

# **Slider Custom Controls**

SRC Enterprises

Steven R Clabaugh  
550 N. Pantano Road #286  
Tucson, Arizona 85710

(520) 296-7570  
Voice or FAX

## Slider Concepts

Slider controls provide an alternative to using the standard Windows scroll bar control for numeric value control. The appearance of a Slider control is completely defined by the programmer with bitmaps. A Slider control uses two bitmaps: one bitmap for the Slider face, a second bitmap for the Slider knob.

Several features are available for the Slider window and are set at Slider creation time and some features can be controlled on the fly with program code in an application program.

- A Slider can be defined as vertical or horizontal with the Slider style settings. (Creation time only)
- A Slider's value range can be reversed with the window style settings. (At creation time or on the fly, more on this later)
- A Slider's mouse cursor can be selected to be the standard Windows arrow or to a hand cursor. (At creation time or on the fly)
- A Slider's mouse cursor can be enabled or disabled. (At creation time or on the fly)
- With the proper set up, a Slider window can simulate a linear analog gauge. (Creation time only)
- When using a Slider as an analog linear gauge, the gauge can be defined as needle or thermometer style. (Creation time only)
- The Slider can be added as a custom control to the Windows SDK Dialog Editor.
- C/C++ language API programming support, including a C++ Slider Class.
- Support for Visual Basic Extention (VBX).

## Slider Operation

Slider operation is supported by the mouse only. Pressing the left mouse button while the mouse pointer is on the Slider knob, grabs the knob and the mouse can be used to move the Slider knob along the Slider. Mouse movement is restricted to within the Slider control as long as the left button is held down. Releasing the mouse button, releases the Slider knob. Clicking the left mouse button within the Slider control on either side of the Slider knob will increment and decrement the Slider value and reposition the knob accordingly.

## Slider Range and Value

From a programming stand-point, the Slider has been designed to be visually ergonomic. This means that the low value is at the bottom on vertical Sliders and at the left side on horizontal Sliders. As mentioned above, one of the features allows the Slider to be reversed. If the Slider is set reversed, the low value is at the top on vertical Sliders and at the right side on horizontal Sliders. The initial Slider range is set in parameters in the **SRCENTSLCreate()** function and reset at any time with the **SRCENTSLSetRange()** function. The value reflected by the Slider knob depends on the Slider range.

## Slider Message

When the user clicks within the Slider control or moves the Slider knob, the Slider control sends a **WM\_COMMAND** message to its parent window message processing function. The **wParam** parameter contains the ID value of the Slider control. The **lParam** low-order word contains the Slider window handle. The **lParam** high-order word is not used and contains a value of 0(Zero).

## Slider Files

The required files for incorporating Slider controls in an application are:

**SRCENTSL.DLL** This file contains the Slider window message processing function for Slider controls along with the API functions and the interface to the SDK Dialog Editor and the Visual Basic Extension (VBX) support.

To use the Slider Custom Control as a Visual Basic Extension(VBX), you need only to copy or rename this file to **SRCENTSL.VBX** to use the Slider in the Visual Basic and Visual C++ MFC environments.

**SRCENTSL.H** This file should be included in the application C(PP) source code file. It contains the Slider style and API function definitions and the declaration for a Slider class for C++ programming.

**SRCENTSL.LIB** Library of import functions created with **IMPLIB** for Slider controls. Specify this library in the **LINK** command's *lib-files* parameter when linking the application program.

## C++ Slider Class

For C++ programmers, the SRCENTSL.DLL file also contains implementation for a C++ class for Slider controls. The Slider class declaration is in the SCRENTSL.H file. Since the class implementation is simply an encapsulation of the Slider API functions, all parameters and return values for the Slider class member functions are the same as the API functions, but with the omission of the API's first parameter, which is the slider window handle parameter (*HWND hwnd*). The Slider class also includes a member function to retrieve the Slider window handle and the overloaded operators for: +=, -=, ++, and --.

## Creating Slider Controls with C/C++

Slider controls are child windows. They are initially created by one of two ways: Using the **SRCEntSLCreate()** API function in an application C(PP) source code file or with the dialog **CONTROL** statement in an application's resource script file.

When using the **CONTROL** statement in a resource file, keep in mind the Slider location specified is in dialog box base units, ie: relative to the size of the font in use for the dialog box. This does not enable the the Slider control to be located with pixel precision. If pixel precision location is required in a dialog box you will have to use the **SRCEntSLCreate()** function in the **WM\_INITDIALOG** message of the dialog.

An example of a **CONTROL** statement in a resource script file to create a horizontal Slider might look like this:

```
CONTROL "", ID_MYSLIDER, "SRCEnt_Slider",  
        SL_HORZ | WS_CHILD | WS_VISIBLE, 10,10,0,0
```

As in the example above, the Slider size should be specified as 0,0. This is because a Slider's window size is determined by the size of face bitmap used. Also, when using a **CONTROL** statement to create a Slider, it **must** be initialized with the **SRCEntSLInit()** function in the **WM\_INITDIALOG** message.

# Slider C/C++ API Functions

## Slider API Function Summary

Function	Purpose
SRCEntSLCreate	Creates a Slider control and initializes its settings.
SRCEntSLEnableMouse	Enable/Disable the mouse for a Slider window.
SRCEntSLGetRange	Retrieve the minimum and maximum values of a Slider.
SRCEntSLGetRevVal	Retrieve the current reverse value of a Slider.
SRCEntSLInit	Initialize a Slider.
SRCEntSLGetVal	Retrieve the current value of a Slider.
SRCEntSLSetFace	Sets the bitmap used for a Slider face.
SRCEntSLSetHand	Sets a Slider window mouse cursor.
SRCEntSLSetKnob	Sets the bitmap used for a Slider knob.
SRCEntSLSetRange	Sets the minimum and maximum values of a Slider.
SRCEntSLSetVal	Sets the value of a Slider.

## Slider API Function Detail

### SRCEntSLCreate

---

**Purpose** Creates a Slider control and initializes its settings.

**Syntax** `HWND SRCEntSLCreate(DWORD dwStyle, LPSTR FaceName, LPSTR KnobName, int X, int Y, int ks, int kt, int kb, int rmin, int rmax, int val, BOOL bHand, BOOL bMouse, HWND hWndParent, HMENU IDValue, HINSTANCE hInstance);`

**Description** Creates a Slider control and initializes its settings.

Note that there is no Slider sizing parameters because a Slider's window size is determined by the size of the face bitmap specified in the *hFace* parameter.

**Returns** HWND: A handle to the Slider control created.

#### Parameters

*dwStyle* DWORD: This determines the style for the Slider being created. The Slider styles are defined in SRCENTSL.H.

Specify NULL for the default non-reversed, vertical Slider.

SL\_HORZ Defines a horizontal Slider.  
SL\_REVERSED Defines a reversed Slider.  
SL\_THERM Defines thermometer style gauge.

When using a Slider as an analog gauge with the SL\_THERM (thermometer style) it is suggested the knob bitmap be defined as a single pixel height for vertical Sliders and a single pixel width for horizontal Sliders. This will ensure proper thermometer simulation.

The WS\_CHILD and WS\_VISIBLE window styles are specified in the function and need not be specified here.

Example: This specifies a reversed horizontal Slider.

```
SL_HORZ | SL_REVERSED
```

*FaceName* LPSTR: The resource name of the bitmap used for the Slider face.

*KnobName* LPSTR: The resource name of the bitmap used for the Slider knob.

<i>X</i>	<code>int</code> : The horizontal position of the upper left corner of the Slider control within the parent window client area.
<i>Y</i>	<code>int</code> : The vertical position of the upper left corner of the Slider control within the parent window client area.
<i>ks</i>	<code>int</code> : This value is used to position the knob within the Slider window. For vertical Sliders, this is the X position in the Slider window that the left edge of the knob slides along. For horizontal Sliders, this is the Y position in the Slider window that the top edge of the knob slides along.
<i>kt</i>	<code>int</code> : For vertical Sliders, this is the highest Y position in the Slider window the knob can go. For horizontal Sliders, this is the furthest left X position in the Slider window the knob can go.
<i>kb</i>	<code>int</code> : For vertical Sliders, this is the lowest Y position in the Slider window the knob can go. For horizontal Sliders, this is the furthest right X position in the Slider window the knob can go.
<i>rmin</i>	<code>int</code> : The Slider minimum range value.
<i>rmax</i>	<code>int</code> : The Slider maximum range value.
<i>val</i>	<code>int</code> : The Slider initial value.
<i>bHand</i>	<code>BOOL</code> : Specifies the mouse cursor used within the Slider control. Set to <code>TRUE</code> to use the hand mouse cursor. Set to <code>FALSE</code> to use the arrow mouse cursor.
<i>bMouse</i>	<code>BOOL</code> : Specifies whether or not the mouse is enabled for the Slider control. Set to <code>TRUE</code> to enable the mouse. Set to <code>FALSE</code> to disable the mouse.  Note: When using a slider as an analog linear gauge, this would normally be set to <code>FALSE</code> .
<i>hWndParent</i>	<code>HWND</code> : A handle to the Slider control's parent window.
<i>hMenu</i>	<code>HMENU</code> : The Slider control's ID value.
<i>hInstance</i>	<code>HINSTANCE</code> : The instance handle for the program module creating the Slider control.

---

## SRCEntSLEnableMouse

---

<b>Purpose</b>	Enables or disables the mouse for a Slider control.
<b>Syntax</b>	<code>VOID SRCEntSLEnableMouse (HWND <i>hWnd</i>, BOOL <i>bMouse</i>);</code>
<b>Returns</b>	VOID. No returned value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle
<i>bMouse</i>	BOOL: Set to TRUE to enable the mouse. Set to FALSE to disable the mouse.

## SRCEntSLGetRange

---

<b>Purpose</b>	Retrieve the current minimum and maximum range values of a Slider.
<b>Syntax</b>	<code>VOID SRCEntSLGetRange (HWND <i>hWnd</i>, LPINT <i>lpMinVal</i>, LPINT <i>lpMaxVal</i>);</code>
<b>Returns</b>	VOID: No returned value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle.
<i>lpMinVal</i>	LPINT: A pointer to the integer variable that will receive the minimum value of the Slider range.
<i>lpMaxVal</i>	LPINT: A pointer to the integer variable that will receive the maximum value of the Slider range.

## SRCEntSLGetRevVal

---

<b>Purpose</b>	Retrieve the current reverse value of a Slider.
<b>Syntax</b>	<code>int SRCEntSLGetRevVal (HWND <i>hWnd</i>);</code>
<b>Description</b>	<p>Using this function to retrieve the value value of a Slider knob gives the appearance of reversing the top and bottom of the Slider control.</p> <p>If the Slider is set to reversed style, the minimum value returned is at the top of vertical Sliders and and at the right side of horizontal Slider. If the reverse is set to TRUE, the minimum value returned is at the bottom of vertical Sliders and at the left side of horizontal Sliders.</p>
<b>Returns</b>	<code>int</code> : The Slider reversed value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle.

## SRCEntSLGetVal

---

<b>Purpose</b>	Retrieve the current value of a Slider.
<b>Syntax</b>	<code>int SRCEntSLGetVal (HWND <i>hWnd</i>);</code>
<b>Description</b>	<p>Retreive the Slider value.</p> <p>If the Slider is set to reversed style, the minimum value returned is at the bottom of vertical Sliders and at the left side of horizontal Sliders. If the reverse is set to TRUE, the minimum value returned is at the top of vertical Sliders and at the right side of horizontal Sliders.</p>
<b>Returns</b>	<code>int</code> : The Slider value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle.

## SRCEntSLInit

---

<b>Purpose</b>	Initializes the Slider control settings.
<b>Syntax</b>	<pre>VOID <b>SRCEntSLInit</b>(HWND <i>hWnd</i>, LPSTR <i>FaceName</i>, LPSTR <i>KnobName</i>, int <i>ks</i>, int <i>kt</i>, int <i>kb</i>, int <i>rmin</i>, int <i>rmax</i>, int <i>val</i>, BOOL <i>bHand</i>, BOOL <i>bMouse</i>);</pre>
<b>Description</b>	This function will initialize a Slider's settings. A call to this function is normally only used ( <b>and must be used</b> ) in a dialog's WM_INITDIALOG message handler to initialize a Slider created with a CONTROL statement in a resource file.
<b>Returns</b>	VOID: No returned value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle.
<i>FaceName</i>	LPSTR: The resource name of the bitmap used for the Slider face.
<i>KnobName</i>	LPSTR: The resource name of the bitmap used for the Slider knob.
<i>ks</i>	int: This value is used to position the knob within the Slider window. For vertical Sliders, this is the X position in the Slider window that the left edge of the knob slides along. For horizontal Sliders, this is the Y position in the Slider window that the top edge of the knob slides along.
<i>kt</i>	int: For vertical Sliders, this is the highest Y position in the Slider window the knob can go. For horizontal Sliders, this is the furthest left X position in the Slider window the knob can go.
<i>kb</i>	int: For vertical Sliders, this is the lowest Y position in the Slider window the knob can go. For horizontal Sliders, this is the furthest right X position in the Slider window the knob can go.
<i>rmin</i>	int: The Slider minimum range value.
<i>rmax</i>	int: The Slider maximum range value.
<i>val</i>	int: The Slider initial value.
<i>bHand</i>	BOOL: Specifies the mouse cursor used within the Slider control. Set to TRUE to use the hand mouse cursor. Set to FALSE to use the



## SRCEntSLSetKnob

---

<b>Purpose</b>	Sets the bitmap used for a Slider knob.
<b>Syntax</b>	<pre>VOID      <b>SRCEntSLSetKnob</b> (HWND      <i>hWnd</i>,      LPSTR <i>KnobName</i>,  int  <i>ks</i>,  int  <i>kt</i>,  int  <i>kb</i>,  BOOL <i>bRedraw</i>);</pre>
<b>Returns</b>	VOID. No returned value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle
<i>KnobName</i>	LPSTR: The resource name of the bitmap used for the Slider knob.
<i>ks</i>	int: For vertical Sliders, this is the X position in the Slider control that the left edge of the knob slides along. For horizontal Sliders, this is the Y position in the Slider control that the top edge of the knob slides along.
<i>kt</i>	int: For vertical Sliders, this is the Y position in the Slider control indicating the highest the knob can go. For horizontal Sliders, this is the X position in the Slider control indicating the furthest right position the knob can go.
<i>kb</i>	int: For vertical Sliders, this is the Y position in the Slider control indicating the lowest the knob can go. For horizontal Sliders, this is the X position in the Slider control indicating the furthest left position the knob can go.
<i>bRedraw</i>	BOOL: TRUE if the new knob should be redrawn , FALSE if not.

## SRCEntSLSetRange

---

<b>Purpose</b>	Sets the upper and lower ranges of a Slider.
<b>Syntax</b>	<code>VOID <b>SRCEntSLSetRange</b>(HWND <i>hWnd</i>, int <i>nMin</i>, int <i>nMax</i>, BOOL <i>bRedraw</i>);</code>
<b>Description</b>	Use this function to set the Slider control range.
<b>Returns</b>	VOID: No returned value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle.
<i>nMin</i>	int: The Slider lower limit value.
<i>nMax</i>	int: The Slider upper limit value.
<i>bRedraw</i>	BOOL: TRUE if the knob should be redrawn to show the new position, FALSE if not.

## SRCEntSLSetVal

---

<b>Purpose</b>	Sets the value of a Slider.
<b>Syntax</b>	<code>int <b>SRCEntSLSetVal</b>(HWND <i>hWnd</i>, int <i>nVal</i>, BOOL <i>bRedraw</i>);</code>
<b>Returns</b>	int: The previous value.
<b>Parameters</b>	
<i>hWnd</i>	HWND: The Slider control handle.
<i>nVal</i>	int: The new value.
<i>bRedraw</i>	BOOL: TRUE if the knob is to be re-drawn at the new position, FALSE if not.

## Slider Visual Basic Extention (VBX)

For Visual Basic and Visual C++ programmers using MFC, the Slider Custom Control can be used as a Visual Basic Extention by copying or renaming the SRCENTSL.DLL file to SRCENTSL.VBX. The Slider can then be added to the Visual Basic design mode control toolbox.

### Slider VBX Properties

Enable_Mouse	<p>Enables or disables the mouse for a Slider control. Set to TRUE to enable the mouse. Set to FALSE to disable the mouse.</p> <p>When using a slider as an analog linear gauge, this would normally be set to FALSE.</p> <p>This is a design-time and run-time controllable property.</p>
Face_Bitmap	<p>Sets the bitmap used for a Slider face.</p> <p>This is a design-time only property.</p>
Mouse_Pointer	<p>With this property, either the arrow or the hand mouse pointer can be selected.</p> <p>This is a design-time and run-time controllable property.</p>
Orientation	<p>With this property you select whether the Slider operates vertically or horizontally.</p> <p>This is a design-time only property.</p>
Knob_Bitmap	<p>Sets the bitmap used for a Slider knob.</p> <p>This is a design-time only property.</p>

Knob_Top_Position	<p>For vertical Sliders, this is the highest Y position in the Slider face the knob can go. For horizontal Sliders, this is the furthest right X position in the Slider face the knob can go.</p> <p>This is a design-time only property.</p>
Knob_Bottom_Position	<p>For vertical Sliders, this is the lowest Y position in the Slider face the knob can go. For horizontal Sliders, this is the furthest left X position in the Slider face the knob can go.</p> <p>This is a design-time only property.</p>
Knob_Slide_Position	<p>This property is used to position the knob within the Slider face. For vertical Sliders, this is the X position in the Slider face that the left edge of the knob slides along. For horizontal Sliders, this is the Y position in the Slider face that the top edge of the knob slides along.</p> <p>This is a design-time only property.</p>
Left	<p>This is the standard Left property that specifies the horizontal location of the upper left corner of the Slider control within the form.</p>
Max	<p>The Slider maximum range value.</p> <p>This is a design-time and run-time controllable property.</p>
Min	<p>The Slider minimum range value.</p> <p>This is a design-time and run-time controllable property.</p>
Name	<p>This is the standard Name property that specifies the name of the Slider. Use this name when referencing the Slider in program code.</p> <p>The default name is SRCEntSlider1.</p>
Reversed	<p>Defines a reversed Slider. Set to FALSE to define a NON-reversed Slider. Set to TRUE to define a reversed Slider.</p> <p>This is a design-time only property.</p>

Thermometer	<p>When using the Slider as an analog linear gauge, set this to FALSE to use a needle style value indicator. Set to TRUE to use a thermometer style value indicator.</p> <p>When using a Slider as an analog gauge with the Thermometer style set to TRUE, it is suggested the knob bitmap be defined as a single pixel height for vertical Sliders and a single pixel width for horizontal Sliders. This will ensure proper thermometer simulation.</p> <p>This is a design-time only property.</p>
Top	<p>This is the standard Top property that specifies the vertical location of the upper left corner of the Slider control within the form.</p>
Value	<p>The Slider initial value. (16 bit signed integer)</p> <p>This is a design-time and run-time controllable property.</p>

## Slider VBX Event

The Slider control has only one event called **NewVal**. When the user clicks within the Slider control or moves the Slider knob, the Slider control fires the **NewVal** event. It is in this **NewVal** event that your program code should reference the Value property of the Slider and assign the Value to other variables or controls depending on what your application program is designed to do.