when the control would normally do its incremental lookups. If you assign a procedure to this event, the default searching will no longer happen. If [ExtendedSearch](#) is set to True, this event will still only get called once for every key typed and [ExtendedSearch](#) will have no purpose. If you would like for [ExtendedSearch](#) to still work and would still like to do custom searching, then use the [OnCustomSearchRec](#) event.

It will be up to the assigned event to set the index appropriately for the lookup. The control will automatically reset the index after the call to this event.

In Delphi 16bit if you desire to use a TQuery as the [LookupSource](#) dataset then you must either use this event or the [OnCustomSearchRec](#) event to do your incremental searching. If this control is a [TDBIncLookupCombo](#) you will also have to assign an event to then [OnCustomLocate](#) event.

## OnCustomSearchRec Event

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** OnCustomSearchRec: [TSearchNotifyEvent](#);

### Description

The OnCustomSearchRec event, when assigned, gets called when the control would normally do its actual locate or lookup for the record during incremental searches. If you assign a procedure to this event, the default searching will no longer happen. If [ExtendedSearch](#) is set to True, this event will get called multiple times for each lookup.

It will be up to the assigned event to set the index appropriately for the lookup. The control will automatically reset the index after the call to this event.

In Delphi 16bit if you desire to use a TQuery as the [LookupSource](#) dataset then you must either use this event or the [OnCustomSearch](#) event to do your incremental searching. If this control is a [TDBIncLookupCombo](#) you will also have to assign an event to then [OnCustomLocate](#) event.

## OnNewRecord Event

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** OnNewRecord: [TNewRecordEvent](#);

### Description

The OnNewRecord event gets called whenever a new item is typed into the combo box that does not exist in the list. It is up to the user, within this event, to add the new item to the [LookupSource](#) dataset and for [TDBIncLookupCombo](#) set the datafields actual value to link appropriately with the [LookupField](#). This event actually gets called when the control loses focus.

If for any reason, you decide not to update the dataset, you must abort from this event. If you do not have this event assigned, and a new item needs to be added, the control will abort the addition automatically.

## TNewRecordEvent Type

### Unit

IncDBSrc

### Declaration

#### type

TNewRecordEvent = procedure(Sender: TObject; DataSet: TDataSet) of object;

### Description

The TNewRecordEvent type defines the event type for when a new items is type into a combo box for either a [TDBIncLookupCombo](#) or a [TDBIncSearchCombo](#).

## **TSearchNotifyEvent Type**

### **Unit**

IncDBSrc

### **Declaration**

#### **type**

TSearchNotifyEvent = procedure(Sender: TObject; DataSet: TDataSet; SearchText: String) of object;

### **Description**

The TSearchNotifyEvent type defines the event type for doing custom lookups within the DataSet. SearchText is the actual text value to search on.

## TBDropDownOptions Object

[Properties](#)

### Unit

IncDBSrc

### Description

The [TBDropDownOptions](#) object defines the attributes of the drop down list box for a [TDBIncLookupCombo](#) or a [TDBIncSearchCombo](#) [DropDownOptions](#) property

## Properties

<a href="#"><u>AlignHorizontally</u></a>	<a href="#"><u>ParentColor</u></a>
<a href="#"><u>AlignVertically</u></a>	<a href="#"><u>ParentCtl3D</u></a>
<a href="#"><u>ColLines</u></a>	<a href="#"><u>ParentFont</u></a>
<a href="#"><u>Color</u></a>	<a href="#"><u>RowLines</u></a>
<a href="#"><u>Ctl3D</u></a>	<a href="#"><u>TitleColor</u></a>
<a href="#"><u>Cursor</u></a>	<a href="#"><u>TitleFont</u></a>
<a href="#"><u>Font</u></a>	<a href="#"><u>Titles</u></a>
<a href="#"><u>MaxRows</u></a>	<a href="#"><u>Width</u></a>

## ColLines Property

### Applies to

[TBBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** ColLines: Boolean;

### Description

The ColLines property defines whether or not column lines will appear in the drop down list or not.

## RowLines Property

### Applies to

[TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** RowLines: Boolean;

### Description

The RowLines property defines whether or not row lines will appear in the drop down list or not.

## TitleColor Property

### Applies to

[TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property TitleColor: TColor;
```

### Description

The TitleColor property defines the color to be used for the title of the drop down list.

## TitleFont Property

### Applies to

[TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property TitleFont: TFont;
```

### Description

The TitleFont property defines the font to be used for the title of the drop down list.

## Titles Property

### Applies to

[TDBDropDownOptions](#) object for [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** Titles: Boolean;

### Description

The Titles property defines whether or not the drop down list will have titles or not. Title names are taken from the [LookupDisplay](#) property.

## TDBIncSearchCombo Component

[Properties](#)

[Methods](#)

[Events](#)

### Unit

IncDBSrc

### Description

The TDBIncSearchCombo is a incremental combo box used for navigating a datasource. A the user types in a value in the edit box, the [LookupSource](#) dataset is navigated to the new location. Likewise if the [LookupSource](#) dataset is navigated, the value in the edit box changes to match the new location.

In addition to these properties, methods, and events, this component also has the properties, methods, and events that apply to all windowed controls.

## Properties

### ▶ Run-Time only

<a href="#">AutoDropDown</a>		<a href="#">DataSource</a>		<a href="#">Left</a>		<a href="#">SearchText</a>
<a href="#">AutoInsert</a>	▶	<a href="#">DisplayDBFld</a>	▶	<a href="#">LookupDBFld</a>		<a href="#">ShowComboButton</a>
<a href="#">AutoSelect</a>		<a href="#">DragCursor</a>		<a href="#">LookupDisplay</a>		<a href="#">ShowHint</a>
<a href="#">AutoSize</a>		<a href="#">DragMode</a>		<a href="#">LookupIndex</a>		<a href="#">Style</a>
<a href="#">AutoSuppressCancel</a>		<a href="#">DropDownOptions</a>		<a href="#">LookupSource</a>		<a href="#">TabOrder</a>
<a href="#">BorderStyle</a>		<a href="#">Enabled</a>		<a href="#">MaxLength</a>		<a href="#">TabStop</a>
▶ <a href="#">BufferedLookupSource</a>		<a href="#">ExtendedSearch</a>		<a href="#">Name</a>		<a href="#">Tag</a>
<a href="#">ButtonCursor</a>		<a href="#">Font</a>		<a href="#">ParentColor</a>		<a href="#">Top</a>
<a href="#">CharCase</a>	▶	<a href="#">Found</a>		<a href="#">ParentCtl3D</a>	▶	<a href="#">Value</a>
<a href="#">Color</a>		<a href="#">Height</a>		<a href="#">ParentFont</a>		<a href="#">Visible</a>
<a href="#">ComboButtonVisible</a>		<a href="#">HelpContext</a>		<a href="#">ParentShowHint</a>	▶	<a href="#">VisibleRows</a>
▶ <a href="#">Ctl3D</a>		<a href="#">Hint</a>		<a href="#">PopupMenu</a>		<a href="#">Width</a>
<a href="#">Cursor</a>	▶	<a href="#">ItemIndex</a>		<a href="#">ReadOnly</a>		
<a href="#">DataField</a>		<a href="#">Items</a>	▶	<a href="#">SearchResult</a>		

## AutoInsert Property

### Applies to

[TDBIncSearchCombo](#) component

### Declaration

**property** AutoInsert: Boolean;

### Description

The AutoInsert property determines if a new text value is type into the edit control if the control will place the [LookupSource](#) dataset into insert mode automatically.

## AutoSuppressCancel Property

### Applies to

[TDBIncSearchCombo](#) component

### Declaration

**property** AutoSuppressCancel: Boolean;

### Description

The AutoSuppressCancel property determine if when its time to cancel changes because the control needs to lookup a new value, if the control will suppress the OnBeforeCancel event for the [LookupSource](#) dataset. The only time this property will have any effect is when the control starts the inserting and the user has not left the controls edit field. If the user instigates the edit or insert, the OnBeforeCancel event will not get suppressed regardless of the setting of the property.

## LookupIndex Property

### Applies to

[TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

```
property LookupIndex: String;
```

### Description

The LookupIndex property is determines the index used when doing incremental searching on the [LookupSource](#). This property is only valid when [LookupSource](#) dataset inherits from TTable.

**NOTE:** You only need to set this field if you have some specialized indexing needs. If left blank, the control will look for an index that starts with the first field in the [LookupDisplay](#) property.

## OnCustomLocate Event

### Applies to

[TDBIncLookupCombo](#) component

### Declaration

**property** OnCustomLocate: [TSearchNotifyEvent](#);

### Description

The OnCustomLocate event, when assigned, gets called when the control would normally try to locate the [LookupField](#) using the [DataField](#). If you assign a procedure to this event, the default locating will no longer happen.

It will be up to the assigned event to set the index appropriately for the lookup. The control will automatically reset the index after the call to this event.

In Delphi 16bit if you desire to use a TQuery as the [LookupSource](#) dataset then you must use this event along with either the [OnCustomSearch](#) event or the [OnCustomSearchRec](#) event to do your incremental searching.

## Events

[OnChange](#)

[OnClick](#)

[OnCloseUp](#)

[OnCustomSearch](#)

[OnCustomSearchRec](#)

[OnDbClick](#)

[OnDragDrop](#)

[OnDragOver](#)

[OnDropDown](#)

[OnEndDrag](#)

[OnEnter](#)

[OnExit](#)

[OnKeyDown](#)

[OnKeyPress](#)

[OnKeyUp](#)

[OnMouseDown](#)

[OnMouseUp](#)

[OnNewRecord](#)

## OnDropDown Event

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

property OnDropDown: [TDropDownEvent](#);

### Description

The OnDropDown event gets called right before the drop down list appears.

## TDropDownEvent Type

### Unit

IncBase

### Declaration

#### type

TDropDownEvent = procedure(Sender: TObject; State: [TDropDownState](#)) of object;

### Description

The TDropDownEvent type defines the event type for when the drop down list is about to appear. The [DropDownState](#) defines the possible reasons why the list is about to appear.

## TDropDownState Type

### Unit

IncBase

### Declaration

#### type

```
TDropDownState = (dsChanged,dsFocused,dsUserAction);
```

### Description

The TDropDownState type defines the possible values for the reason why the drop down list is about to appear. This type gets passed as a parameter for the [OnDropDown](#) event and the [DropDown](#) method.

The possible values are described in the following table:

Value	Description
dsChanged	The text changed.
dsFocused	The control got focus.
dsUserAction	The user pushed the combo button.

## OnCloseUp Event

### Applies to

[TIncCombo](#), [TDBIncCombo](#), [TDBIncLookupCombo](#), [TDBIncSearchCombo](#) components

### Declaration

**property** OnCloseUp: TNotifyEvent;

### Description

The OnCloseUp event gets called right before the drop down list disappears.

## History

### **Version 1.8**

**06/11/96** Fixed problem in 32 bit with TQuery not working as the [LookupSource](#) in [TDBIncLookupCombo](#).

**06/11/96** Made About box declaration less generic to remove any conflicts with other controls.

**06/13/96** Within [TDBIncSearchCombo](#) and [TDBIncLookupCombo](#), fix problem with calculation of the width of the drop down list If DropDownOptions.Width = 0.

**06/13/96** Check for horizontal clipping of the drop down list and realign as needed.

**06/13/96** Within [TDBIncSearchCombo](#) and [TDBIncLookupCombo](#), fix problem with TQuery not being uniquely indexed and causing an exception when user tries to drop down the list.

**Released 06/13/96**

### **Version 1.9**

**06/19/96** Fixed problem with changing the font size of the control and causing the combo button to get overwritten.

**06/19/96** Fixed problem with combo button getting clipped by text when there is no border or the control is not 3D.

**06/19/96** Added a new event, [OnDropDown](#), for when the drop down list appears.

**06/19/96** Added a new event, [OnCloseUp](#), for when the drop down list disappears.

**06/19/96** In [TDBIncSearchCombo](#), fix problem that prevented user from inserting new records externally.

**06/19/96** In all controls, start allowing the drop down list to appear even if empty.

**06/19/96** In [TIncCombo](#) fix problem that occurs when [OnNewItem](#) is not assigned and a user clicks into the control and cant get out without typing a valid item from the list.

**06/19/96** Fixed problem with drop down list not appearing if the user clicks on the combo button and [AutoDropDown](#) = adFocused.

**06/19/96** [CanDropDown](#) method will no longer be affected by whether or not there are items in the drop down list.

**06/19/96** The combo button appearing disabled in 32Bit will no longer be affected by whether or not there are items in the drop down list.

**06/19/96** Fixed problem in [TDBIncLookupCombo](#) when trying to locate the [DataField](#) in the [LookupField](#) and DataFields actual value is blank. This happens when you insert a new record and the DataField is not a string type field.

**06/21/96** When there is no border, offset text top and left by 2 so that it aligns better within grids.

**06/21/96** In [TIncCombo](#) and [TDBIncCombo](#) when the user sets [ItemIndex](#) to -1, always clear the text.

**06/24/96** In Delphi 16bit, changed formula for drawing combo button.

**Released 06/25/96**

