# PG2 VNC Client

## Index



## Initiation

This Program can be used to display remote computer screens into 800xA PG2 graphic displays by using the old, but relyable VNC protocol.

## Background Story

On my plant we have some engines with independent controllers. This controllers are common industrial PC's, running Windows. All of them have also some small and ugly operator panels. This panels are also small PC's running some kind of Windows. The pannels are connected to the controllers over ethernet, using the VNC protocol. This operator panels are difficult to use because no keyboard, only a inaccurate touchscreen. This is why we decide to give our operators access to this engines from 800xA. We connect this engines to the 800xA network, and now we can access the engines by using a standard VNC client. That was already very handy, but we won't give our operators access to a full grown VNC client. So we integrated a VNC client into some 800xA PG2 graphics. Now the 800xA graphic display shows the engine's operator panel.

## Disclaimer

The author of this program is not liable for damage to software or hardware or property damage incurred by the use of the program.

## License

This program is freeware. My parts of the source code are subject to the 2-clause BSD license, the parts of David Humphrey's VncSharp are subject to the GPL2 license.

## Prerequisites

You will need:

- 800xA 5.1 with PG2 graphics
  or
- 800xA 6.0 or 6.1

## Compatibility

The program has been successfully tested with the 800xA versions:

- 5.1 FP4 (Win7)  Use the installer in 5.1 path!
- 6.0 and 6.1 (Win8.1)  Use the installer in 6.0 path!

and with this VNC-Servers:

- RealVNC (Windows)
- x11vnc (Linux)
- TightVNC (Linux & Windows)

## VNC and Security

VNC is a very old protocol. Encryption and secure login are not mandatory features of VNC, and the encryption of VNC does not count as secure enymore. So this VNC Client does not support encryption, or everything else than basic authentication. If you want to use this VNC client, you should keep in mind:

- No encryption. It is possible to spy keystrokes and screencaptures send trough the network.
- No secure login. VNC does not know usernames, only a few passwords for different access levels. The password will be transfered encrypted, but this kind of encryption is weak.
- The VNC Password is (maybe) visible into PG2 Graphic Editor, and Graphics Documentations.

Summary:

**DO NOT USE VNC OVER THE INTERNET!**
**DO NOT USE VNC OVER UNPROTECTED LAN NETWORKS!**
**DO NOT USE VNC FOR SECURE CRITICAL APPLICATIONS!**

If you really need to use VNC over insecure networks, you should consider to use additional secutity options like VPN or SSH Tunneling.

## Installation

- Run **install.bat** on every node where you want to use the VNC Viewer.
- Close every workplace and graphics builder on this node. Re-open the workplace.
  (The first workplace started on this node reads all graphic primitives to the cache.)

On nodes without the VNC Viewer installed the graphic display will display a crossed purple rectangle instead of the VNC Viewer.

## Uninstallation

- Close every workplace and graphics builder on this node.
  (The dll file is write protected if one workplace is still running)
- Run **uninstall.bat** on every node where the VNC Viewer is installed.

## Update

- Close every workplace and graphics builder on this node.
  (The dll file is write protected if one workplace is still running)
- Replace the old **c:\Program Files (x86)\ABB Industrial IT\Operate IT\Process Portal A\bin\GraphicPrimitives\PG2VNCViewer.dll** by the new one.

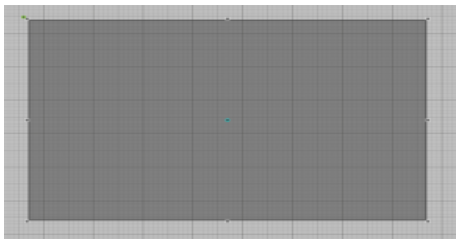
## Known Issues

- Empty passwords are not supported.
- Some keys does not working. Some keys are catched by 800xA, some keys are maybe missing in the keycode conversion table.
- Keyboard encoding defaults to german.  Most keys should be equal on other keyboards. If you find some wrong keys, you can change the encoding with the **Keyconvert** property.
- No Sound support. (VNC does not support sound at all)
- No clipboard support yet. If someone needs it, it should be easy to add.

## Configuration

After successfull install, a new control appears in your Process Graphics Editor's toolbox.


Add this control to a Graphic Display.



(Not really impressive, I know)

Apart from the usual properties the control contains this new ones:

**Connect**: True: Connect to VNC-Server, (tries to reconnect every 60s if the connecion lost)
False: Disconnect from VNC-Server

**Hostname**: The hostname or IP-Address of the VNC-Server

**Hostport**: Port of the VNC-Server

**Password**: Password to Login into VNC-Server

**Shared**: Allow multiple client connection to the VNC-Server. Depending of the kind of VNC-Server this setting will be ignored by the Server.

**Viewonly**: Disables local mouse and keyboard event transfer.

**Hideremotecursor**: Hides the remote mouse cursor to improve performance and appearance. As drawback we cannot see the remote mouse state. (like hourglass...)

**Loglevel**: Enables collection and display of log messages.

| | | |
|---|---|---|
| -1 | clear | deletes all existing messages |
| 0 | none | no messages |
| 1 | error | error messages only |
| 2 | info | info and error messages |
| 3 | debug | all messages |

**Stretchdraw**: True: stretch the size of the remote screen to the size of the local screen.
False: keep the remote screen resolution. Oversized will be cut away.

**Keyconvert**: Custom keyboard encoding table. See chapter *Keyboard encoding*.

**Keysim***: To simulate keyboard press. See chapter *Keyboard simulation*.

**Mousesim***: To simulate mouse button press. See chapter *Mouse simulation.*

## Keyboard encoding

The VNC protocol has it‘s own keyboard codes. We are using .Net technology, who only provides an enumeration instead of key codes. So a konvertion table is needed. I implemented a fixed convertion table for the only keyboard type I have access to: german locale.

For the case some of you need a different type of keyboard locale, or some else modification, I implemented the **Keyconvert** property. In this property you can set additional or replacement key encodings.

Each single key encoding have to be in **Net=Key** format. Multiple key encodings have to be separated by comma. There is no space characters allowed.

**Net** must be one of this values:

None Cancel Back Tab LineFeed Clear Return Pause Capital KanaMode JunjaMode FinalMode HanjaMode Escape ImeConvert ImeNonConvert ImeAccept ImeModeChange Space PageUp Next End Home Left Up Right Down Select Print Execute Snapshot Insert Delete Help D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z LWin RWin Apps Sleep NumPad0 NumPad1 NumPad2 NumPad3 NumPad4 NumPad5 NumPad6 NumPad7 NumPad8 NumPad9 Multiply Add Separator Subtract Decimal Divide F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 NumLock Scroll LeftShift RightShift LeftCtrl RightCtrl LeftAlt RightAlt BrowserBack BrowserForward BrowserRefresh BrowserStop BrowserSearch BrowserFavorites BrowserHome VolumeMute VolumeDown VolumeUp MediaNextTrack MediaPreviousTrack MediaStop MediaPlayPause LaunchMail SelectMedia LaunchApplication1 LaunchApplication2 Oem1 OemPlus OemComma OemMinus OemPeriod OemQuestion Oem3 AbntC1 AbntC2 OemOpenBrackets Oem5 Oem6 OemQuotes Oem8 OemBackslash ImeProcessed System OemAttn OemFinish OemCopy DbeSbcsChar OemEnlw OemBackTab DbeNoRoman DbeEnterWordRegisterMode DbeEnterImeConfigureMode EraseEof Play DbeNoCodeInput NoName Pa1 OemClear DeadCharProcessed

If you not know which one to take, set the Loglevel property to 3 and just press the key. You shold see keypress messages with the Net keyname in the red message text. Keyboard handling does not work in graphics editor, you have to check the keys in workplace mode.

It is also possible to add one *shift+* , *alt+* or *strg+* modifer.

**Key** must be decimal numeric. To find the right one you can take a look into the keycodes.txt file. This file is a partial copy of the keysym.h file from one popular open source VNC implementation. The numbers in this file are in the hexadecimal format, you have to convert them to decimal first.

Examples:

| | |
|---|---|
| „a=98,b=97,shift+a=66,shift+b=65“ | swaps a and b, A and B |
| „NumPad5=65299“ | Sets Numpad „5“ to „Pause“ |
| „Apps=65383“ | Enables the „Menu“ key |

## Keyboard simulation

You can simulate up to 3 keyboard events by setting the Keysim* properties.

The **KeySim*Keycodes** property hold one or more Key codes, separated by space. See keycodes.txt to get the right codes. The numbers in this file are in the hexadecimal format, you have to convert them to decimal first.

Examples:

| | |
|---|---|
| „65507 65513 65535“ | strg+alt+del |
| „65507 99“ | strc+c |
| „65507 118“ | strg+v |

If the **KeySim*Trigger** property raises from false to true, the keycodes will be sent as „key pressed“. From true to false the keys will be sent as „key released“.

The keyboard simulation function is not intended for typing long text in, because all keys you entered will be „pressed“ at once, and „released“ at once.

## Mouse simulation

You can simulate up to 3 mouse clicks by using the **MouseSim*** properties

The **MouseSim*X** and **MouseSim*Y** holding the screen coordinates. This are serverside pixel coordinates. To find out the right numbers you can set the Loglevel property to 3 and click. You should see mouseevent messages with coordinates and button codes in the red message text.

With **MouseSim*Button** you can chose the simulated mouse buttons. This value is just a bitmask combination. You are familliar with bits?

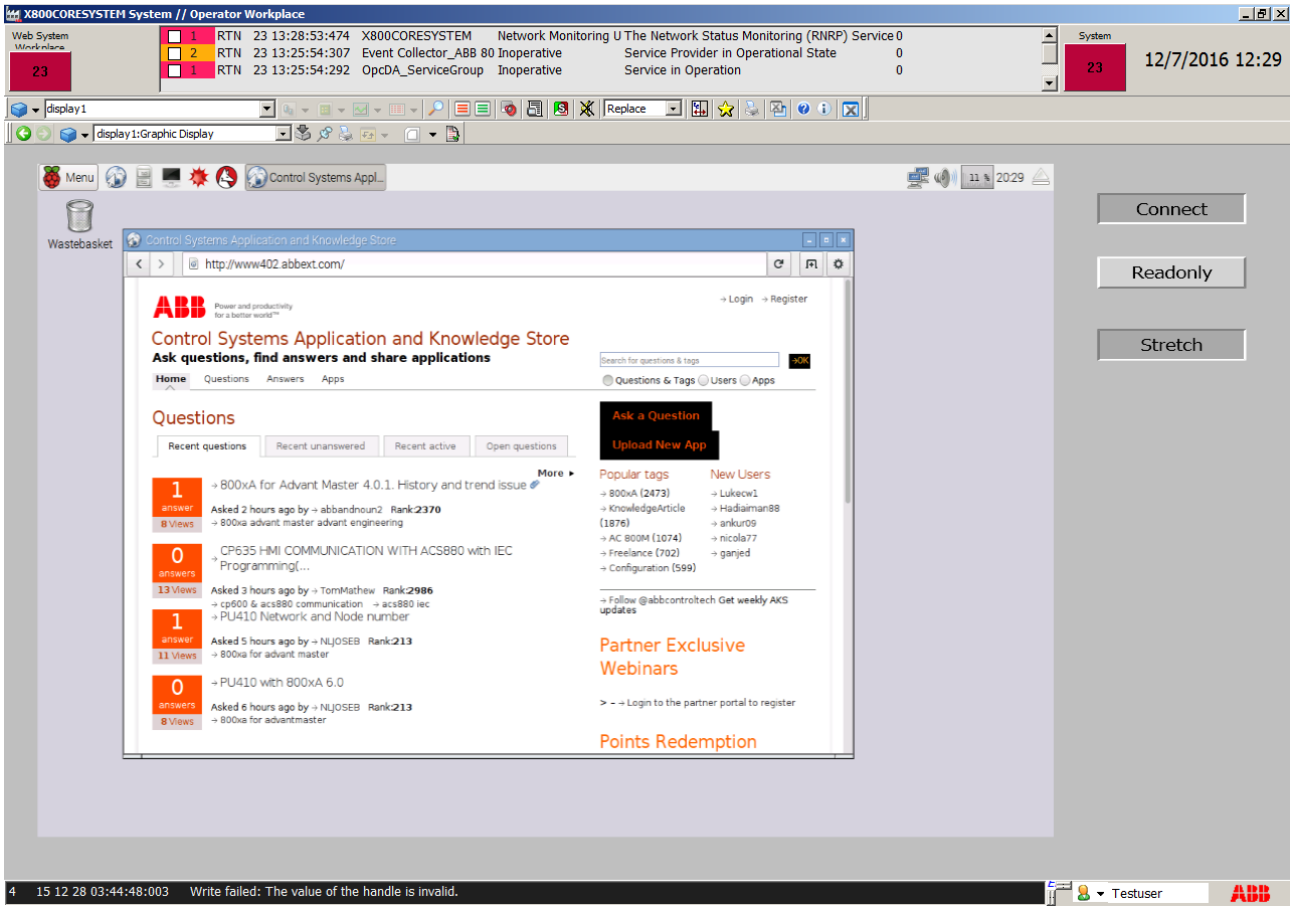
Bit0 (Value 1) is the left mouse button.
Bit1 (Value 2) is the middle mouse button (~the wheel button).
Bit2 (Value 4) is the right mouse button.

If the **MouseSim*Trigger** property raises from false to true, one mouseevent will be sent with the X, Y and Button value. If it falls from true to false one mouseevent will be sent with X, Y and „0“ (no mouse button pressed).

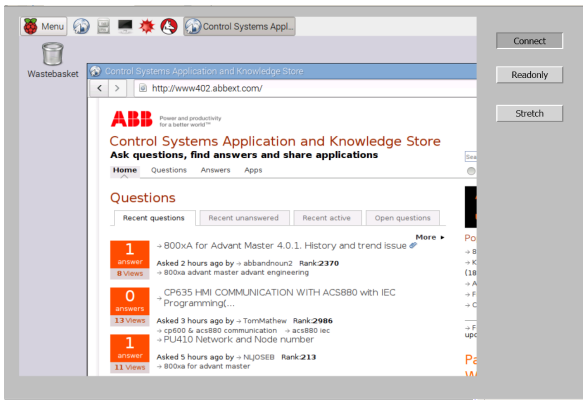
# Usage Examples

Not really useful examples, but they shows the possible usage well.

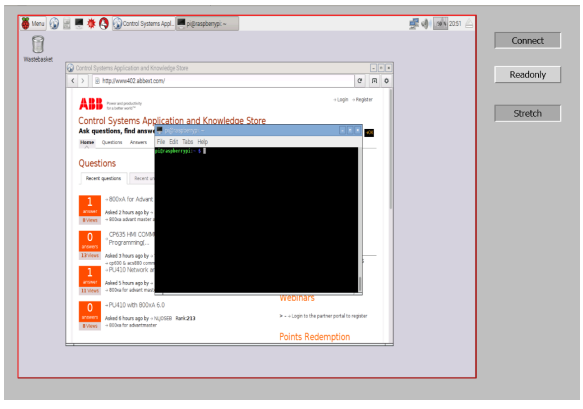
## Example#1 x11vnc, running Raspberry PI desktop




The Connect, Viewonly, Stretchdraw properties are bound to toggle buttons.

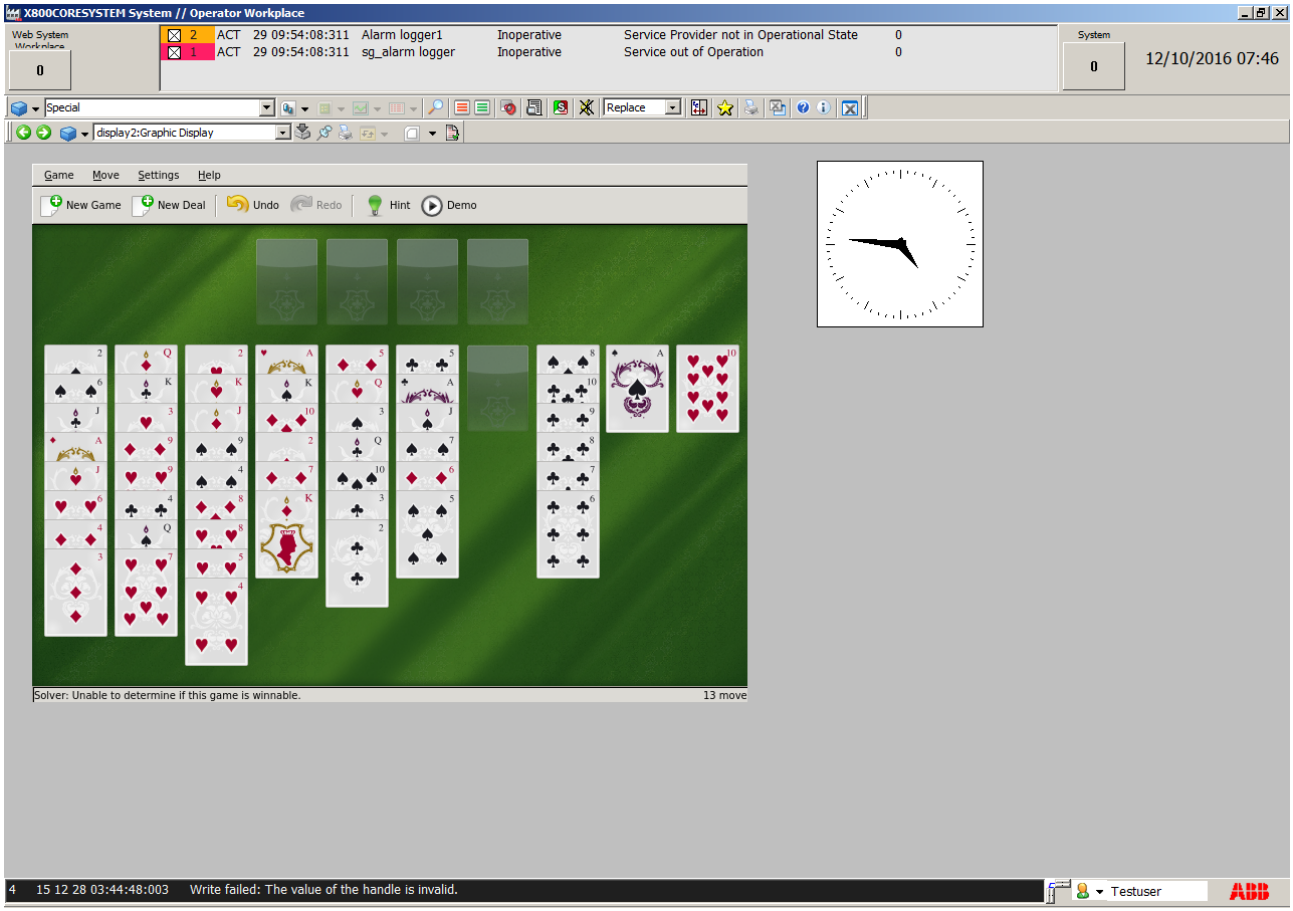


Here you can see the effect of Stretchdraw.
(the remote desktop is too big)



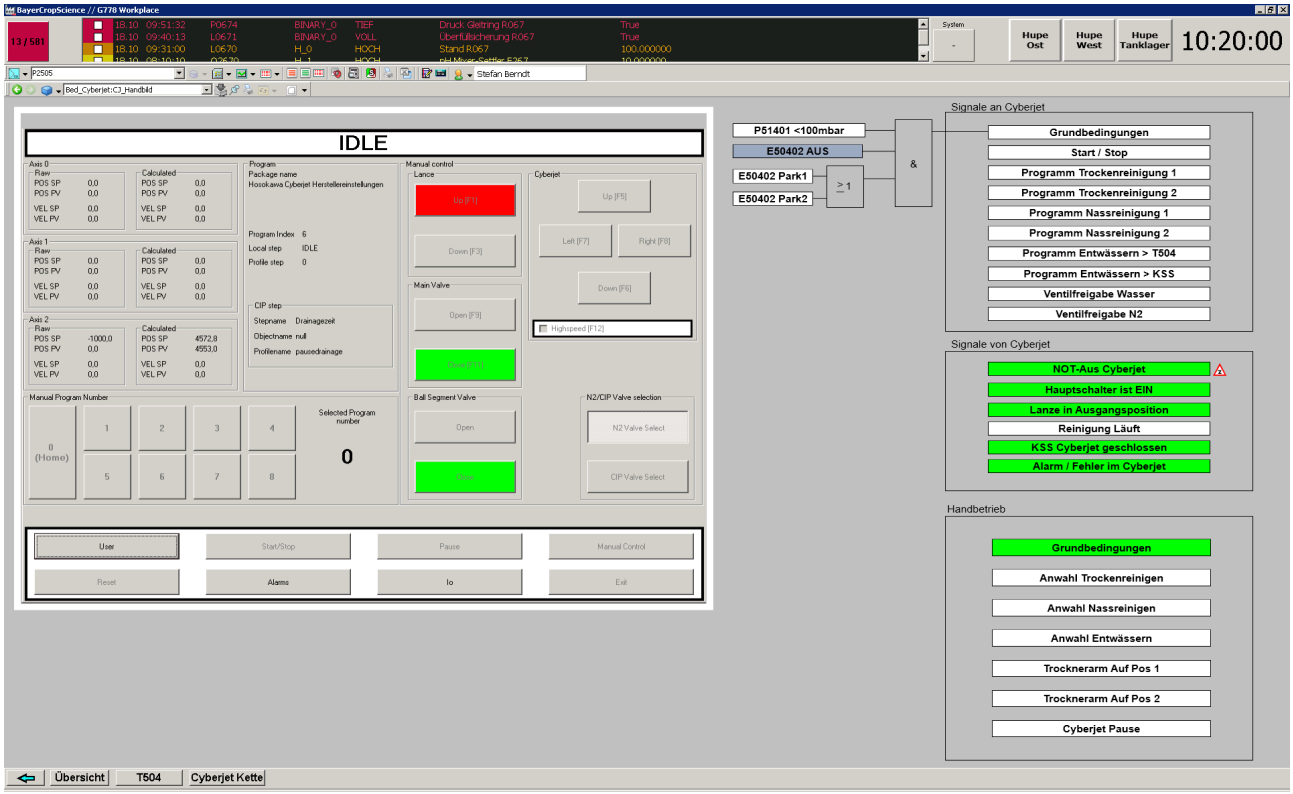
The alarmframe indicates keyboard focus.
(requires setting FrameWidth > 0)

## Example#2 TightVNC, running KPatience and xclock




Under Linux we can start single applications within virtual screens.
This is KPatience on screen:1 (Port 5901) and xclock on screen:2 (Port 5902).
„Connect“ is set to True, so the game and the clock is shown immediately after open the graphic.

## Example#3 At last one real life example




This is one of the cleaning machines in my plant. The VNC control is in the big white square in the left half.