



Remote Core .NET Viewer SDK General description

Remote Core SDK – is a set of libraries that implements the functionality of the VNC Client (Viewer), allowing developers (users) to easily integrate this functionality into their applications.

The Remote Core .NET Viewer SDK is the part of the Remote Core SDK that implements VNC Viewer functionality for the .NET environment.

The Remote Core .NET Viewer SDK provides ready-to-use controls for WPF and WinForms applications. Using these controls - is the most convenient way to integrate remote desktop (VNC) functionality into your application.

In addition to the ready-to-use controls for UI frameworks, the SDK provides a lower-level (independent of specific UI frameworks) component – a Core component. This component implements the functionality of the VNC client, encapsulating all the complex implementation details, hiding them behind a convenient API.

Implementing a functionality of the VNC client, the Core component, in particular, establishes a connection with a remote VNC server, provides interaction with this Server via the RFB protocol, maintains the current state of the FrameBuffer and keeps the user code updated regarding FrameBuffer changes, and transmits user input events to the server side.

The Core component underlies WPF and WinForms controls. This component can be used to implement similar controls for other .NET compatible UI frameworks. As an option this component can be used in situations where the UI is not needed (for example, to develop an application that takes screenshots).

Supported versions of .NET

Supported versions of .NET are listed in the “README.txt”.

RFB protocol implementation

The SDK implements the RFB protocol according to the RFC and therefore is compatible with any software that implements such protocol.

Implementation of the additional features of [TightVNC](#)

In addition to the RFB standard, the SDK implements additional TightVNC extensions provided by implementation of the standard protocol:

- Comprehensive compression algorithms.
- Improved performance.
- File transfer functionality.
- UTF8 clipboard support.
- Connection health check.
- The ability to implement and add your own extensions if necessary, etc.

Communication possibilities with VNC Server

The .NET Viewer SDK supports flexible communication possibilities with VNC Server. Connecting to a remote Server can be accomplished in different ways:

- Direct connection establishment. The SDK components independently establish a connection with server, receiving the host address and port from the user. The most obvious and comprehensible way.
- Use of a ready connection. SDK components can accept previously established connections as a Socket type object. This option can be useful when the Viewer side accepts an incoming connection from the Server (as Listening Mode in TightVNC Viewer for example).
- Using a Data Stream. The SDK components can accept a custom data stream as an object of Stream type. This option can be useful, for example, when there is a need to accept encrypted data or data received via some specific (non-TCP) channels, to play back recorded sessions, etc.

Automatic connection recovery

The .NET Viewer SDK supports Automatic connection recovery. SDK components can keep track of a connection state and automatically restore the connection in case it is interrupted.

Rendering options

The .NET Viewer SDK supports flexible rendering options. UI controls provide the ability to customize rendering of the video stream from the Server in a flexible way. Additional information is presented in the “Rendering.pdf”. Most of the options are available in compiled “Demo.exe”.

Customization possibilities

The .NET Viewer SDK supports flexible customization possibilities. WPF control provides wide opportunities for customizing the appearance of the application interface. The option can be useful, for example, if you want to add your own labels, customize the password entry dialogs and the screen in the connection setup mode.