ThinVNC Software Development Kit

Developer's guide

Table of Contents

1 Introduction

ThinVNC SDK is a component library that allows you to add Desktop and Application Sharing capabilities to your Windows Application. ThinVNC SDK is suitable for any ActiveX compatible development environment and frameworks, including .NET/Mono framework.

Why ThinVNC SDK?

1. Enables you to share the windows applications, making them accessible both from your own Windows or a Web Client (browser).

2. Possible uses include custom remote support procedures, peer-to-peer application collaboration, authorization procedures, etc.

3. ThinVNC SDK also allows you to publish "virtual" windows by sharing animated bitmaps. A sample code is included that allows you to publish Macromedia Flash movies over the web. With this example, a Macromedia Flash movie can be visualized on any HTML5-capable Web Browser, including Apple Safari on the IPad/IPhone.

See more:

Architecture

Getting Started

JavaScript Reference

ActiveX Reference

2 Arquitecture

ThinVNC SDK contains the following components:

- **ActiveX Server Components**
- **ActiveX Client Control**
- **Javascript Client Library**

2.1 ActiveX Server Components

A set of ActiveX Server components allows you to build a Server application, which will listen for HTTP requests, capture windows bitmaps and send them back to the client side.

There are two components: ThinVnc and ThinWebServer. ThinVnc contains all the capture logic while ThinWebServer is the HTTP server that will communicate with the client.

2.2 ActiveX Client Control

The ActiveX control ThinVncClient allows you to build a Client application, which will connect to a ThinVNC server (or a ThinVNC SDK-based server application) and display the remote desktop or windows.

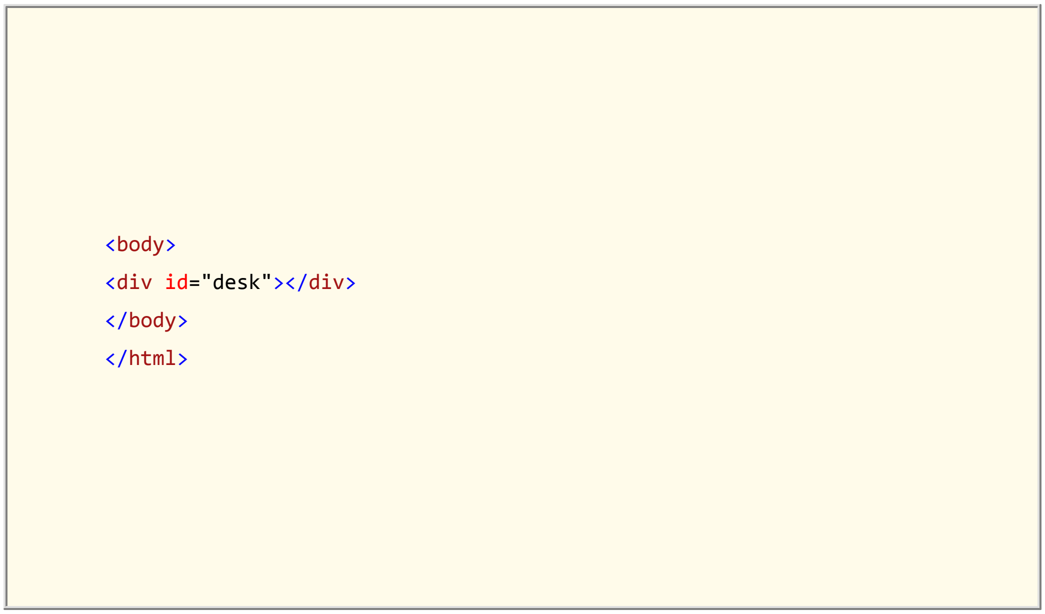
2.3 Javascript Client Library

On the client side, a div element and a Javascript object will be in charge of rendering the remote windows on a local Web Browser.
This is the basic code to embed ThinVNC within an HTML using jquery:

```
<script type="text/javascript">
    $(document).ready(function(){
        mythinvnc = new ThinVNC();
        mythinvnc.init({

            divId:'desk',
            scaled : true,
            mouseControl:true

        });
    });
</script>
```



```
<body>
<div id="desk"></div>
</body>
</html>
```

3 Getting Started

To get started, let's try building a couple of simple examples.

- **Creating a ThinVNC server**
- **Filtering Windows and Processes**
- **Creating a ThinVNC Client**

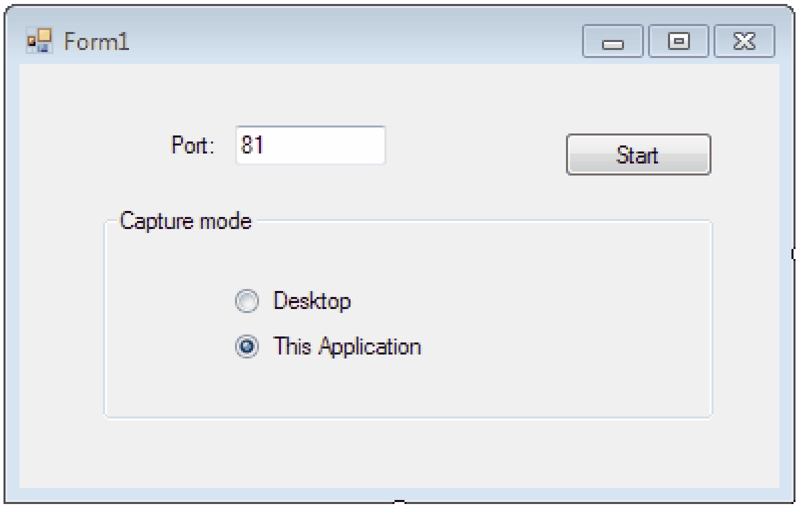
3.1 Creating a ThinVNC Server

Let's create a simple ThinVNC server using Microsoft Visual Studio:

1. Create a new project: Open Microsoft Visual Studio and start a new Windows Form Application project. Let's name it "ThinVncServer".

2. Add a reference to "ThinVNC Server" ActiveX library: On the Solution Explorer pane right-click on the ThinVncServer project node and select "Add Reference".

3. Add a few controls to the form:



Port: 81

Capture mode

○ Desktop

● This Application

4. Add code:

An initialization code.

```
      Public Class Form1

          ' Declares ThinVnc and ScreenScraper components
          Dim tvnc As New ThinVncX.ThinVnc
          Dim scraper As New ThinVncX.ThinScreenScraper
```

```
    Public Sub New()
        ' This call is required by the Windows Form
Designer.
        InitializeComponent()

        ' Instanciates a new ThinVnc and ScreenScraper
component
        tvnc = New ThinVncX.ThinVnc()
        scraper = New ThinVncX.ThinScreenScraper()

        ' Tells ThinVnc to use ScreenScraper
        tvnc.Scraper = scraper
    End Sub
```

The "Start" button code

```
Private Sub bStart_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles bStart.Click
    If Not tvnc.WebServer.HttpActive Then
       tvnc.WebServer.HttpPort = CInt(textPort.Text)
    End If
    tvnc.WebServer.HttpActive = Not
tvnc.WebServer.HttpActive
End Sub
```

5. Compile and run the project.

6. Try it out by typing a port number into the textbox and press the "Start" button. Most likely, Windows Firewall will ask you to set up an exception. Add the exception and then open a Web Browser (preferably in another PC) and type http://yourpc:port.

You should be able to see and control the PC desktop.

3.2 Filtering Windows and Processes

Continuing with the previous example, let's add code to the radio buttons placed on the form. They will allow us to choose to publish the entire desktop or just the application process windows.

Add a handler to radiobutton CheckedChanged event:

```
Private Sub rbDesktop_CheckedChanged(
ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rbDesktop.CheckedChanged, rbThisapp.CheckedChanged

   CaptureModeChanged()

End Sub

Private Sub CaptureModeChanged()

   If IsNothing(scraper) Then Exit Sub

   If rbDesktop.Checked Then
      scraper.CaptureMode = ThinVncX.AxCaptureMode.cmAllProcesses
   Else
      scraper.CaptureMode = ThinVncX.AxCaptureMode.cmThisProcess
   End If

End Sub
```

Now compile and run the project and try selecting "This Application". On the Web Browser page you should see just the application window.

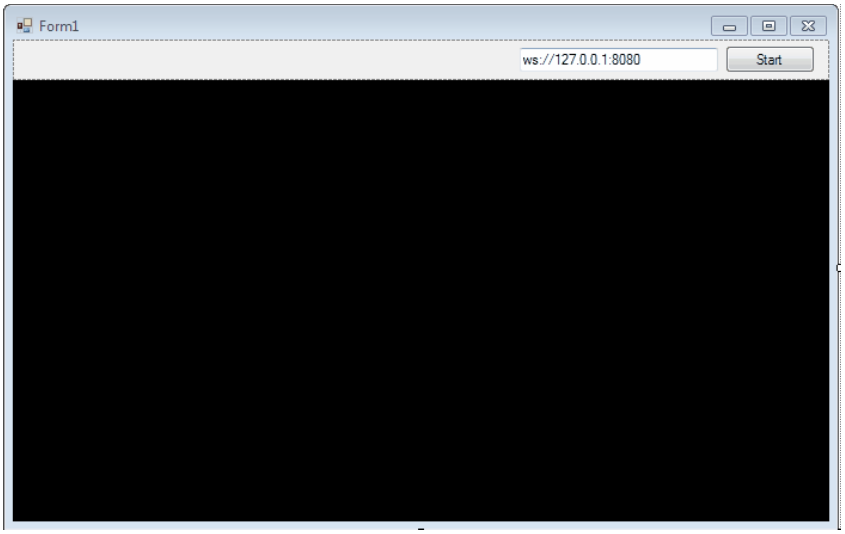
3.3 Creating a ThinVNC Client

Let's create a simple ThinVNC client using Microsoft Visual Studio.

1) Create a new project.
Open Microsoft Visual Studio and start a new Windows Form Application project. Let's name it "VncClient".

2) Add ThinVNC Client Control to the toolbox.
Open the toolbox, right-click on it and select "Choose Items". Select the "COM Components" tab and look for the ThinVNC Client Control. Select it and click OK.

3) Look for the ThinVNC Client Control in the toolbox, and drop it on the form. Add a textbox and a "Start" button as shown:



4) Add code to the "Start" button.

```
Private Sub bStart_Click(ByVal sender As System.Object, ByVal e As
System.         EventArgs) Handles bStart.Click
    AxThinVNCClient1.Connect(textURL.Text)
End Sub
```

5) Try it.

4 Javascript Reference

4.1 ThinVNC Javascript Object

The ThinVNC javascript class contains all the client-side logic for the Web. After it is create, it also needs to be initialized passing the id of the div element that will display the remote desktop, as follows:

```
<script type="text/javascript">

   $(document).ready(function(){
         mythinvnc = new ThinVNC();
         mythinvnc.init({
               divId:'desk',
               scaled : true,
               mouseControl:true
         });
        mythinvnc.connect();
</script>
```

Methods

- init
- connect
- disconnect
- refresh
- setGrayscale
- setKdbControl
- setMonitor
- setMouseControl
- setPixelFormat
- setScaled
- stop
- start

Events

Remarks

The PlatformHTML5 global variable holds a TPlatformHTML5 class object that is automatically instantiated every time your application is executed.

4.1.1 Methods

4.1.1.1 init

The init method initializes the thinVNC object and receive several optional arguments as follows:

Javascript Syntax

```
    thinvnc.init([
           divId:null,
           mouseControl: true,
           kbdControl: true,
           pixelFormat: 0,
           grayscale: false,
           scaled: true,
           monitor: 0,
    ])
```

Arguments

divId	Id of the div element that will display the remote desktop. Defaults to null.	string
mouseControl	This parameter allows the session to be started with or without remote mouse control. Defaults to true.	boolean
kbdControl	This parameter allows the session to be started with or without remote keyboard control. Defaults to true.	boolean
pixelFormat	This parameter controls the initial color depth. Valid values are 0 for 16-bit/32-bit (depending on server settings) color depth and 1 for 256-bit color depth. Default: 0.	int
grayscale	This parameter allows the session to be started in grayscale. Default: false.	boolean
scaled	This parameter controls whether the remote desktop view is stretched when it doesn't fit in the client area. Defaults to true.	boolean
monitor	This parameter controls which monitor will be visualized. Defaults to 0. Set to -1 to see all monitors at once.	integer

See also

See also the connect method.

4.1.1.2 connect

The connect method creates a connection with the server and positions the remote data on the specified div.

Javascript Syntax

```
thinvnc.connect();
```

See also

See also the disconnect method.

4.1.1.3 disconnect

The disconnect method terminates a remote session.

Javascript Syntax

```
thinvnc.disconnect();
```

See also

See also the disconnect method.

4.1.1.4 refresh

The refresh requests a fresh copy of the remote desktop content.

Javascript Syntax

```
thinvnc.refresh();
```

See also

See also the start and the stop methods.

4.1.1.5 setGrayscale

The setGrayscale method enables/disables the gray scaling according to the argument sent.

Javascript Syntax

```
thinvnc.setGrayscale(enabled);
```

Arguments

enabled	Configures whether the gray scale should be activated or not.	boolean

See also

See also the setKdbControl, setMonitor, setMouseControl, setScaled and setPixelFormat methods.

4.1.1.6 setKbdControl

The setKbdControl method enables/disables the remote keyboard control.

Javascript Syntax

```
thinvnc.setKbdControl(enabled);
```

Arguments

enabled	Configures whether the keyboard should be activated or not.	boolean

See also

See also the setGrayscale, setMonitor, setMouseControl, setScaled and setPixelFormat methods.

4.1.1.7 setMonitor

The setMonitor method changes the active remote monitor.

Javascript Syntax

```
thinvnc.setMonitor(monitor);
```

Arguments

monitor	Valid values are from -1 to n. Use -1 to see all monitors at once.	int

See also

See also the setGrayscale, setKdbControl, setMouseControl, setScaled and setPixelFormat methods.

4.1.1.8 setMouseControl

The setMouseControl Enables/disables remote mouse control.

Javascript Syntax

```
thinvnc.setMouseControl(enabled);
```

Arguments

enabled	Configures whether the remote mouse control should be activated or not.	boolean

See also

See also the setGrayscale, setKdbControl, setMonitor, setScaled and setPixelFormat methods.

4.1.1.9 setPixelFormat

The setPixelFormat changes the color depth of the remote connection.

Javascript Syntax

```
thinvnc.setPixelFormat(colorDepth);
```

Arguments

colorDepth	Valid values are 0 for 16-bit color depth and 1 for 256-bit color depth.	int

See also

See also the setGrayscale, setKdbControl, setMonitor, setScaled and setMouseControl methods.

4.1.1.10 setScaled

The setScaled changes enables/disables the remote connection scaling. When scaled, the remote desktop view is stretched when it doesn't fit in the client area.

Javascript Syntax

```
thinvnc.setPixelFormat(enabled);
```

Arguments

enabled	Configures whether the scaled mode should be activated or not.	boolean

See also

See also the setGrayscale, setKdbControl, setMonitor, setPixelFormat and setMouseControl methods.

4.1.1.11 stop

The stop method pauses an active session.

Javascript Syntax

```
thinvnc.stop();
```

See also

See also the start method.

4.1.1.12 start

The start method resumes a paused session.

Javascript Syntax

```
thinvnc.start();
```

See also

See also the stop method.

5 ActiveX Reference

5.1 Syntax conventions

The following text formats are used throughout the Components Reference that will follow this guide:

For each property and method

Italic text	Denotes an object.
Bold text	Denotes a property or a method name.
[= *value type*]	Denotes a value type to be assigned.
[*value type* =]	Denotes a return value type.
(*TimeOut***)**	Denotes a parameter to be passed to a method.
(*[TimeOut]***)**	Denotes an optional parameter to be passed to a method.

5.2 IThinAbstractScraper

IThinAbstractScraper is a base interface implemented by all scrapers. A scraper is a component that allows for capturing image data from a specific source. Currently there are tree scrapers implemented:

ThinScreenScraper
Allows for capturing desktop windows images. This is the standard scraper and is the one active in ThinVNC server. The capture is done on top window basis, allowing the filtering by top window handle or process id.

ThinBitmapScraper
This scraper allows you to publish images from bitmaps. You are responsible for updating the bitmap's content. The bitmap can be produced from a number of sources: like graphic report generation or a flash movie.

ThinWindowScraper
This scraper is much like the ThinScreenScraper, but it allows you to add non-top windows to be captured.

5.2.1 Properties

5.2.1.1 CapturePixelFormat

Sets/gets the color depth that ThinVNC will use to capture windows and bitmaps. The default value 16bit.

ActiveX/.NET Syntax

```
Scraper.CapturePixelFormat [= AxPixelFormat]
```

Delphi Syntax

```
Scraper.CapturePixelFormat [:= TPixelFormat]
```

See also

See also the Syntax conventions used on the definitions above.

5.2.1.2 ImageMethod

Sets/gets the image type that ThinVNC will generate when capture windows and bitmaps. The defaults is imMixed.

ActiveX/.NET Syntax

Scraper.**ImageMethod** [= ***AxImageMethod***]

Delphi Syntax

Scraper.**ImageMethod** [:= ***TImageMethod***]

See also

See also the Syntax conventions used on the definitions above.

5.2.1.3 JpgQuality

Sets/gets the JPEG compression level. The defaults value is 80 percent.

ActiveX/.NET Syntax

Scraper.**JpgQuality** [= ***int***]

Delphi Syntax

Scraper.**JpgQuality** [:= ***integer***]

See also

See also the Syntax conventions used on the definitions above.

5.2.1.4 FrameDelay

Allows you to specify a minimum time between captures. Defaults to 50 ms.

ActiveX/.NET Syntax

```
Scraper.FrameDelay [= int]
```

Delphi Syntax

```
Scraper.FrameDelay [:= integer]
```

See also

See also the Syntax conventions used on the definitions above.

5.2.1.5 SynchronizeCapture

Allows you to force the capture process to work synchronized with the main thread. The default is set to false;

ActiveX/.NET Syntax

```
Scraper.SynchronizeCapture [= bool]
```

Delphi Syntax

```
Scraper.SynchronizeCapture [:= boolean]
```

See also

See also the Syntax conventions used on the definitions above.

5.2.1.6 SynchronizeInput

Allows you to force input events to work synchronized with the main thread. The default is set to false.

ActiveX/.NET Syntax

```
Scraper.SynchronizeInput [= bool]
```

Delphi Syntax

```
Scraper.SynchronizeInput [:= boolean]
```

See also

See also the Syntax conventions used on the definitions above.

5.2.2 Enums

5.2.2.1 ImageMethod

Enum that specifies the image type generated by the server.

ActiveX/.NET Syntax

```
AxImageMethod = (imJpeg, imPng, imMixed)
```

See also

See also the Syntax conventions used on the definitions above.

5.2.2.2 PixelFormat

Enum that specifies the color depth.

ActiveX/.NET Syntax

```
AxPixelFormat = (pf4bit,pf8bit,pf15bit,pf16bit,pf24bit,pf32bit)
```

See also

See also the Syntax conventions used on the definitions above.

5.3 ThinBitmapScraper

This scraper allows you to publish images from bitmaps. You are responsible for updating the bitmaps content. The bitmap can be produced from a number of sources: like graphic report generation or flash movie.

5.3.1 Properties

5.3.1.1 Height

Sets/gets the capture area height.

ActiveX/.NET Syntax

*Scraper.***Height** [= ***int***]

Delphi Syntax

*Scraper.***Height** [:= ***integer***]

See also

See also the Syntax conventions used on the definitions above.

5.3.1.2 Width

Sets/gets the capture area width.

ActiveX/.NET Syntax

*Scraper.***Width** [= ***int***]

Delphi Syntax

*Scraper.***Width** [:= ***integer***]

See also

See also the Syntax conventions used on the definitions above.

5.3.2 Methods

5.3.2.1 AddBitmap

Let's you add a single bitmap as a capture source. The bitmap is requested each time the scraper needs a new fresh copy of it.

ActiveX/.NET Syntax

```
Scraper.AddBitmap (IBitmapCallback callback)
```

Delphi Syntax

```
Scraper.AddBitmap (callback: IBitmapCallback)
```

See also

See also the Syntax conventions used on the definitions above.

5.3.2.2 BitmapChanged

Informs the scraper when the bitmap content has changed.

ActiveX/.NET Syntax

```
Scraper.BitmapChanged()
```

Delphi Syntax

```
Scraper.BitmapChanged
```

See also

See also the Syntax conventions used on the definitions above.

5.3.3 Events

5.3.3.1 OnFunctionKey

This event occurs when a keyboard sequence corresponding to a function key arrives from the client.

ActiveX/.NET Syntax

```
void OnFunctionKey (string func, BOOL* handled)
```

Delphi Syntax

```
procedure OnFunctionKey (sender:TObject; func: string; var handled:boolean)
```

Remarks

Recognized function keys are:

'CtrlAltDel'
'AltTab'
'AltShiftTab'
'ShiftCtrlEsc'
'CtrlEsc'
'AltEsc'
'LeftWin'
'RightWin'
'PrtScr'
'AltPrtScr'

See also

See also the OnKeyboardInput and OnMouseInput events.

5.3.3.2 OnKeyboardInput

Occurs when a keyboard event is received from the client.

ActiveX/.NET Syntax

```
void OnKeyboardInput (short key, BOOL down, BOOL* handled)
```

Delphi Syntax

```
procedure OnKeyboardInput (sender:TObject; key: word; down:boolean;
var handled:boolean)
```

See also

See also the OnFunctionKey and OnMouseInput events.

5.3.3.3 OnMouseInput

Occurs when a mouse event is received from the client.

ActiveX/.NET Syntax

```
void OnMouseInput (int x, int y, AxMouseButton button, AxMouseAction action,
     BOOL* handled)
```

Delphi Syntax

```
procedure OnMouseInput (sender:TObject; x,y: integer; button:TMouseButton;
            action:TMouseAction; var handled:boolean)
```

See also

See also the OnFunctionKey and OnKeyboardInput events.

5.3.4 Interfaces

5.3.4.1 IThinBitmapScraperCallback

The IThinBitmapScraperCallback interface must be implemented by the bitmap source to provide a fresh bitmap image when it is required by the scraper.

ActiveX/.NET Syntax

```
void (int*hbmp, int* left, int* top, int* right, int* bottom, bool* delbmp);
```

Delphi Syntax

```
procedure (var hbmp: integer; var left,top,right,bottom: integer; delbmp:wordbool);
```

Parameters

hbmp	Bitmap handle (HBITMAP)	
left	Left bitmap coordinates	
top	Left bitmap coordinates	
right	Left bitmap coordinates	
bottom	Left bitmap coordinates	
delbmp	Indicates whether the calling object must delete the bitmap handle.	

See also

See also the OnFunctionKey and OnKeyboardInput events.

5.4 ThinScreenScraper

Allows for capturing desktop windows images. This is the standard scraper and is the one that's active in ThinVNC server. Capture is done on top-window basis, allowing the filtering by top-window handle or process id.

5.4.1 Properties

5.4.1.1 ExcludeWindows

Enables/disables windows exclusion/inclusion. When ExcludeWindows is true, windows added using AddTopWindow are excluded from the windows sent to the client, otherwise they are included. The defaults is set to true.

ActiveX/.NET Syntax

```
Scraper.ExcludeWindows [= bool]
```

Delphi Syntax

```
Scraper.ExcludeWindows [:= boolean]
```

See also

See also the ExcludeProcess events.

ActiveX/.NET Syntax

Delphi Syntax

5.4.1.2 ExcludeProcesses

Enables/disables process exclusion/inclusion. When ExcludeProcesses is true, all the windows belonging to the process added using AddProcessId are excluded from the windows sent to the client, otherwise they are included. The defaults is set to true.

ActiveX/.NET Syntax

> *Scraper.***ExcludeProcesses** [= ***bool***]

Delphi Syntax

> *Scraper.***ExcludeProcesses** [:= ***boolean***]

See also

See also the ExcludeWindows events.

5.4.2 Methods

5.4.2.1 AddProcessId

Allows you to add a process id to the process filter list. See ExcludeProcesses.

ActiveX/.NET Syntax

*Scraper.***AddProcessId (int pid)**

Delphi Syntax

*Scraper.***AddProcessId (pid:cardinal)**

5.4.2.2 DelProcessId

Allows you to remove a process id from the process filter list. See ExcludeProcesses.

ActiveX/.NET Syntax

*Scraper.***DelProcessId (int pid)**

Delphi Syntax

*Scraper.***DelProcessId (pid:cardinal)**

5.4.2.3 AddTopWindow

Allows you to add a windows handle to the windows filter list. See ExcludeWindows.

ActiveX/.NET Syntax

*Scraper.***AddTopWindow (HWND wnd)**

Delphi Syntax

*Scraper.***AddTopWindow (wnd: HWND)**

5.4.2.4 DelTopWindow

Allows you to remove a windows handle from the windows filter list. See ExcludeWindows.

ActiveX/.NET Syntax

*Scraper.***DelTopWindow (HWND wnd)**

Delphi Syntax

*Scraper.***DelTopWindow (wnd: HWND)**

5.4.2.5 IsProcessInList

Allows you to check whether a process id is in process filter list.

ActiveX/.NET Syntax

[***bool*** =] *Scraper.***IsProcessInList (int pid)**

Delphi Syntax

[***boolean*** **:**=] *Scraper.***IsProcessInList (pid: cardinal)**

5.4.2.6 IsWindowInList

Allows you to check whether a windows handle is in the windows filter list.

ActiveX/.NET Syntax

[***bool*** =] *Scraper.***IsWindowInList (HWND wnd)**

Delphi Syntax

[***boolean*** **:**=] *Scraper.***IsWindowInList (wnd: HWND)**

5.5 ThinVNCClient

This activeX control allows you to include a ThinVNC Remote Desktop Client in your application.

5.5.1 Properties

5.5.1.1 Active

Returns/sets the remote control session state.

ActiveX/.NET Syntax

*VncClient.***Active** [= ***bool***]

Delphi Syntax

*VncClient.***Active** [:= ***boolean***]

5.5.1.2 Connected

Returns the connection state.

ActiveX/.NET Syntax

[***bool*** =] *VncClient.***Connected**

Delphi Syntax

[***boolean*** **:=**] *VncClient.***Connected**

5.5.1.3 Control

Activates/deactivates remote mouse and keyboard control.

ActiveX/.NET Syntax

*VncClient.***Control** [= ***bool***]

Delphi Syntax

*VncClient.***Control** [:= ***boolean***]

5.5.1.4 FullColor

Sets/gets the color depth. When false, 8-bit color depth is in place. When true, 16-bit or 32-bit color depth is in place, according to the server settings.

ActiveX/.NET Syntax

*VncClient.***FullColor** [= ***bool***]

Delphi Syntax

*VncClient.***FullColor** [:= ***boolean***]

5.5.1.5 Scaled

Enables/disables the scaling feature.

ActiveX/.NET Syntax

*VncClient.***Scaled** [= ***bool***]

Delphi Syntax

*VncClient.***Scaled** [:= ***boolean***]

5.5.1.6 Seamless

Enables/disables the seamless windows feature.

ActiveX/.NET Syntax

*VncClient.***Seamless** [= ***bool***]

Delphi Syntax

*VncClient.***Seamless** [:= ***boolean***]

5.5.1.7 GrayScale

Enables/disables the gray scale feature.

ActiveX/.NET Syntax

*VncClient.***GrayScale** [= ***bool***]

Delphi Syntax

*VncClient.***GrayScale** [:= ***boolean***]

5.5.1.8 ImageMethod

Sets/gets the image type that ThinVNC will generate when capture windows and bitmaps. Defaults to imMixed.

ActiveX/.NET Syntax

*VncClient.***ImageMethod** [= ***AxImageMethod***]

Delphi Syntax

*VncClient.***ImageMethod** [:= ***TImageMethod***]

5.5.1.9 Quality

Enables/disables the quality windows feature.

ActiveX/.NET Syntax

*VncClient.***Quality** [= ***int***]

Delphi Syntax

*VncClient.***Quality** [:= ***integer***]

5.5.1.10 UseProxy

Enable/disable the use of the proxy.

ActiveX/.NET Syntax

*VncClient.***UseProxy** [= ***bool***]

Delphi Syntax

*VncClient.***UseProxy** [:= ***boolean***]

5.5.1.11 ProxyHost

Proxy server IP or URL.

ActiveX/.NET Syntax

*VncClient.***ProxyHost** [= ***string***]

Delphi Syntax

*VncClient.***ProxyHost** [:= ***string***]

5.5.1.12 ProxyPort

Proxy server port.

ActiveX/.NET Syntax

*VncClient.***ProxyPort** [= ***int***]

Delphi Syntax

*VncClient.***ProxyPort** [:= ***integer***]

5.5.1.13 ProxyUsername

Username in case the proxy requires authentication.

ActiveX/.NET Syntax

*VncClient.***ProxyUsername** [= ***string***]

Delphi Syntax

*VncClient.***ProxyUsername** [:= ***string***]

5.5.1.14 ProxyPassword

Password in case the proxy requires authentication.

ActiveX/.NET Syntax

*VncClient.***ProxyPassword** [= ***string***]

Delphi Syntax

*VncClient.***ProxyPassword** [:= ***string***]

5.5.2 Methods

5.5.2.1 Connect

Initiates the connection to the remote ThinVNC server. Url parameter takes the format: ws://server:port for non-ssl connections and wss://server:port for ssl connections.

ActiveX/.NET Syntax

*VncClient.***Connect (string url)**

Delphi Syntax

*VncClient.***Connect (url: string)**

5.5.2.2 Disconnect

Disconnects the current connection with the remote ThinVNC server.

ActiveX/.NET Syntax

*VncClient.***Disconnect**

Delphi Syntax

*VncClient.***Disconnect**

5.5.2.3 Refresh

Refreshes the content of current remote desktop view.

ActiveX/.NET Syntax

*VncClient.***Refresh**

Delphi Syntax

*VncClient.***Refresh**

5.5.3 Events

5.5.3.1 OnConnect

Occurs when the connection with the remote ThinVNC server has been established.

ActiveX/.NET Syntax

void OnConnect ()

Delphi Syntax

procedure OnConnect (Sender:TObject)

5.5.3.2 OnConnectError

Occurs when the connection with the remote ThinVNC server couldn't be established.

ActiveX/.NET Syntax

void OnConnectError (string Err)

Delphi Syntax

procedure OnConnectError (Sender:TObject; const Err: WideString)

5.5.3.3 OnDisconnect

Occurs when the connection with the remote ThinVNC server has been terminated.

ActiveX/.NET Syntax

void OnDisconnect ()

Delphi Syntax

procedure OnDisconnect (Sender:TObject)

5.5.3.4 OnStatusChanged

Is fired upon connection or session status changes. Session status changes include changes in parameters that affect the remote control session: like mouse control, color depth, etc.

ActiveX/.NET Syntax

void OnStatusChanged ()

Delphi Syntax

procedure OnStatusChanged (Sender:TObject)

5.5.4 Enums

5.5.4.1 ImageMethod

Enum that specifies the image type do you want to receive by the client.

ActiveX/.NET Syntax

AxImageMethod = (imJpeg, imPng, imMixed)

Delphi Syntax

TImageMethod = (imJpeg, imPng, imMixed)

5.6 ThinVNC

This component is the main ThinVNC component needed to build a ThinVNC server.

5.6.1 Properties

5.6.1.1 SessionCount

Retireves the number of client sessions currently open.

ActiveX/.NET Syntax

[***int*** =] *ThinVnc*.**SessionCount**

Delphi Syntax

[***integer*** :=] *ThinVnc*.**SessionCount**

5.6.1.2 Scraper

Sets/gets the **Scraper** instance that will be used as a content source.

ActiveX/.NET Syntax

ThinVnc.**WebServer** [= ***IThinAbstractScraper***]

Delphi Syntax

ThinVnc.**WebServer** [:= ***TThinAbstractScraper***]

5.6.1.3 WebServer

Gets the **ThinWebServer** instance that will be used for communicating with the client.

ActiveX/.NET Syntax

[***IThinWebServer =***] *ThinVnc*.**WebServer**

Delphi Syntax

[***TThinWebServer :=***] *ThinVnc*.**WebServer**

5.6.2 Events

5.6.2.1 OnSessionStarted

Occurs when a new client session has started.

ActiveX/.NET Syntax

```
void OnSessionStarted ()
```

Delphi Syntax

```
procedure OnSessionStarted (sender:TObject)
```

5.6.2.2 OnSessionEnded

Occurs when a new client session has ended.

ActiveX/.NET Syntax

```
void OnSessionEnded ()
```

Delphi Syntax

```
procedure OnSessionEnded (sender:TObject)
```

5.7 ThinWebServer

This component is a web server that is used for accessing ThinVNC server from HTTP or websockets clients.

5.7.1 Properties

5.7.1.1 DefaultPage

Sets/gets the default html page file.

ActiveX/.NET Syntax

*WebServer.***DefaultPage** [= ***string***]

Delphi Syntax

*WebServer.***DefaultPage** [:= ***string***]

5.7.1.2 HttpActive

Activates/deactivates the server's http protocol.

ActiveX/.NET Syntax

*WebServer.***HttpActive** [= ***bool***]

Delphi Syntax

*WebServer.***HttpActive** [:= ***boolean***]

5.7.1.3 HttpEnabled

Enables/disables the server's http protocol.

ActiveX/.NET Syntax

*WebServer.***HttpEnabled** [= ***bool***]

Delphi Syntax

*WebServer.***HttpEnabled** [:= ***boolean***]

5.7.1.4 HttpsActive

Activates/deactivates the server's https protocol.

ActiveX/.NET Syntax

*WebServer.***HttpsActive** [= ***bool***]

Delphi Syntax

*WebServer.***HttpsActive** [:= ***boolean***]

5.7.1.5 HttpsEnabled

Enables/disables the server's https protocol.

ActiveX/.NET Syntax

*WebServer.***HttpsEnabled** [= ***bool***]

Delphi Syntax

*WebServer.***HttpsEnabled** [:= ***boolean***]

5.7.1.6 HttpPort

Sets/gets the listening port for http protocol.

ActiveX/.NET Syntax

*WebServer.***HttpPort** [= ***int***]

Delphi Syntax

*WebServer.***HttpPort** [:= ***integer***]

5.7.1.7 HttpsPort

Sets/gets the listening port for https protocol.

ActiveX/.NET Syntax

*WebServer.***HttpsPort** [= ***int***]

Delphi Syntax

*WebServer.***HttpsPort** [:= ***integer***]

5.7.1.8 RootPath

Sets/gets the root path to the web files.

ActiveX/.NET Syntax

*WebServer.***RootPath** [= ***string***]

Delphi Syntax

*WebServer.***RootPath** [:= ***string***]

5.7.1.9 CertFile

Sets/gets the certificate file.

ActiveX/.NET Syntax

*WebServer.***CertFile** [= ***string***]

Delphi Syntax

*WebServer.***CertFile** [:= ***string***]

5.7.1.10 KeyFile

Sets/gets the key file.

ActiveX/.NET Syntax

*WebServer.***KeyFile** [= ***string***]

Delphi Syntax

*WebServer.***KeyFile** [:= ***string***]

5.7.1.11 RootCertFile

Sets/gets the root for the certificate file.

ActiveX/.NET Syntax

*WebServer.***RootFile** [= ***string***]

Delphi Syntax

*WebServer.***RootFile** [:= ***string***]

5.7.1.12 WebSocketsEnabled

Activates/deactivates the WebSockets.

ActiveX/.NET Syntax

*WebServer.***WebSocketsEnabled** [= ***bool***]

Delphi Syntax

*WebServer.***WebSocketsEnabled** [:= ***boolean***]

5.7.1.13 HttpBindIp

Binds the HTTP connection to a specific IP.

ActiveX/.NET Syntax

*WebServer.***HttpBindIp** [= ***string***]

Delphi Syntax

*WebServer.***HttpBindIp** [:= ***string***]

5.7.1.14 HttpsBindIp

Binds the HTTPS connection to a specific IP.

ActiveX/.NET Syntax

*WebServer.***HttpsBindIp [= *string*]**

Delphi Syntax

*WebServer.***HttpsBindIp [:= *string*]**

5.7.1.15 AuthenticationType

Authentication type you want to use with the server.

ActiveX/.NET Syntax

WebServer.**AuthenticationType** **[=** ***AxAxAuthType*****]**

Delphi Syntax

WebServer.**AuthenticationType** **[:=** ***TAuthenticationType*****]**

5.7.1.16 Username

Valid Username for dialog authentication.

ActiveX/.NET Syntax

WebServer.**Username** [= ***string***]

Delphi Syntax

WebServer.**Username** [:= ***string***]

5.7.1.17 Password

Valid Password for dialog authentication.

ActiveX/.NET Syntax

WebServer.**Password** [= ***string***]

Delphi Syntax

WebServer.**Password** [:= ***string***]

5.7.2 Enums

5.7.2.1 AuthenticationType

Enum that specifies the authentication type you want to useBinds the HTTPS connection to a specific IP.

ActiveX/.NET Syntax

AxAxAuthType = (atNone, atDigest, atNtlm)

Delphi Syntax

TAuthenticationType = (atNone, atDigest, atNtlm)

5.8 ThinWindowScraper

This scraper is much like the ThinScreenScraper, but it allows you to add non-top windows to be captured.

5.8.1 Methods

5.8.1.1 AddWindow

Lets you add a single window as a capture source. Unlike ThinScrenScraper's AddTopWindow, it also allows you to capture non-top windows.

ActiveX/.NET Syntax

*Scraper.***AddWindow (HWND wnd)**

Delphi Syntax

*Scraper.***AddWindow (wnd: HWND)**

6 Purchasing ThinVNC SDK

By purchasing any edition of ThinVNC SDK you will access to technical support, free upgrades and updates and the activation of advanced features in your edition.

In this section you will find information regarding the different licensing options you have that will help you choose the type of order you need to place. Also this section explains how to place your order and finally activate your product so that you can enjoy all of the ThinVNC SDK benefits.

- Licensing Information
- How to Place an Order
- Registering ThinVNC SDK Trial Version

6.1 Licensing Information

Cybele Software offers perpetual licensing for all its products.

We offer Technical Support by e-mail and/or phone, which also includes free updates and upgrades during the covered period and our full commitment to timely fix bugs and problems.

We encourage users to renew the annual maintenance contract in order to be eligible for technical support and product upgrades. The maintenance fee after the first year will still be 20% of the updated price of the purchased product.

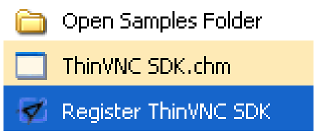
Cybele Software offers volume pricing according to the amount of the purchase.

If you have any other question, contact us at sales@cybelesoft.com. Our sales representatives will get in touch with you to assist you with your licensing situation.

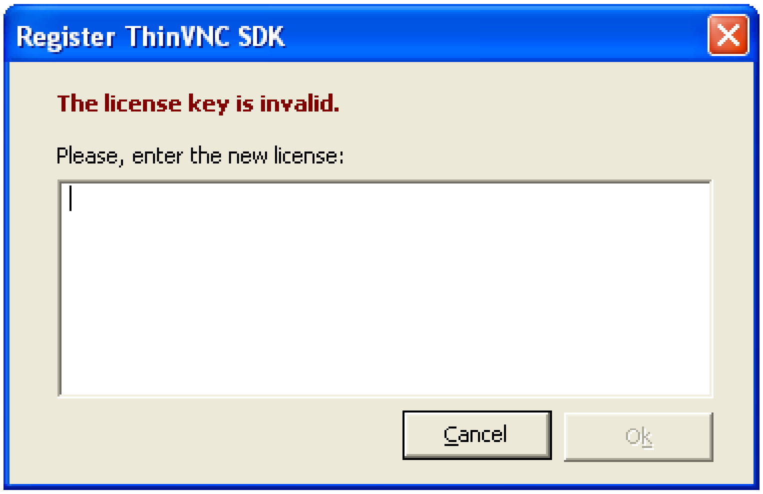
6.2 Registering ThinVNC SDK Trial Version

If you downloaded ThinVNC SDK Trial version from our web site or a distribution site and you have already purchased a license, you must follow these steps in order to register the product:

- Look for 'Register ThinVNC SDK' in the start menu under 'All Programs' - 'ThinVNC SDK':



By pressing this option you will be presented with the following prompt:

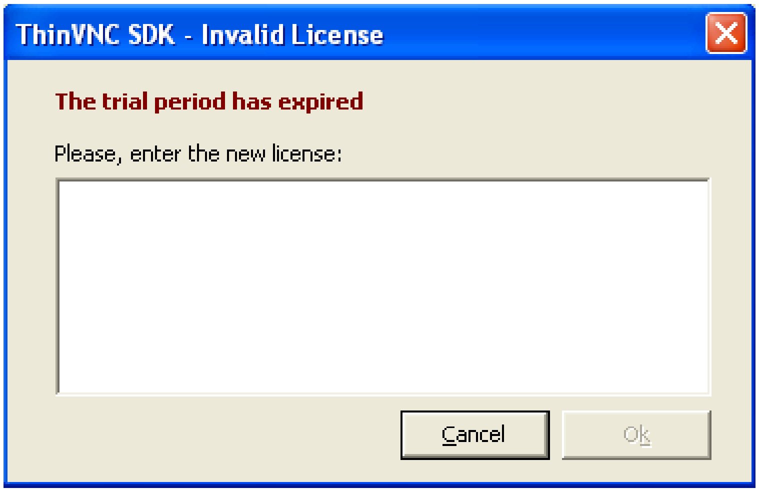


The license key is invalid.

Please, enter the new license:

Open your *.xml license file with a text editor and copy the entire content into the dialog box. Afterwards, press 'Ok'.

- If you haven't entered a license in a month's use, the software will prompt you for the license:



The trial period has expired

Please, enter the new license:

Enter the license information and press 'OK'.

6.3 How to Place an Order

There are many ways to order your ThinVNC SDK licenses:

- Contact us at sales@cybelesoft.com. Let us know about your licensing needs and we will send you an official quotation. Our sales representatives will get in touch with you to assist you with the purchase.

- You can also call us anytime to any of these phone numbers and place the order immediately:

  Toll Free: 1-866-462-9768
  Local line: 1-302-892-9625
  Fax: 1-302-295-9995

- You can also contact us through Live Chat by pressing this icon in our website:

  Live Help - Online
  click for live support

  and immediately have a conversation with a representative without even having to pick up the phone.

When you buy ThinVNC SDK, you will receive a Key to register the Trial version. For instructions on how to register ThinVNC SDK when you purchase a license, see Registering ThinVNC SDK Trial Version.

7 Obtaining Technical Support

Cybele's goal is to offer high quality products and services to increase the efficiency and ease-of-use of legacy systems. The whole Company focuses on this goal, and the results of our unique expertise are our reliable solutions. We believe passionately that modern, solid and feature-rich host access solutions can actually increase its users' productivity.

Technical support is a very important benefit to consider, especially when it comes to mission critical software solutions.

Using registered Cybele Software's applications not only allows you to receive free product upgrades and updates but also the certainty that you will have our team of experienced developers and technical support representatives working hard to assist you with any issue, thus making the product much more accessible in any situation.

We are here to help you out from monday to friday 9 a.m. to 5 p.m. eastern time on the phone numbers:

Toll Free: 1-866-462-9768
Local line: 1-302-892-9625
Fax: 1-302-295-9995

If you make your call outside this hour range, you can leave a message and we will get back to you.

You can send us an email to support@cybelesoft.com and we will write you back timely. You can also contact us through Live Chat by pressing this icon in our website:

and immediately have a conversation with a representative without even having to pick up the phone.

Cybele Software Inc.
3422 Old Capitol Trail, suite 1125
Wilmington, DE - 19808
Phone: (302) 892-9625
Fax: (302) 295-9995
e-mail: support@cybelesoft.com
http://www.cybelesoft.com