

# #<sup>1</sup>\$<sup>2</sup>K<sup>3</sup>+<sup>4</sup>axCool Help

Properties

Methods

Events

axCool is a toolbar control containing a collection of Button objects used to create a toolbar that is associated with an application.

Typically, a toolbar contains buttons that correspond to items in an application's menu, providing a graphic interface for the user to access an application's most frequently used functions and commands.

## File Name

axCool.ocx

## Distribution

- Unzip the source code with directories intact.
- Load the sample Project1 with Visual Basic. This should register the OCX for use with other VB applications. If it does not load correctly, then from the DOS prompt, move to the install directory and type: **regsvr32 axCool.ocx** (you may need to put **c:\windows\system\** in front of regsvr32 if you don't have a path).
- Whenever you want to add the control to a VB application, go to the Project/Components menu and select "ActiveX Coolbar".

Revisions

License

Tech-Support

1Hlp\_Contents

2Contents

3Contents

4Browse:0005

# #567+8 Revisions

## 1.3

- Initial Release

## 1.4

- Removed AddItem method (cannot do this at runtime)
- Removed Remove method (cannot do this at runtime)
- Fixed errors when UpdateButtons was used at runtime

5Hlp\_Revisions

6Revisions

7Revisions

8Browse:0015

## #<sup>9</sup>\$<sup>10</sup>K<sup>11+12</sup>License

axCool is developed and published by Software Solution. You may use it freely for development with Microsoft Visual Basic 5.0. This product is freeware and includes source code which you may change to suit your purposes.

9Hlp\_License

10License

11License

12Browse:0020

## #<sup>13</sup>\$<sup>14</sup>K<sup>15</sup>+<sup>16</sup>Tech Support

If you have any problems installing or running axCool, please feel free to contact our technical support department at one of the following:

Internet:           kirkq@execpc.com  
Telephone:        414-251-0915  
Snail Mail:                N92W17053 Roger Ave  
                              Menomonee Falls, WI 53051

HEY! Check out our world wide web page at:

[HTTP://www.execpc.com/~kirkq](http://www.execpc.com/~kirkq)

13Hlp\_Tech\_Support  
14Tech Support  
15Tech Support  
16Browse:0025

## #<sup>17</sup>\$<sup>18</sup>K<sup>19</sup>+<sup>20</sup> **Properties**

All of the properties for this control are listed below:

### Standard

Align  
DragItem  
DragMode  
Enabled  
Height  
Index  
Left  
Name  
Tag  
ToolTipText  
Top  
Visible  
Width

### **Control Specific**

BorderStyle  
Count  
ShowFlatGrey  
TextLabels

17Hlp\_Properties

18Properties

19Properties

20Browse:0030

## #<sup>21</sup>\$<sup>22</sup>K<sup>23</sup>+<sup>24</sup> **Methods**

All of the methods for this control are listed below:

### Standard

Drag

Move

ShowWhatsThis

ZOrder

### **Control Specific**

Item

ShowAboutBox

UpdateButtons

21Hlp\_Methods

22Methods

23Methods

24Browse:0035

## #<sup>25</sup>\$<sup>26</sup>K<sup>27</sup>+<sup>28</sup> **Events**

All of the events for this control are listed below:

### Standard

DragDrop

DragOver

### **Control Specific**

Click

DropDownClick

25Hlp\_Events

26Events

27Events

28Browse:0040



#<sup>33</sup>\$<sup>34</sup>+<sup>35</sup> Depending on your host environment, this property/method/event may be referred to by a different name or may not apply to this control. Refer to your host environments documentation or help file for further information.

33Hlp\_Standard  
34Standard  
35Browse:0050

# #<sup>36</sup>\$<sup>37</sup>K<sup>38</sup>+<sup>39</sup>BorderStyle Property

Returns or sets the borderstyle for the control

## Syntax:

*object*.BorderStyle [= integer]

The property syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a control object
<i>integer</i>	An integer expression that evaluates to one of the items in the list below

## Settings:

<b><u>Constant</u></b>	<b><u>Value</u></b>	<b><u>Description</u></b>
No Border	0	No border
Single	1	Single line border
Thin Raised	2	Thin raised border
Thick Raised	3	Thick raised border
Thin Inset	4	Thin inset border
Thick Inset	5	Thick inset border
Etched	6	Etched single line border
Bump	7	Raised single line border

36Hlp\_BorderStyle

37BorderStyle

38BorderStyle

39Browse:0055

# #<sup>40</sup>\$<sup>41</sup>K<sup>42</sup>+<sup>43</sup> **Count Property**

Returns the number of buttons in the button collection

## **Syntax:**

*object*.**Count**

The property syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a control object

## **Remarks**

# #<sup>44</sup>\$<sup>45</sup>K<sup>46</sup>+<sup>47</sup> **ShowFlatGrey Property**

Get/Sets a value to determine whether or not to display the picture in greyscale when the mouse is not over the button (Cool Button only)

## **Syntax:**

*object*.**ShowFlatGrey** [= boolean]

The property syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a control object
<i>boolean</i>	A boolean expression that evaluates to true/false

## **Remarks**

## #<sup>48</sup>\$<sup>49</sup>K<sup>50</sup>+<sup>51</sup> **TextLabels Property**

Sets/gets a value to determine whether buttons will show the text labels

### **Syntax:**

*object*.**TextLabels** [= boolean]

The property syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a control object
<i>boolean</i>	A boolean expression that evaluates to true/false

### **Remarks**

48Hlp\_TextLabels

49TextLabels

50TextLabels

51Browse:0070

# #<sup>52</sup>\$<sup>53</sup>K<sup>54</sup>+<sup>55</sup> Click Event

Occurs when the button is pushed

## Syntax:

*object*.Click(**Index**)

The event syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a axGrid object
<i>index</i>	An integer expression that evaluates to the index of the button that was pushed

## Remarks:

52Hlp\_Click

53Click

54Click

55Browse:0075

# #<sup>56</sup>\$<sup>57</sup>K<sup>58</sup>+<sup>59</sup> **DropDownClick Event**

Occurs when a dropdown button is clicked

## **Syntax:**

*object*.**DropDownClick**

The event syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a axGrid object
<i>index</i>	An integer expression that evaluates to the index of the dropdown button that was pushed

## **Remarks:**

## #60\$61K62+63 **Item Method**

This can be used by the programmer to set/get information on a specific button, but most of the time the programmer can just use the Custom property sheet button to edit the buttons at design time.

### **Syntax:**

`object.Item(Key)`

The method syntax has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
object	An object expression that evaluates to a control object
Key	A variant expression that evaluates to an integer representing the index of the button or a string indicating the key of the button

### **Information:**

This item collection contains the following fields:

<b><u>Name</u></b>	<b><u>Description</u></b>
Bitmap	Object: Picture to display on button
Caption	String: Caption of button
Enabled	Boolean: Determines if button is enabled or not Default=1
Style	Integer: Determines style of button as listed below Default=0
Tag	String: Tag of button
ToolTipText	String: ToolTip text to display when mouse is over button
Visible	Boolean: Determines if button is visible Default=1
Left	Integer: Returns the distance between the internal left edge of an object and the left edge of its container <i>This is autocalculated at design time</i>

### **Constants**

<b><u>Constant</u></b>	<b><u>Value</u></b>	<b><u>Description</u></b>
Cool Button	0	Flat button that highlights upon mouseover
Toolbar Button	1	Button with thin border
Seperator	2	Toolbar seperator (single raised line)
Toolbar Handle	3	Toolbar handle (double raised line)
Standard Button	4	Standard button similar to intrinsic button

### **Example:**

```
axCool.Item(1).Caption="New"  
set axCool.Item(1).Bitmap=loadpicture("new.bmp")  
axCool.Item(1).ToolTipText="New"  
axCool.UpdateButtons
```

60Hlp\_Item

61Item

62Item

63Browse:0090

# #<sup>64</sup>\$<sup>65</sup>K<sup>66</sup>+<sup>67</sup> ShowAboutBox Method

Show the about box

## Syntax:

*object*.ShowAboutBox

The method syntax has these parts:

<u>Part</u>	<u>Description</u>
object	An object expression that evaluates to a control object

## Example:

64Hlp\_ShowAboutBox  
65ShowAboutBox  
66ShowAboutBox  
67Browse:0100

## #<sup>68</sup>\$<sup>69</sup>K<sup>70</sup>+<sup>71</sup> UpdateButtons Method

Redraws all the buttons from the button collection. The user can utilize this method if it is necessary to change something about a button from code.

### Syntax:

*object*.UpdateButtons

The method syntax has these parts:

<u>Part</u>	<u>Description</u>
object	An object expression that evaluates to a control object

### Example:

```
axCool.Item(1).Caption="New caption"  
axCool.UpdateButtons
```

