

# elinkKVM Manual

# TABLE OF CONTENTS

1. About ELINKGATE .....	1
2. Introduction .....	2
2.1. Overview .....	2
2.2. Booster technology .....	3
2.3. Features .....	4
2.4. elinkKVM Physical Connections .....	5
2.5. Technical Specifications .....	8
2.5.1. elinkKVM Hardware Specifications .....	8
2.5.2. elinkKVM Mechanical Specifications .....	9
2.5.3. elinkKVM Firmware Specifications .....	9
2.5.4. elinkViewer Software Specifications .....	9
2.6. Components .....	11
2.6.1. Front view .....	11
2.6.2. Rear view .....	12
3. elinkKVM quick start guide .....	13
3.1. Package Contents .....	13
3.2. Power On elinkKVM .....	13
3.3. elinkKVM in Setup Mode .....	15
3.4. elinkKVM in Basic Operation Mode .....	16
3.5. Network configuration .....	17
3.6. Setup eLinkViewer .....	17
3.7. elinkViewer quick start .....	18
4. elinkViewer .....	23
4.1. Login to elinkKVM .....	23
4.1.1. Establish connection .....	23
4.1.2. Scan for elinkKVM .....	24
4.1.3. Device authentication .....	25
4.1.4. Change Connection Options .....	25
4.1.5. elinkViewer Configuration .....	26
4.2. Access Remote Computer .....	27
4.2.1. Remote Access GUI Layout .....	27
4.2.2. elink Configuration GUI .....	29
KVM Setting .....	30
Virtual Media .....	31
Quick Setting .....	32
4.2.3. Pause and Capture Image .....	32
4.2.4. File Transfer GUI .....	34
4.2.5. IPMI Command Center .....	37
4.2.6. User Management .....	41

4.2.7. elink Event Log .....	43
4.3. Running Python Script .....	44
4.3.1. Setup Python Console .....	44
4.3.2. How to call elinkViewer Python API .....	45
4.3.3. elinkViewer Python API .....	46
elink Module .....	46
VNC Object .....	47
elink Event Object .....	57
5. eLinkKVM Advance Configuration .....	58
5.1. Local Mode .....	58
5.1.1. Network Tab .....	58
5.1.2. DHCP Tab .....	59
5.1.3. IPMI Tab .....	60
5.2. Offline Configuration .....	61
5.3. Factory Reset .....	62
6. Booster .....	64
6.1. Booster Configuration .....	65
6.2. Booster for Windows .....	66
6.3. Booster for Linux (Non supported yet) .....	67
6.4. Booster for UEFI .....	67
6.5. Embbed Booster into operating system setup images with eLinkSetupTool .....	67
6.5.1. eLinkSetupTool instalation .....	67
6.6. eLinkSetupTool usage .....	68
6.6.1. Create OS setup image .....	68
6.6.2. Create a minimal floppy image .....	69
6.6.3. Install UEFI Booster agent to an existing disk image .....	69
6.6.4. Display version information: .....	69

# CHAPTER 1. ABOUT ELINKGATE

---

ELINKGATE is a company that aims to help people solve the computer problems. ELINKGATE has patented a technology called eLinkMe – a USB device that allows an IT expert to remotely access a computer, smart phone or tablet to fix any problem the user may have. Our eLinkMe technology has the following advantages:

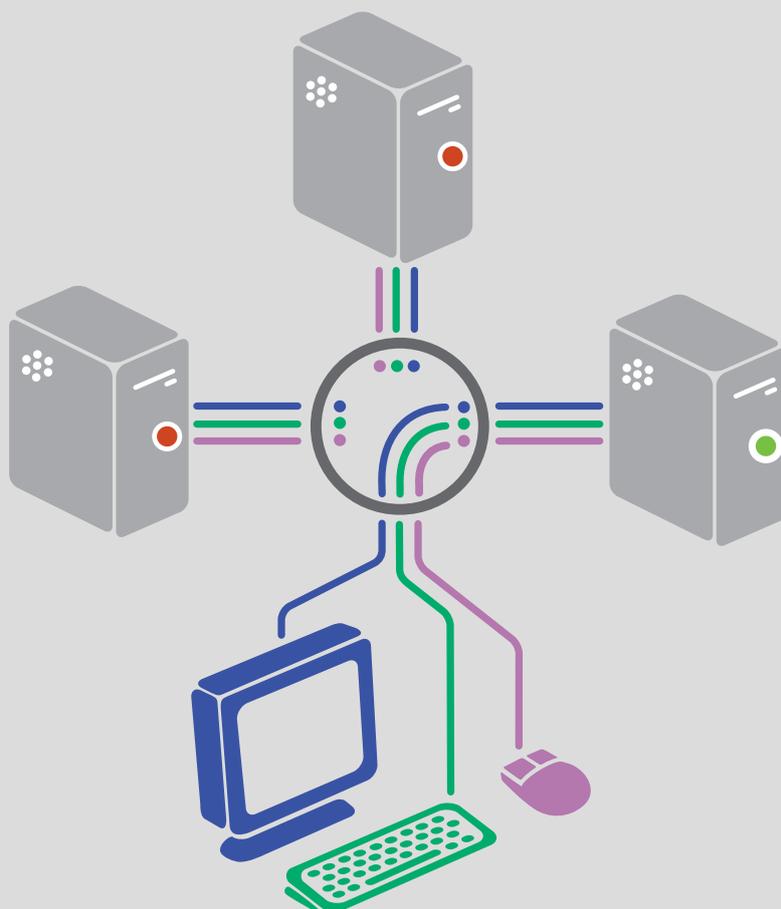
- **Simple:** No pre-install software or hardware required
- **Smart:** Ability to work when there is no OS (works at BIOS level)
- **Safe:** Protects user data and service provider data

It is our belief that the eLinkMe technology, as well as our future technologies will enable users to easily request online service. Inexperienced users lacking sufficient knowledge about computers such as farmers, housewives and the elderly, can also begin to find support that will make their life better. eLinkMe can also assist service providers in providing more services online, while allowing customers to review their results face to face, without worrying about online theft.

## CHAPTER 2. INTRODUCTION

### 2.1. Overview

A KVM switch (an abbreviation standing for **K**eyboard, **V**ideo and **M**ouse") is a hardware device that enables a user to control multiple computers from one or more sets of keyboards, video monitors, and mouse devices.



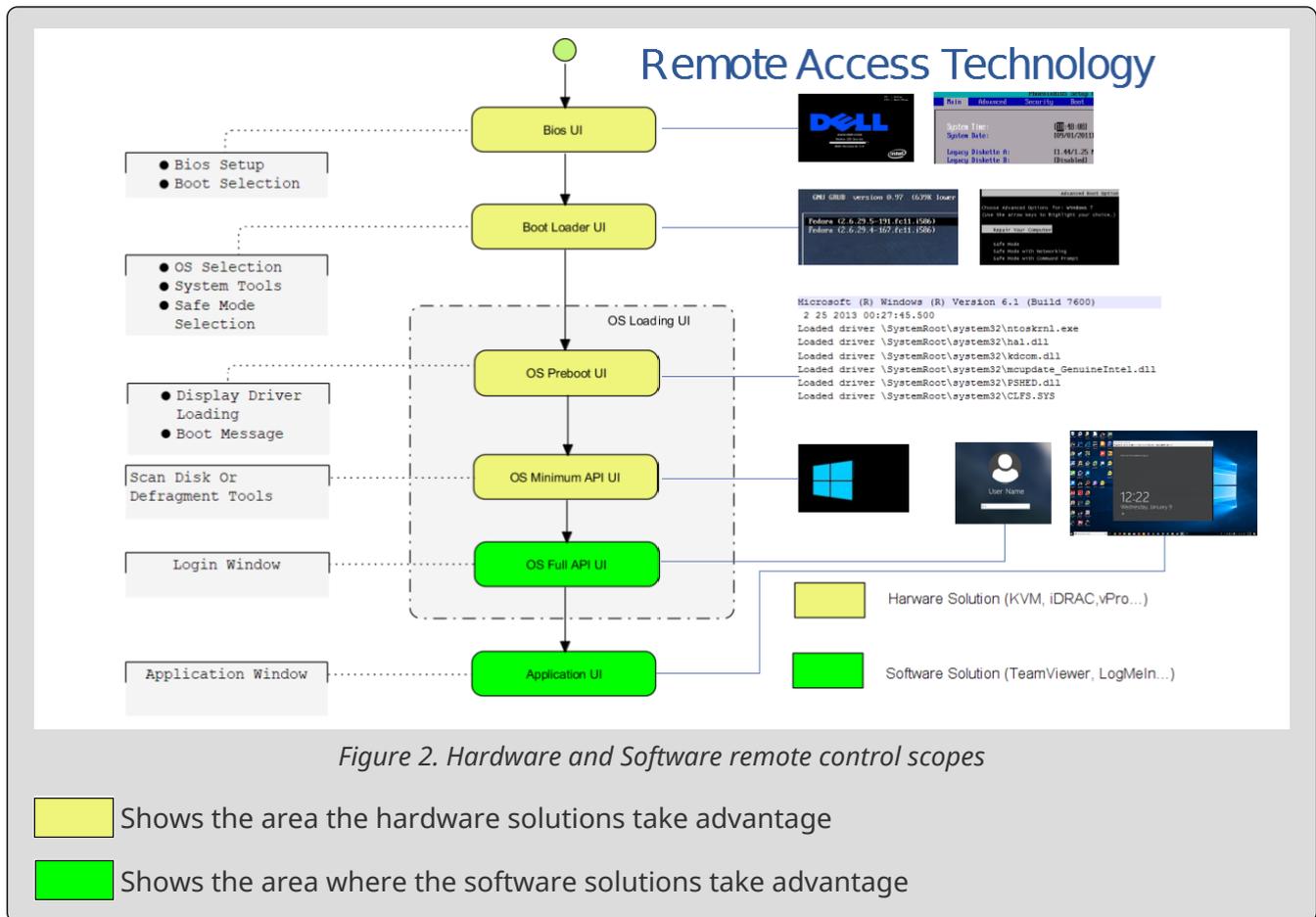
*Figure 1. KVM Switch interworking*

The computer on the right is currently being controlled by the peripherals. Only one set of keyboard and mouse is required to control multiple computers, each connected to a monitor. A user can switch between each monitor to see the output of each connected computer (to that monitor).

Normal KVM switches attach controlling and controlled devices directly to them. In contrast, KVM over-IP switches provide over-the-internet functionality. It means that instead of physically connected to a KVM switch through very long cables and usually limited in a room, a floor or a building, the controlling devices can be anywhere in the world that can connect to a KVM over-IP switch and control the connected computers with a normal Internet connection.

Another approach for remote controlling is using pure software. However, a pure software solution is usually limited only to an operating system environment. Without an operating system, a software solution is infeasible.

Unlike software solutions for remote controlling, a hardware solution, like a KVM switch can be used as a replacement for a computer monitor and can work in either operating system or non-OS (such as Bios, UEFI). The below diagram [Figure 2](#) demonstrates the scope difference between a software and hardware solution:



elinkKVM is one of such KVM-over-IP devices. However, unlike other KVM-over-IP solutions that are solely hardware solutions, elinkKVM adopts a unique hybrid approach that combines both hardware and software. While being a hardware solution similar to other KVM-over-IP switches, but unlike other KVM-over-IP switches, elinkKVM takes a step further with its patented technology called **Booster**. **Booster** is a software solution that works together with elinkKVM hardware to uplift processing performance by utilizing computing resources of the remote hosts. **Booster** is not limited to an operating system environment, but it also works in BIOS and UEFI environments.

## 2.2. Booster technology

**Booster** is a patented technology that allows elinkKVM to utilize computing resources from remote hosts to improve remote performance. To quickly summarize, **Booster** includes software agents that work on major OS platforms (Windows, Linux and Mac OS). However, unlike traditional software solutions, **Booster** agents can also operate in BIOS and UEFI environments, independent of any operating system.

These software agents are installed on remote hosts to utilize the extra computing power on the remote hosts to accelerate screen data processing without affecting performance. This enables higher quality remote screen and more responsive interactions with the remotely remote hosts.

**Booster** also helps reduce the loads on elinkKVM devices, making elinkKVM devices run cooler, and

extend the longevity of elinkKVM devices, while delivering higher performance.

The following diagram shows the scope covered by Booster:

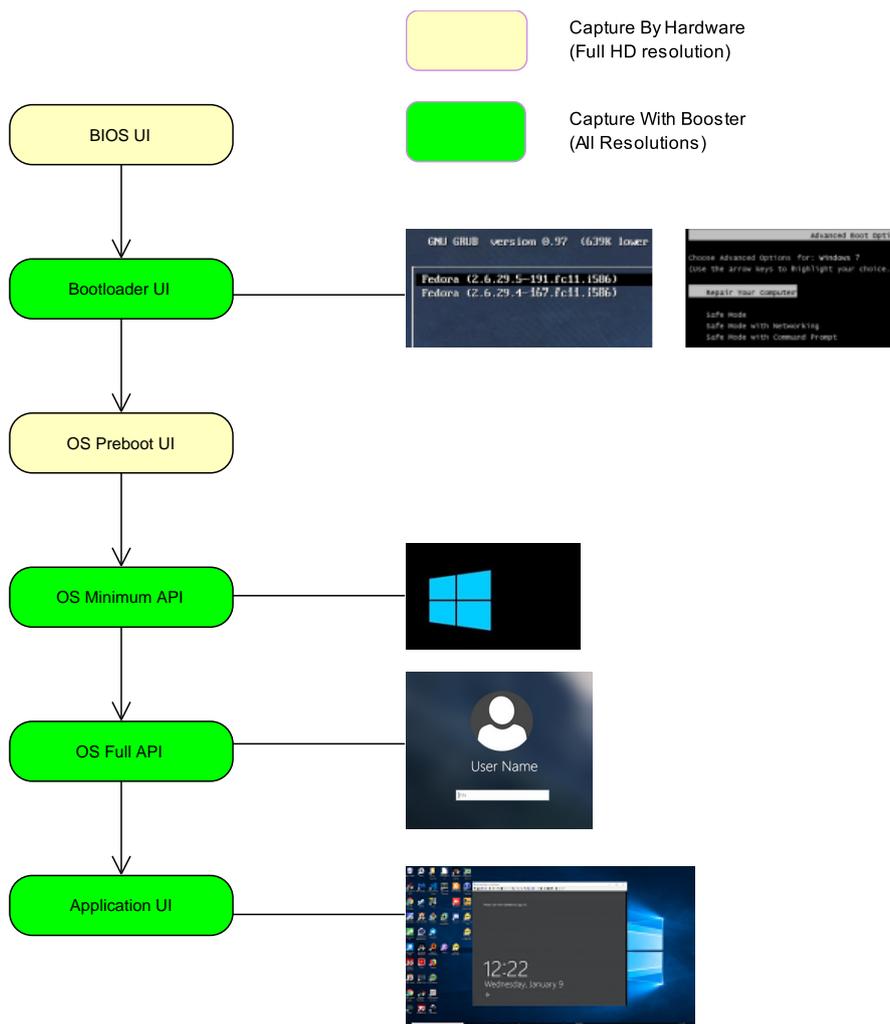


Figure 3. Booster operating scope

## 2.3. Features

The elinkKVM is designed to help the IT person remotely operate the Server in most responsive and efficient way. The device is not only allowing remotely access at Bios Level, which is similar traditional KVM over IP, but also adding more features:

### Full Bios-level control over an IP Network

The video capture hardware (Hardware mode) is designed to decode the video signal up to 1920x1080 that helps to monitor the screen in any situation.

### Patented Booster technology

Faster and higher resolution with **Booster** technology, a patent technology that allows to monitor the screen with faster transfer and any video resolution with just a USB connection for almost any situation. See the **Booster** chapter for more information.

### Feature-full native client software

Client software is bundled with many unique features: scripting and image recognition, to automate manual operations. See the elinkViewer chapter for more information.

### **Virtual Media support with 8GB Internal Flash or 64GB with SD Card**

Instead of using a separate computer as a share network storage to emulate the Virtual Media lets enjoy a high performance local storage device.

### **USB-Powered Design**

The device can just power from USB port of Server; no external power supply is required.

### **RS232 Serial Port**

For Serial Console, Remote Dial In or control external PDU.

### **Built-in IPMI (Intelligent Platform Management Interface) utility**

eLinkKVM understands IPMI and allows to control the server that supports IPMI interface. The user may power on/off/reset server without using an external PDU.

### **Support USB Host Interface**

The USB Host port allows eLinkKVM to control the USB 3G/4G, external USB storage, or USB security device to add more functionality.

### **Flexible Two Ethernet Ports**

The two Ethernet ports can be configured to work independently to connect 2 different networks or bridge together to work as local switch.

### **Built-In USB Hub**

Physically emulates all USB devices (Mouse, Keyboard, Storage) with just one USB cable.

### **Scalability**

eLinkKVM can scale to control more than one computers by connecting the computers to eLinkSwitch.

### **Compact form factor**

Small footprint of eLinkKVM makes attaching it to existing servers very easy.

### **Low cost**

Ideal for distributed IT system environments such as small branch offices, campuses, test labs, and server hosting environments.

### **Remote firmware upgrade**

Firmware can be easily upgraded remotely with GUI or scripting for batch upgrade.

## **2.4. eLinkKVM Physical Connections**

eLinkKVM can be flexibly setup to be connected between a remote terminal and a remote host with many options.

To connect a remote terminal over IP, the following methods can be used:

- Connect eLinkKVM directly to a router. A remote terminal can then access eLinkKVM over the Internet.
- Connect eLinkKVM directly to a USB 3G/4G. A remote terminal can then access eLinkKVM over the Internet.
- Directly connect eLinkKVM to the remote terminal. A remote terminal can then access eLinkKVM

with a direct connection; Internet is not required.

After setting up, elinkKVM is ready to be accessed by a remote terminal.

To connect a remote host to an already connected elinkKVM device, the following methods can be used:

- Connect a VGA cable between the remote host and elinkKVM to capture the host screen.
- Using a RS232 cable to capture its serial output.
- Connect the remote host to the Slave port on elinkKVM to manage through IPMI interface.

After physical connections on both ends are done, a remote connection is up and running.

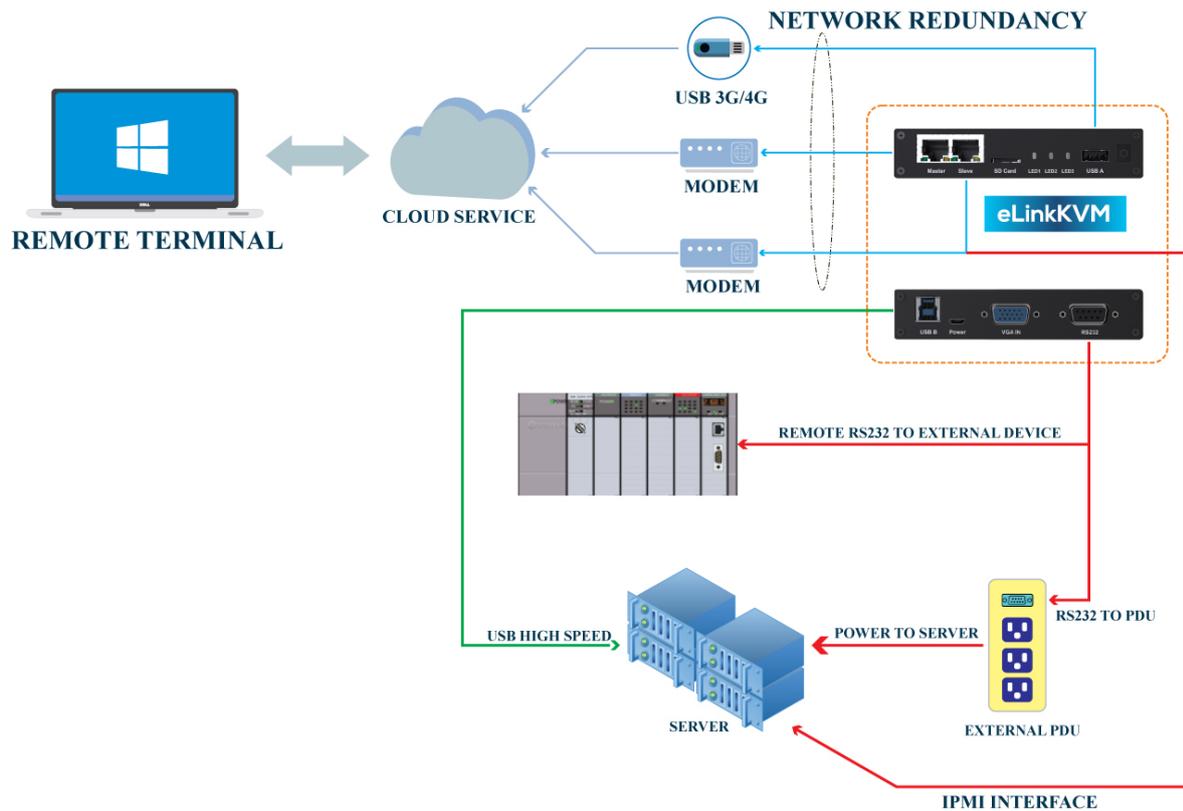


Figure 4. Complete system elinkKVM connect to Router, connect to IPMI server, USB, RS232, external PDU, USB 3G, etc.

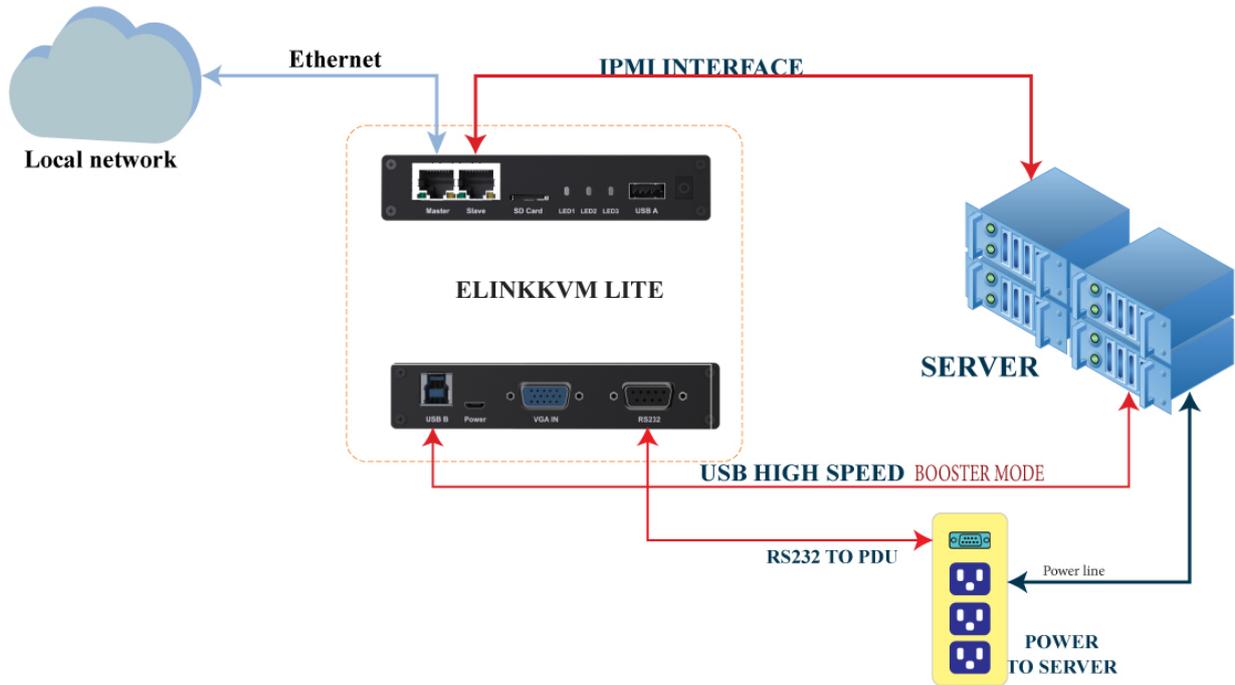


Figure 5. elinkKVM Lite A minimal connection just USB (to run Booster), elinkKVM connect to Router, elinkKVM connect to IPMI.

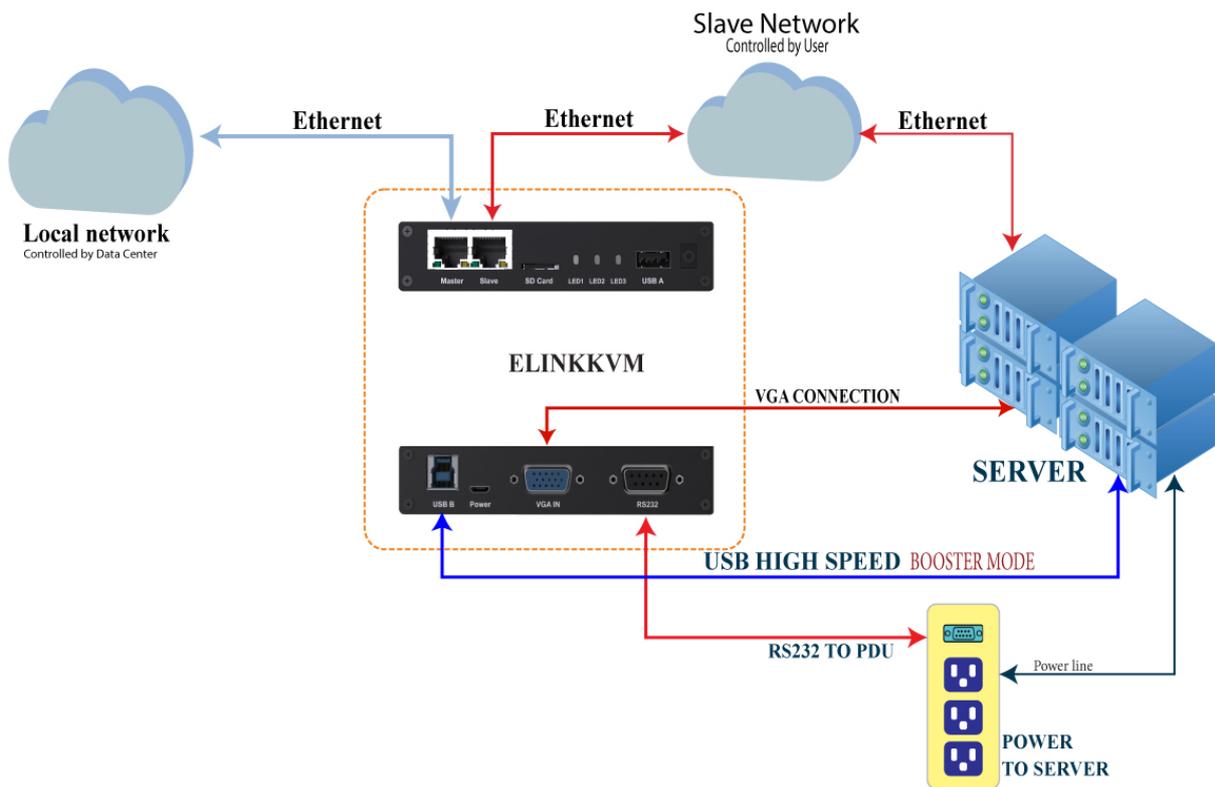


Figure 6. Separate network layer.

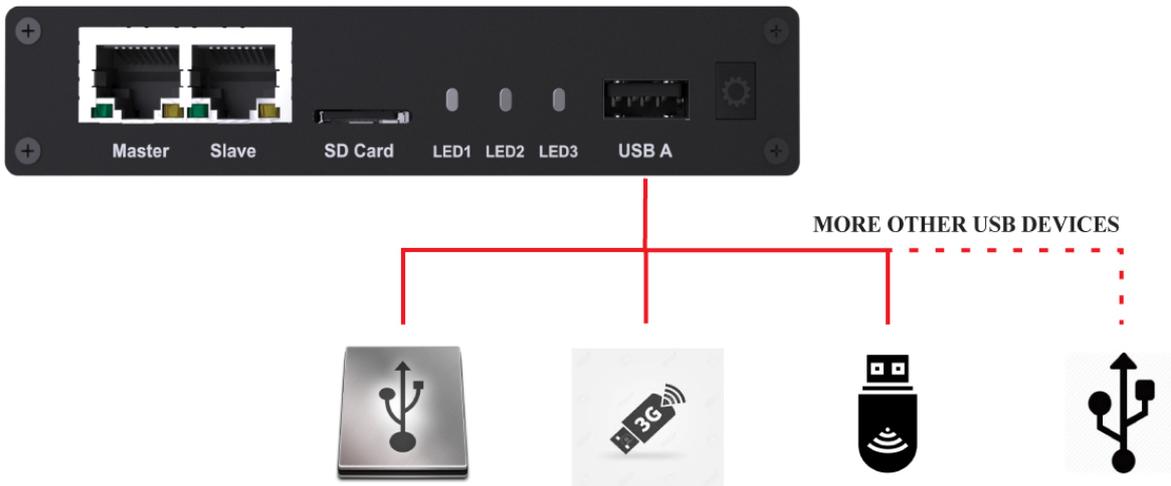


Figure 7. Extend function with USB device.

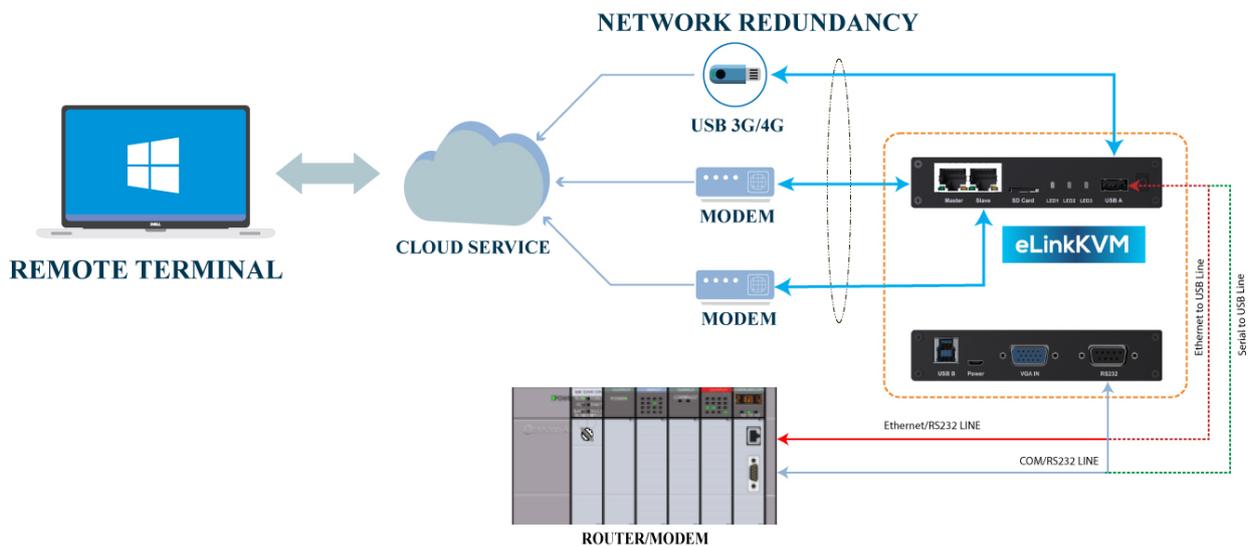


Figure 8. eLinkKVM configure network through multiple network.

## 2.5. Technical Specifications

### 2.5.1. eLinkKVM Hardware Specifications

Feature	Description
Network	2 Ethernet ports (10/100Mbps)
Storage	Internal Flash Storage (8GB) that can be accessed by eLinkViewer software
Interfaces	Micro SD-CARD (extend up to 64GB) RS232 interface USB Host interface to extend to USB 3G/4G, USB Wifi, USB Storage

Feature	Description
Video	VGA Port, Video Full HD Resolution  VGA Text mode: 640x350, 640x480, 720x400  VGA Graphic mode: 640x480, 800x600, 1024x768, 1152x864, 1280x1024, 1440x900,1680x1050,1600x1200,1920 x1080

### 2.5.2. eLinkKVM Mechanical Specifications

Feature	Description
Dimensions	119mm x 69mm x 24mm
Weight	200gr

### 2.5.3. eLinkKVM Firmware Specifications

Feature	Description
Remote Access Features	Supports VNC Protocol  Emulation of the USB Relative Mouse and Absolute Mouse  Emulation of the USB Keyboard  Captures Full HD resolution with Hardware Mode  Captures all resolutions with good screen quality using ELINKGATE patented Booster Mode  Emulation of Virtual Media (file .iso, .isox, .hdd, .hdd2 can be mounted as a USB 2.0 device)  Emulation of RS232 Port
Securities	Secures by SSL  Multiple Login Accounts (Up to 8 users)
Remote Server Power Control	Remote Reset/On/Off Server through an IPMI  GUI Interface from eLinkViewer  Controls the External PDU through RS232

### 2.5.4. eLinkViewer Software Specifications

**eLinkViewer** is an application specifically made to view remote screen content captured by an eLinkKVM device as well as interact with a remote computer.

Feature	Description
General	Supports multiple VNC Servers and elinkKVM  Remote protocol using VNC and extension command sets for elinkKVM  Supports video scaling and full screen
Automation	Supports image recognition and Python scripting allows automation of the activities
IPMI	Supports IPMI protocol to control server with GUI
Platform	Currently supports Windows
Bandwidth Optimization	Provides video quality settings to maximize remote performance for the available network bandwidth
Screen scaling	Viewer screen can be scaled to any resolution independent from remote computer's resolution or even enter a full-screen mode. In full-screen mode, the viewer screen scales according to the remote computer's resolution.

## 2.6. Components

Front and rear views of an elinkKVM device:



Figure 9. elinkKVM Front View

### 2.6.1. Front view

ID	Component	Description
1	Ethernet ports (10/100Mbps)	Master: can be configured as a DHCP server, a DHCP client or a static IP Slave: can be configured as a DHCP Client or a static IP
2	SD card	Allows a user to expand storage capacity with an external micro SD card
3	Status Leds 1,2,3	Signal device statuses through different led colors
4	USB Type-A	Allows attaching an external peripheral device such as USB 3G/4G, USB Mass Storage, etc., to extend elinkKVM functionality.

### 2.6.2. Rear view

ID	Component	Description
1	USB Type-B	Powers the device and acts as a communication channel between eLinkKVM and a server
2	Power	The secondary source is supplied by a micro USB cable. It is necessary when the power from USB Type B port is insufficient, or keeping the connection to eLinkKVM alive when the server is powered off.
3	VGA	For capturing and outputting video signal to send and receive the signal over the internet.
4	RS232	The serial communication port <b>COM</b> port. This port will be connected to the server serial port.

# CHAPTER 3. ELINKKVM QUICK START GUIDE

---

This chapter describes how to install the elinkKVM device and necessary software to work with elinkKVM . It contains the following sections:

- Package contents
- Power on elinkKVM
- elinkKVM in Setup Mode
- elinkKVM in Operating Mode
- Network setup
- elinkViewer setup

## 3.1. Package Contents

In addition to the **elinkKVM** device, the package contains the following items:

- 1 USB 2.0 Type-B Cable
- 1 VGA cable
- 1 RS232 cable (optional)
- 1 Micro USB cable
- 1 USB Power Adapter (5V - 2A)
- 1 Server Rack Mount Kit
- 1 elinkKVM Quick Start Guide \*

## 3.2. Power On elinkKVM

The USB-Powered design allows elinkKVM power directly from server USB port (1) without an external power. But in some cases, it is necessary to keep the device active while the server can be turned off or elinkKVM needs to provide the power to external USB device. This can be done by supplying another 5V power source to the micro USB port (2) on an elinkKVM device.



Figure 10. Rear USB ports for powering up an elinkKVM device

During the bootup process, the 3 LEDs will blink and turn green when the device is ready. Now, the user can start using the device.

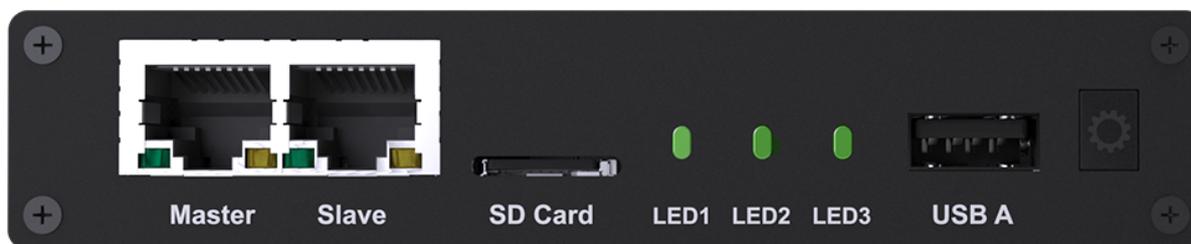


Figure 11. Front status LEDs

LED ID	Color	Description
1	Green	The system is initialized successfully
2	Orange	The network master port is ready. The led will turn green when there is working connection on the port.
3	Orange	The network slave port is ready. The led will turn green when there is working connection on the port.

### 3.3. eLinkKVM in Setup Mode

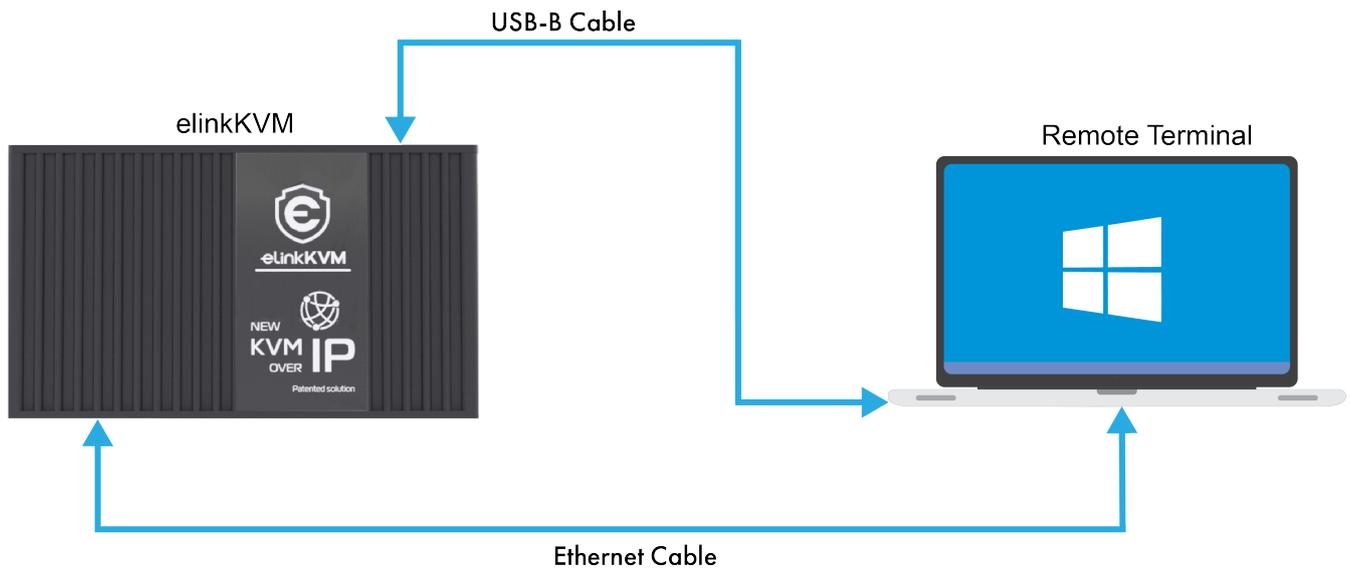


Figure 12. eLinkKVM connection in Setup mode

The eLinkKVM is designed to be just plug and play. User can simply install the device in operation mode and start using immediately. But, there are some cases when the auto DHCP IP address may not work. The user may have to setup it manually. Below are the connection diagrams when setup eLinkKVM.

Perform following steps to configure eLinkKVM in Setup Mode:

1. Connect Ethernet RJ45 cable between eLinkKVM and remote terminal computer.
2. Connect USB Type-B cable to remote terminal computer and Type-B port of eLinkKVM device.

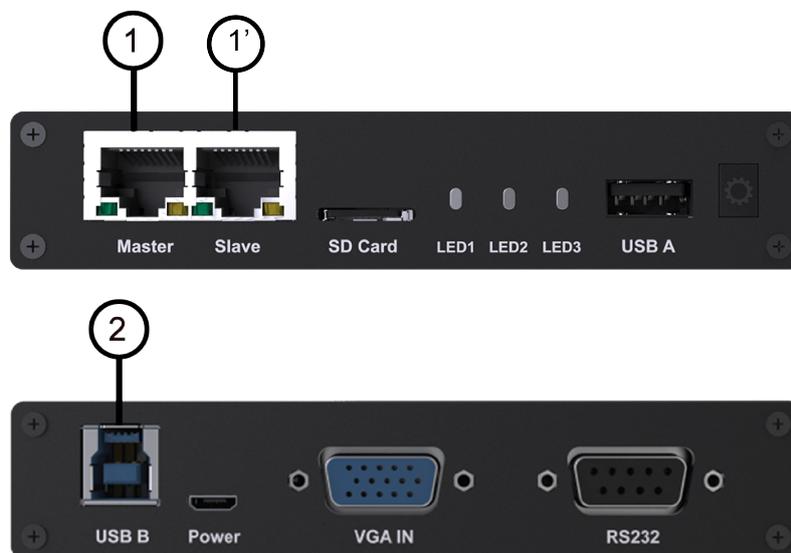


Figure 13. Connection of an eLinkKVM device in Setup Mode

### 3.4. eLinkKVM in Basic Operation Mode

The main usage of eLinkKVM is to remotely monitor a server. The basic connection is described below, where eLinkKVM is connected to Server through USB cable and VGA. It is also connected to remote terminal through Ethernet.

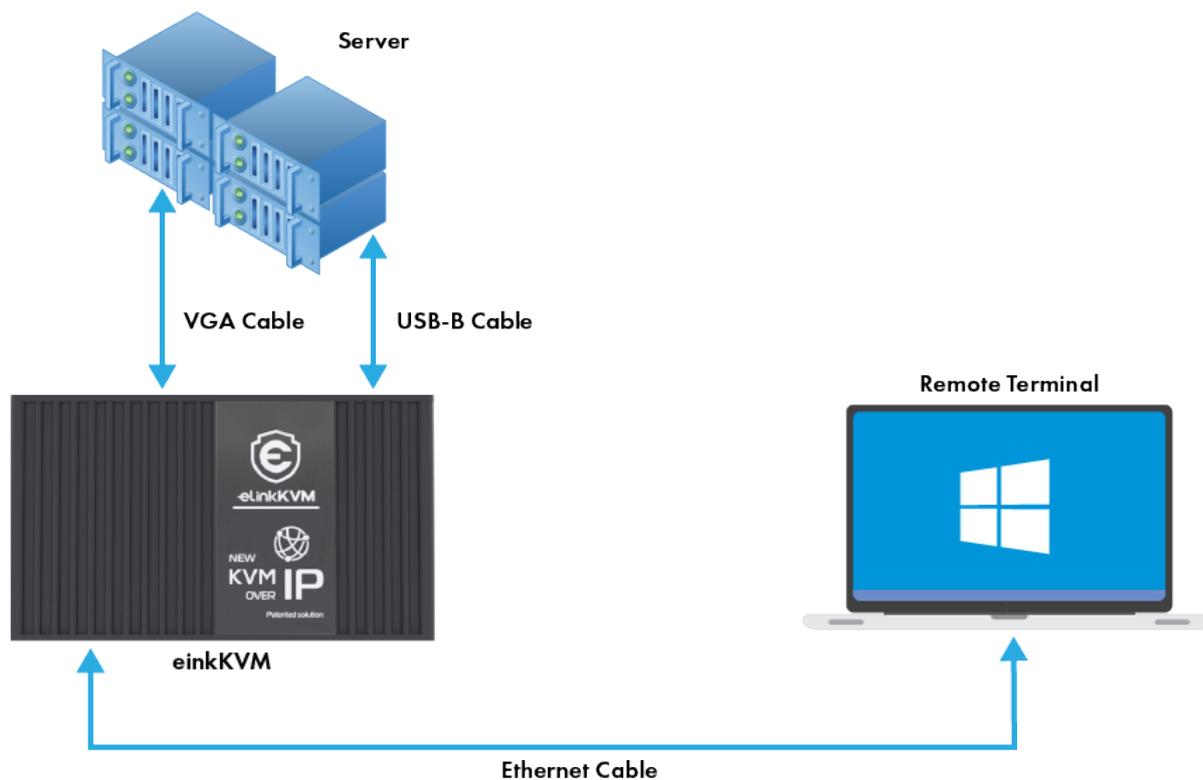


Figure 14. Basic eLinkKVM Operation Mode

Perform following steps to configure eLinkKVM:

1. Connect Ethernet RJ45 cable between eLinkKVM and remote terminal computer.
2. Connect VGA cable between Server and VGA port on eLinkKVM.
3. Connect USB Type-B cable between Server USB port and Type-B port of eLinkKVM device.

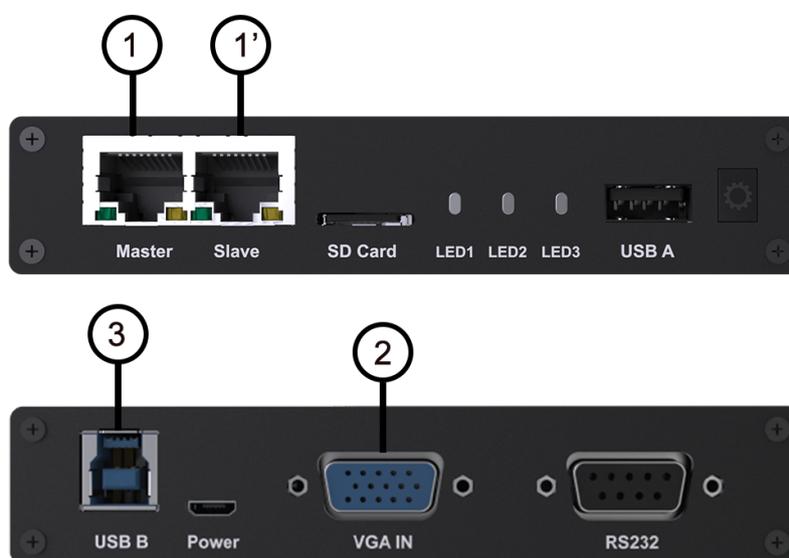


Figure 15. eLinkKVM connection (basic operation mode)

## 3.5. Network configuration

elinkKVM supports two Ethernet ports for network connection:

- Ethernet Master: by default, configured as a DHCP server.
- Ethernet Slave: by default, configured as a DHCP client.

In order to connect to an elinkKVM device, a network connection between a remote terminal and an eLinkKVM device is required. The connection can either be:

- *Direct connect*: An Ethernet cable connects directly between a remote terminal and an elinkKVM device. In this case, we should configure elinkKVM as DHCP server (using Master port) and remote terminal as DHCP client.
- *Local Area Network*: Both the remote terminal and an elinkKVM device are connected to a local area network (LAN). eLinkKVM is plugged to the router of the network. In this case we should configure elinkKVM (using Slave port) and remote terminal as DHCP client
- *Over the Internet*: an elinkKVM device is provided a static address for a remote terminal to access it anywhere in the world.

When a connection is established, the elinkKVM is accessed from remote terminal by the software eLinkViewer. Currently, eLinkViewer is only supporting Windows.

## 3.6. Setup eLinkViewer

Download the eLinkViewer setup package from link: [www.elinkgate.com/support/download.html](http://www.elinkgate.com/support/download.html).

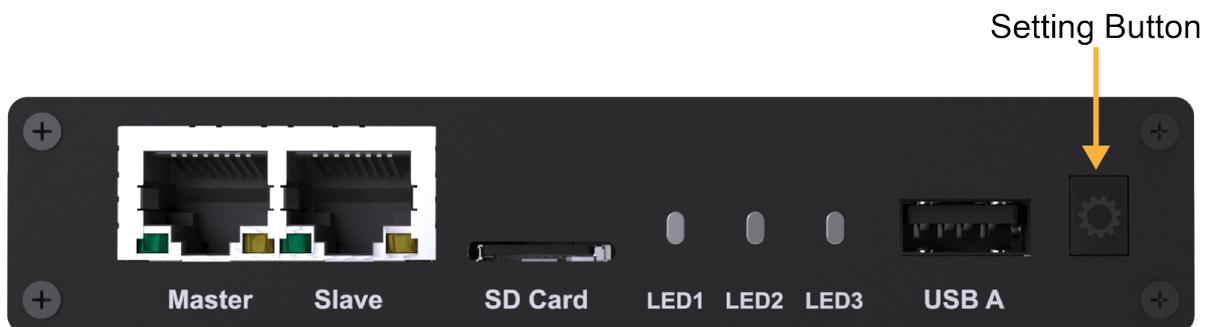


Figure 16. Setting Button of elinkKVM

Optionally, the elinkKVM comes with pre-production setup package of eLinkViewer. When LED1 of elinkKVM turns green, short pressing the Setting Button will result elinkKVM to appear as storage to attached computer. The setup.exe can be found in the attached storage at folder elinkViewer\setup.exe

### 3.7. elinkViewer quick start

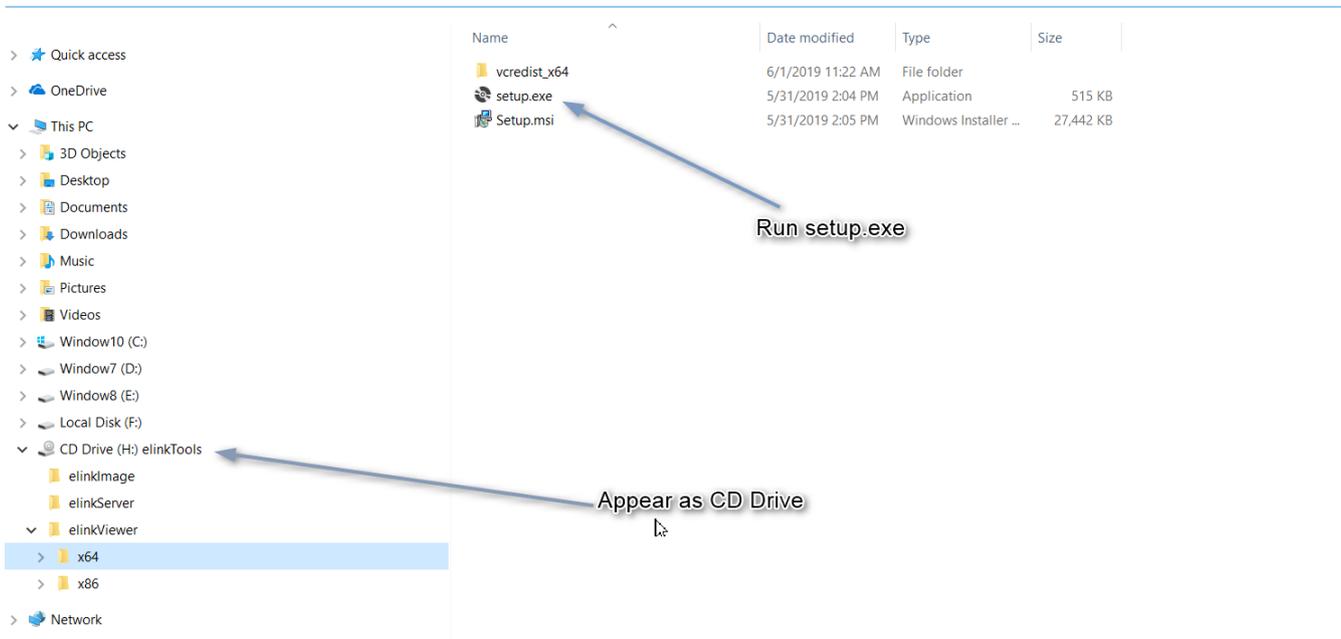


Figure 17. The contents of attached storage

Just run setup.exe and follow the instructions to complete the setup.

Section 3.7 chapter shows how to run and use eLinkViewer.

## 3.7. elinkViewer quick start

1. Run elinkViewer.exe by double clicking the application icon. The login dialog appears as below.

If user knows the elinkKVM address, then just fill in the address and move to connect step. Or click on **Scan** button to pop up Service Discovery Dialog to find the elinkKVM.

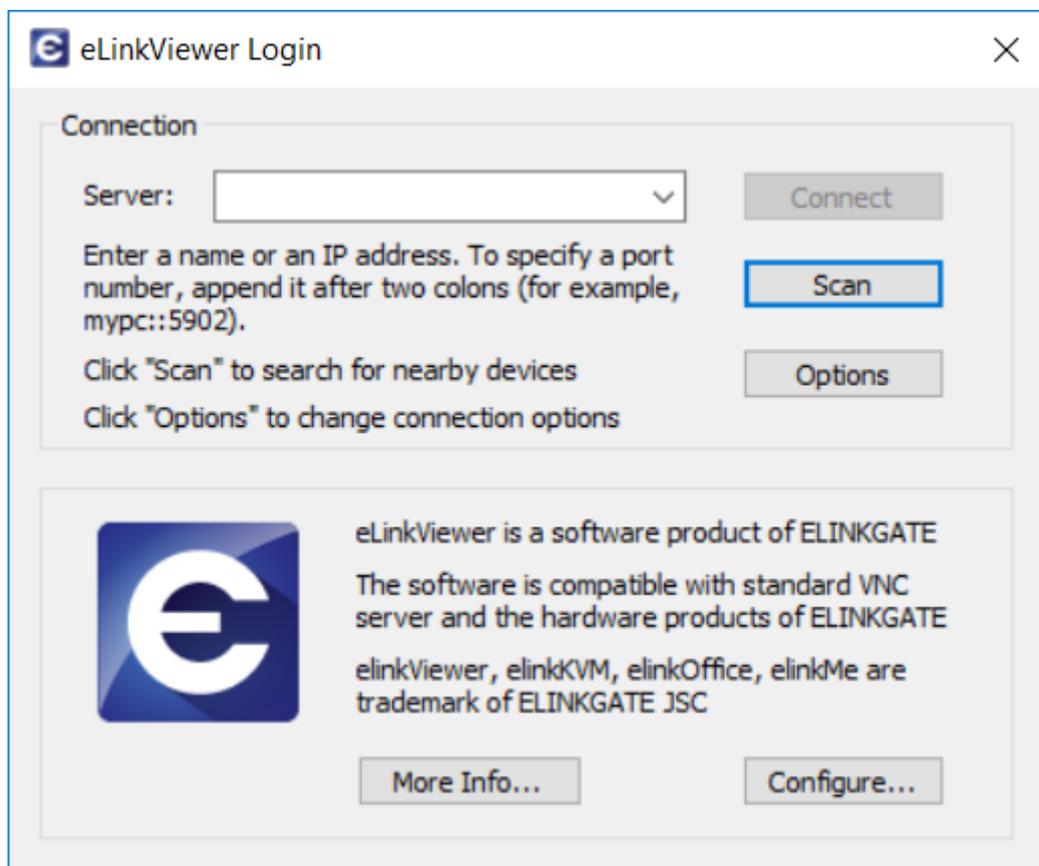


Figure 18. Establish new connection

2. Click on **Scan** button to search for device.

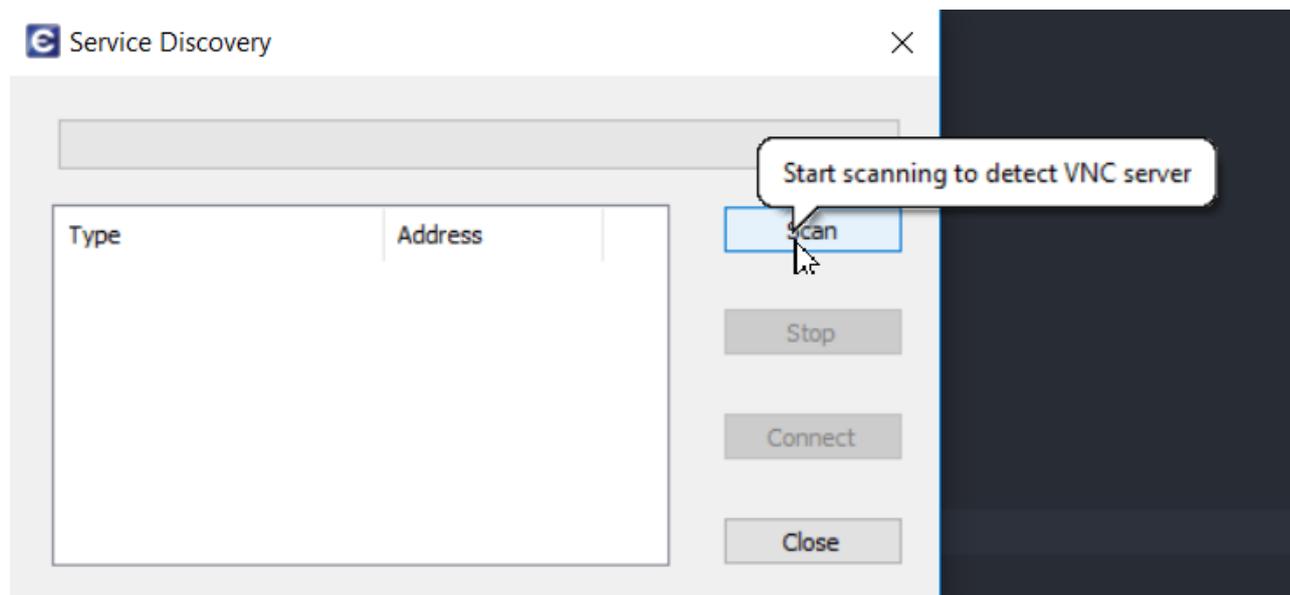


Figure 19. Scan for elinkKVM devices

3. Click **Connect** button to connect to the device

When device is detected, it will appear on the dialog. Select the device and click **Connect** button.

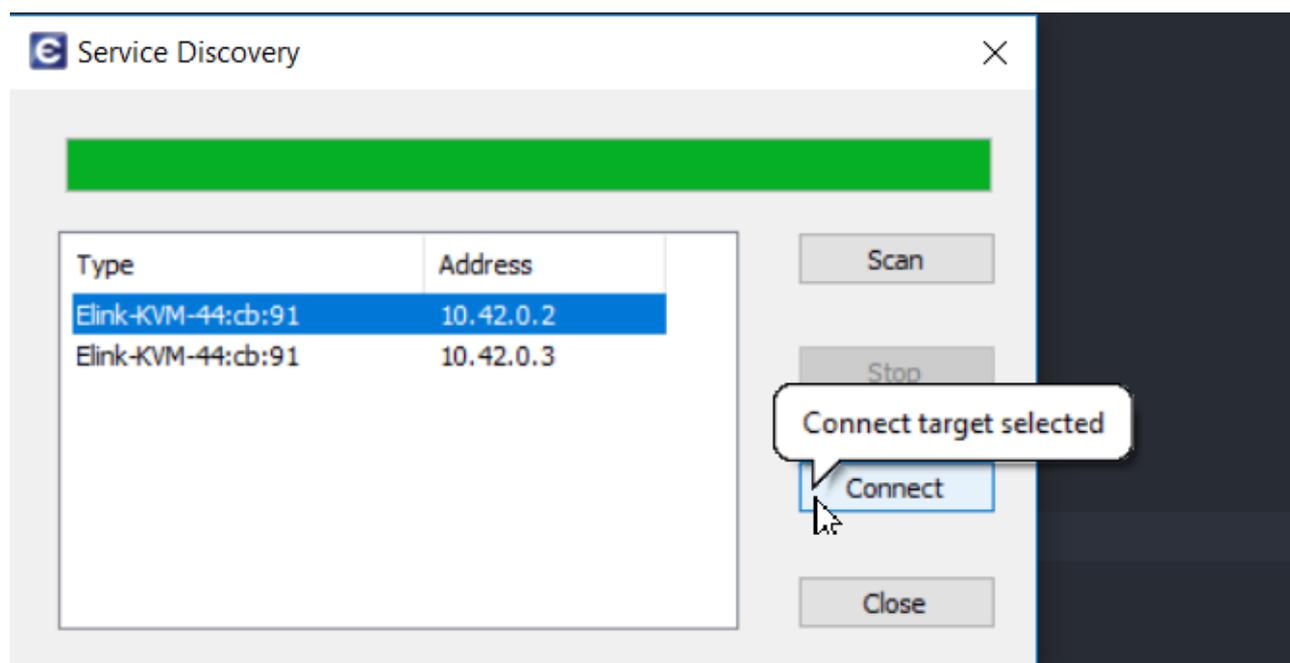


Figure 20. Select the device and click "Connect" button

4. Get the IP address and connect

When the IP address is filled either manually or as a result of scan operation above, user can click on **Connect** button to establish the connection with elinkKVM.

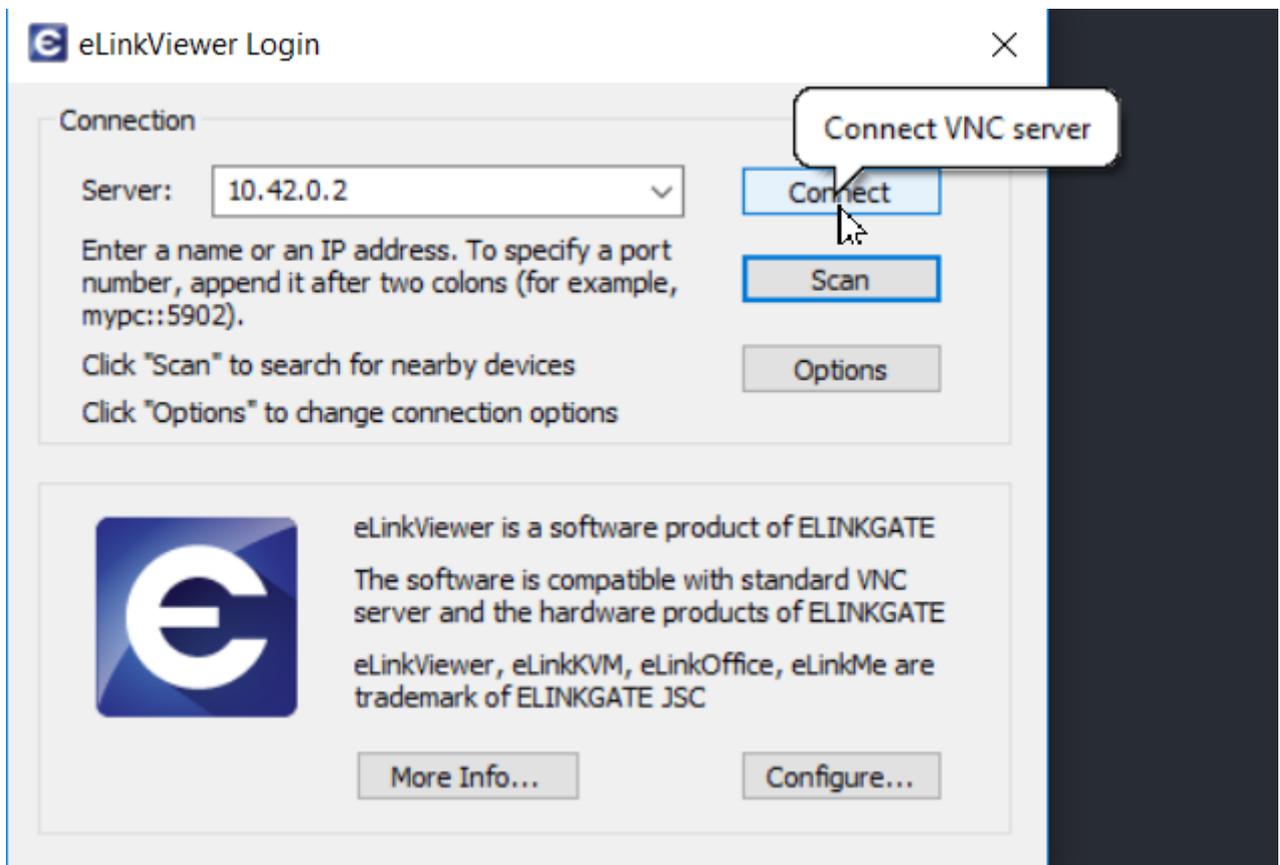


Figure 21. Establish new connection

5. Authenticate with device

When connected to eLinkKVM, an authentication dialog will pop up to ask for username and password. The default username and password is **admin/admin**.

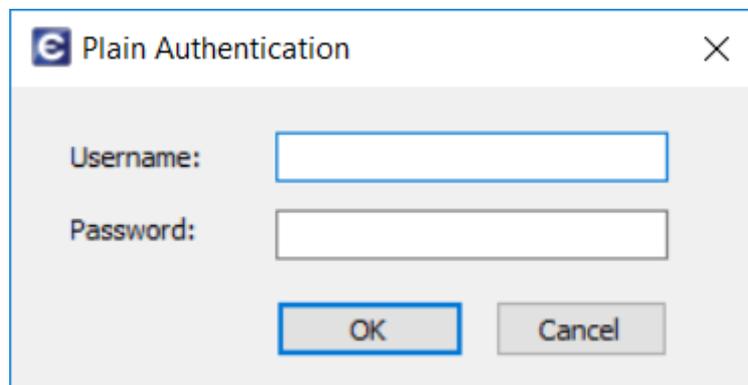


Figure 22. eLinkKVM Authentication UI

6. Switch to Local Screen.

When first connect to eLinkKVM, the last used screen will appear. The default last used screen is **local screen**, which shows the status and setting of eLinkKVM. If the last used screen is not **local screen**, user can quickly switch to by clicking on **local screen** icon of quick tool bar.





Figure 23. Switch to Local Screen

### 7. Switch to VGA Screen

If eLinkKVM is connected in operation mode, user can see the Server screen by clicking on **VGA screen** icon of quick tool bar 

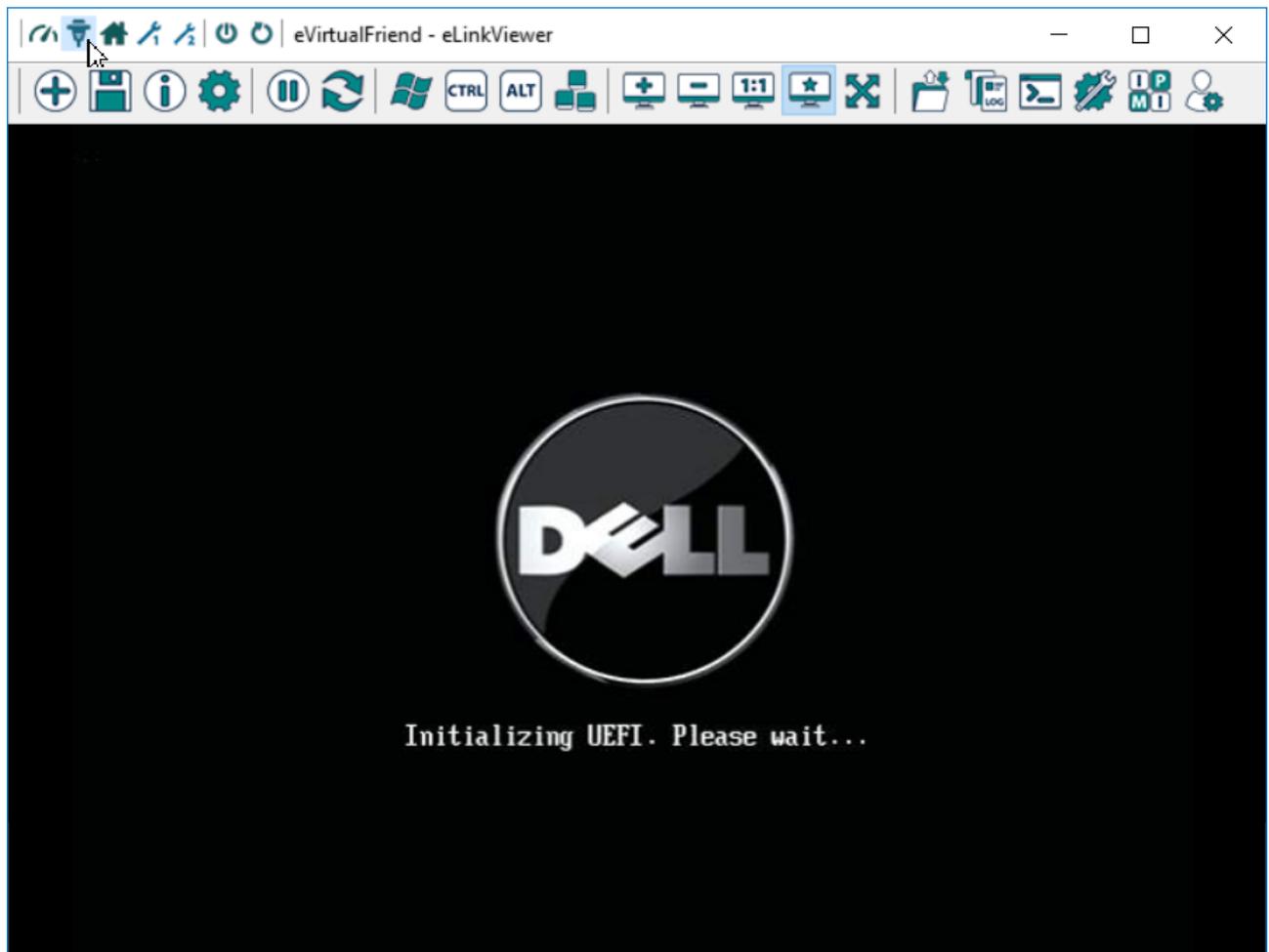


Figure 24. Switch to VGA Screen

For more detailed operation of eLinkKVM, please see [eLinkKVM User Manual](#).

## CHAPTER 4. ELINKVIEWER

eLinkViewer is a specialized software for remote control and management of eLinkKVM devices. **Booster** technology on eLinkKVM is only available while using eLinkViewer.

There are 3 main scenarios while using eLinkViewer:

- Login to eLinkKVM
- Access the remote computer
- Running python script

### 4.1. Login to eLinkKVM

#### 4.1.1. Establish connection

While initially opening **eLinkViewer** by double clicking on the application icon or running at command line the below **login** dialog is presented:

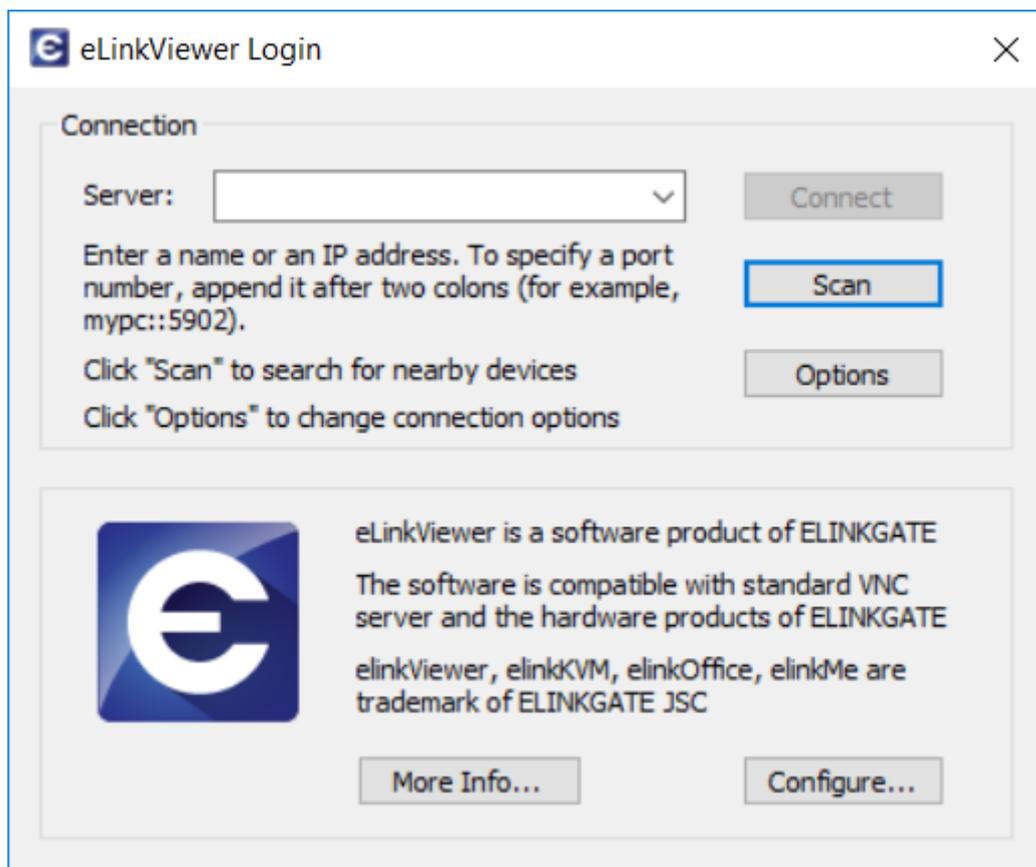


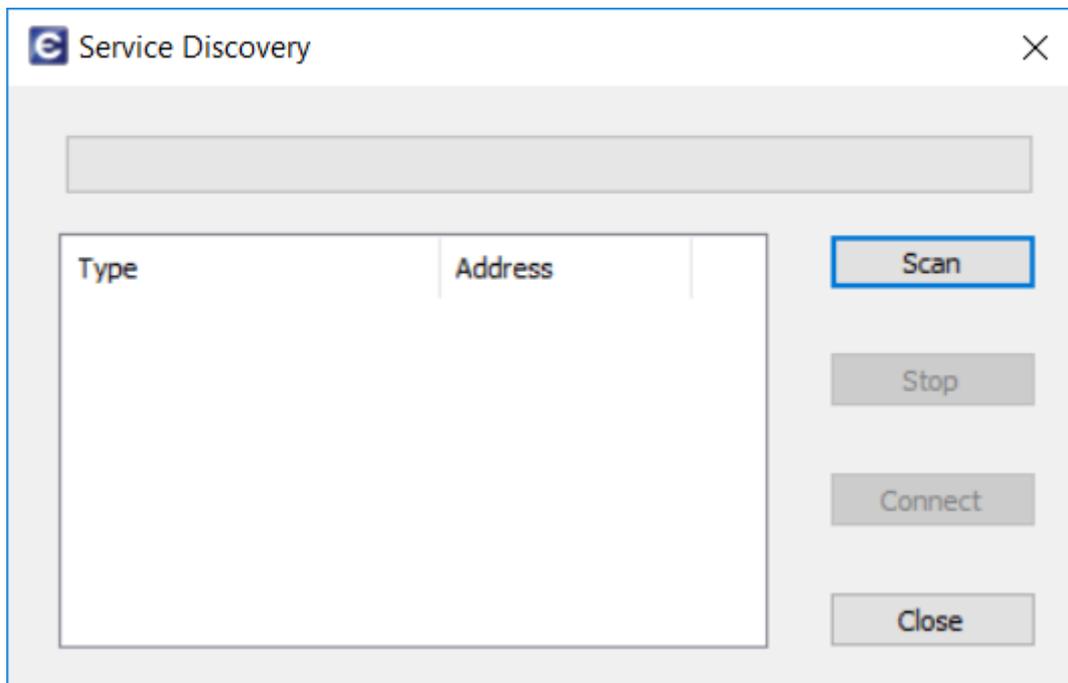
Figure 25. eLinkViewer quick connected UI

- Click on **Connect** button: to connect to eLinkKVM, fill the configured IP address in the text box next to the **Server** field and click **Connect** to connect eLinkViewer to the eLinkKVM at the configured IP address:
- Click on **Options** button: configure eLinkKVM-VNC connection by encoding type, compression level or control restrictions...
- Click on **Scan** button: scan for all eLinkKVM devices in the current local area network (LAN).

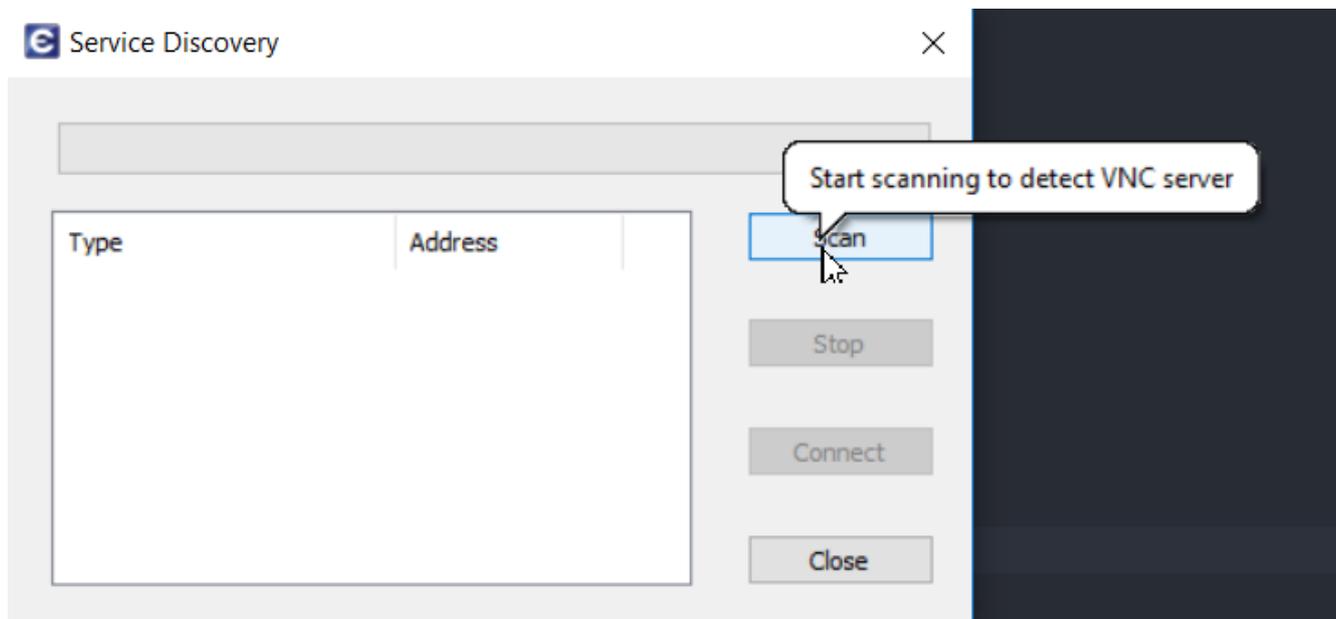
- Click on **More info** button: to get more info about elinkViewer such as copyright, version number, build date...
- Click on **Configure** button: configure GUI for elinkViewer such as show/hide toolbar, keep or clear connection history...

### 4.1.2. Scan for elinkKVM

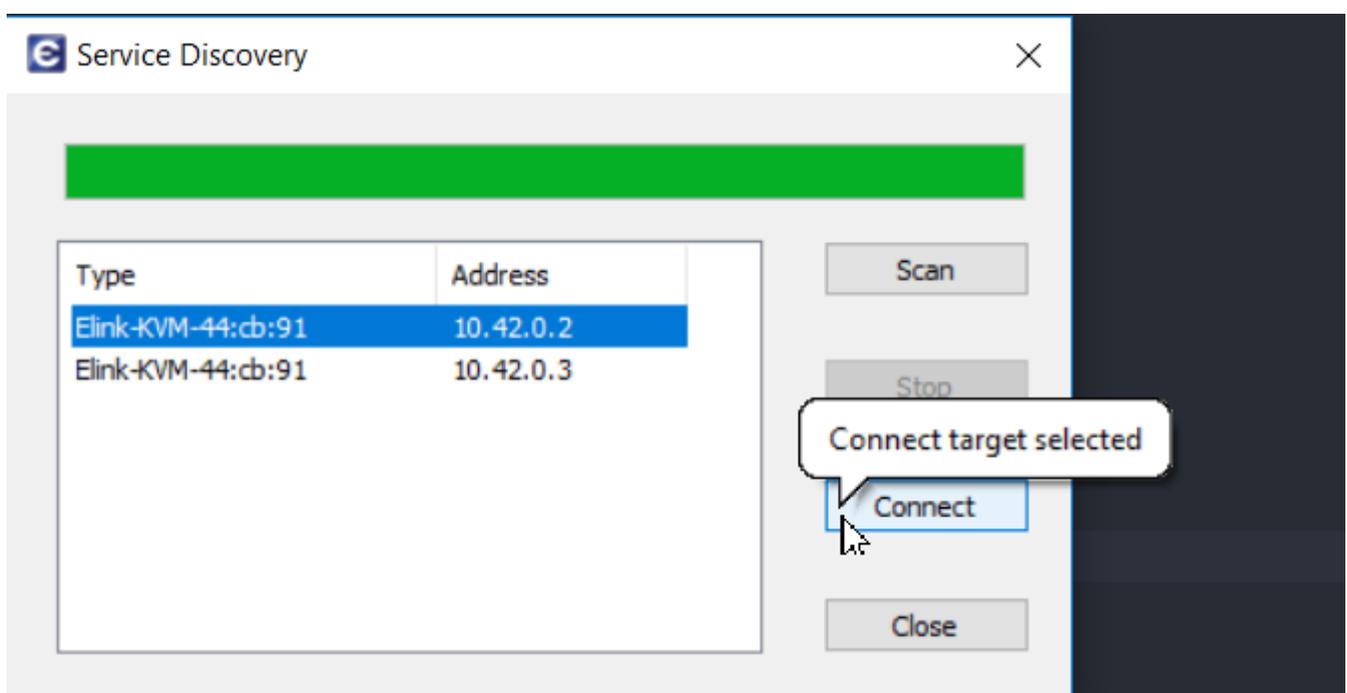
User can scan the local network to find all available elinkKVM. By clicking on **Scan** button of login dialog below dialog will pop up



Continuous clicking on **Scan** button will start to find the devices



The found devices will be listed on dialog by clicking on the selected device then click **connect**. The IP address of device will be filled in **elinkViewer Login** dialog.



### 4.1.3. Device authentication

After clicking on **Connect** button, the viewer will establish a new connection to elinkKVM. Depending on the security setting, the device may require an authentication process, where a dialog will pop up to ask for username/password

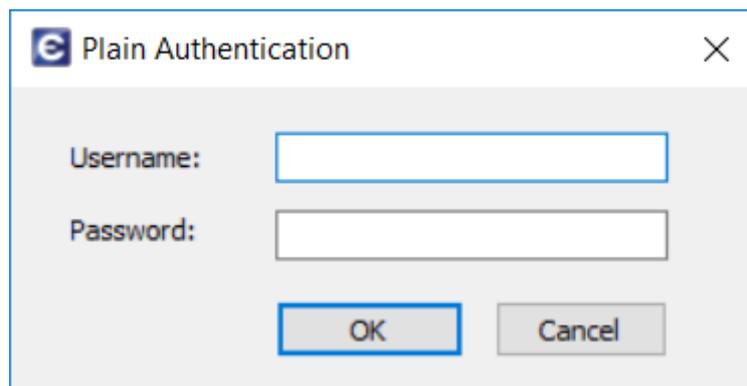
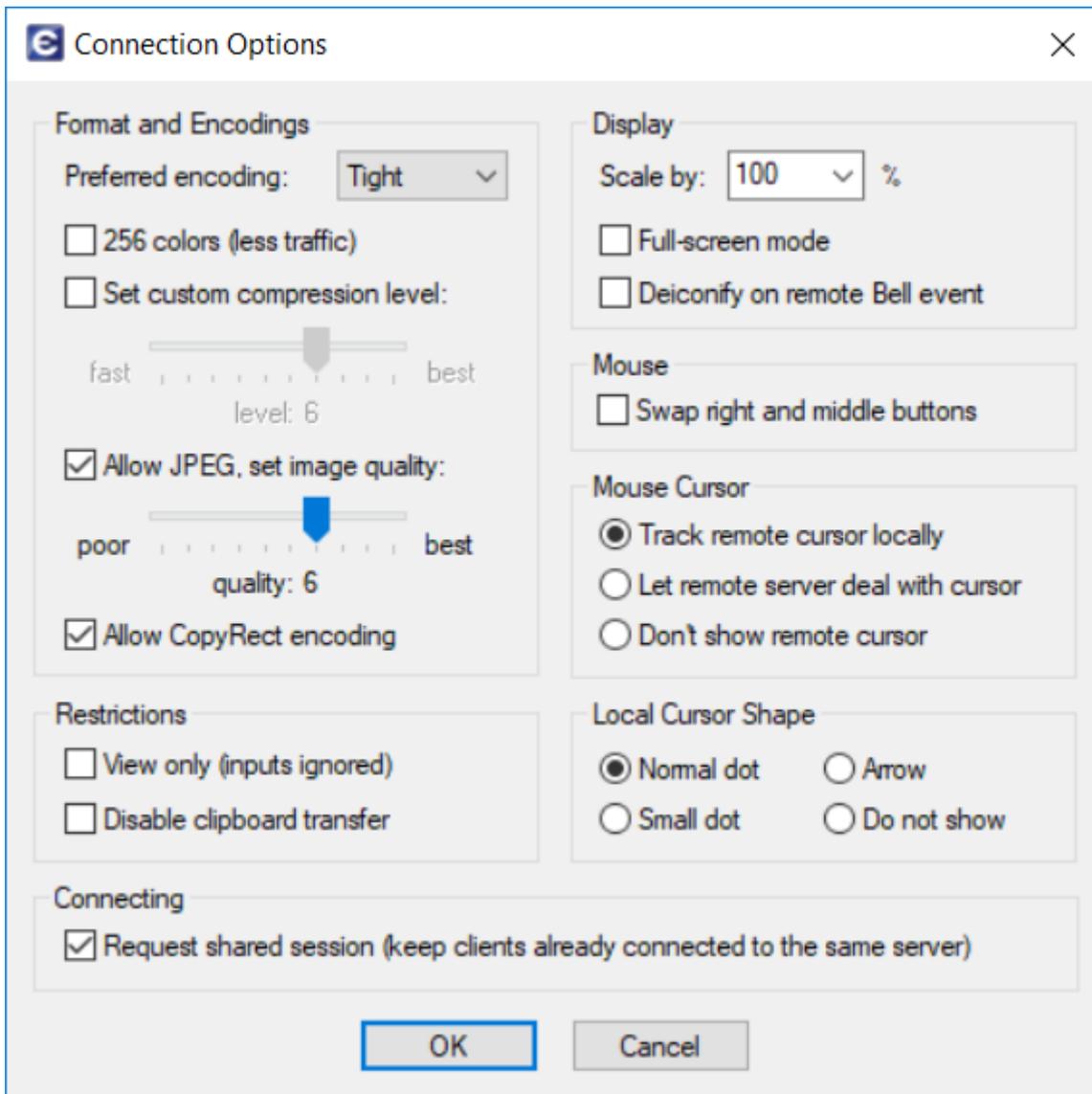


Figure 26. elinkViewer login UI

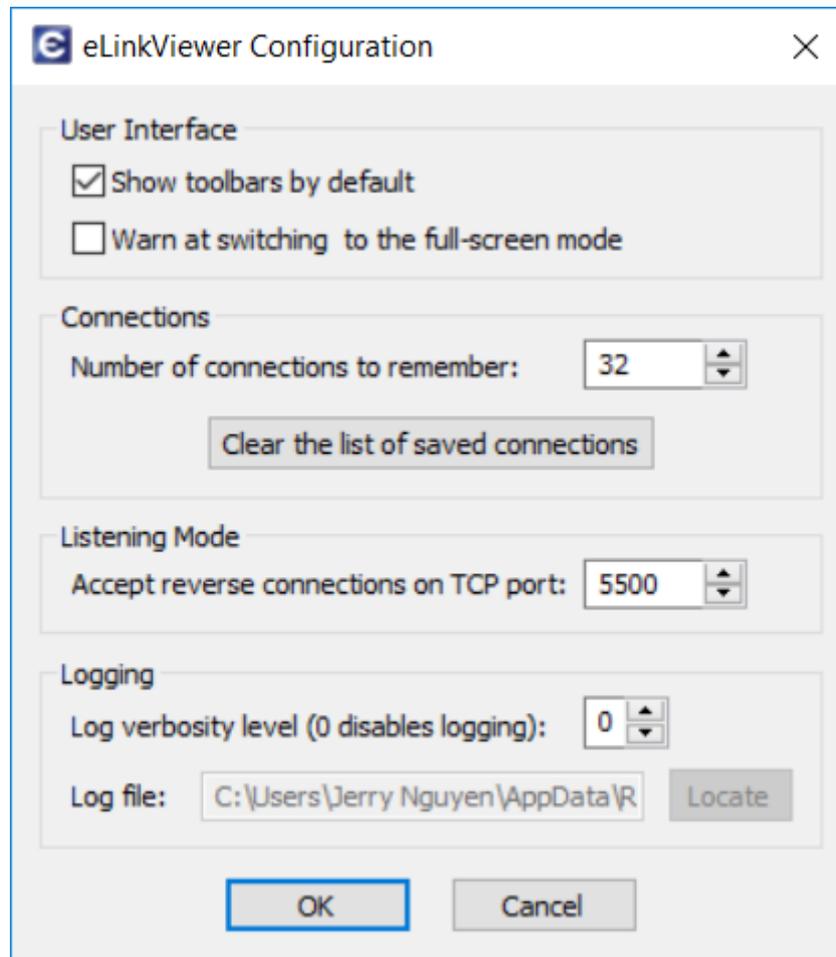
### 4.1.4. Change Connection Options

Before connecting to elinkKVM, user can change the connection setting by clicking on **Options** button. A below dialog will pop up to allow user to change encoding types, compression level, restriction of remote control...Just change the setting and click **OK** button to apply the changes.



#### 4.1.5. elinkViewer Configuration

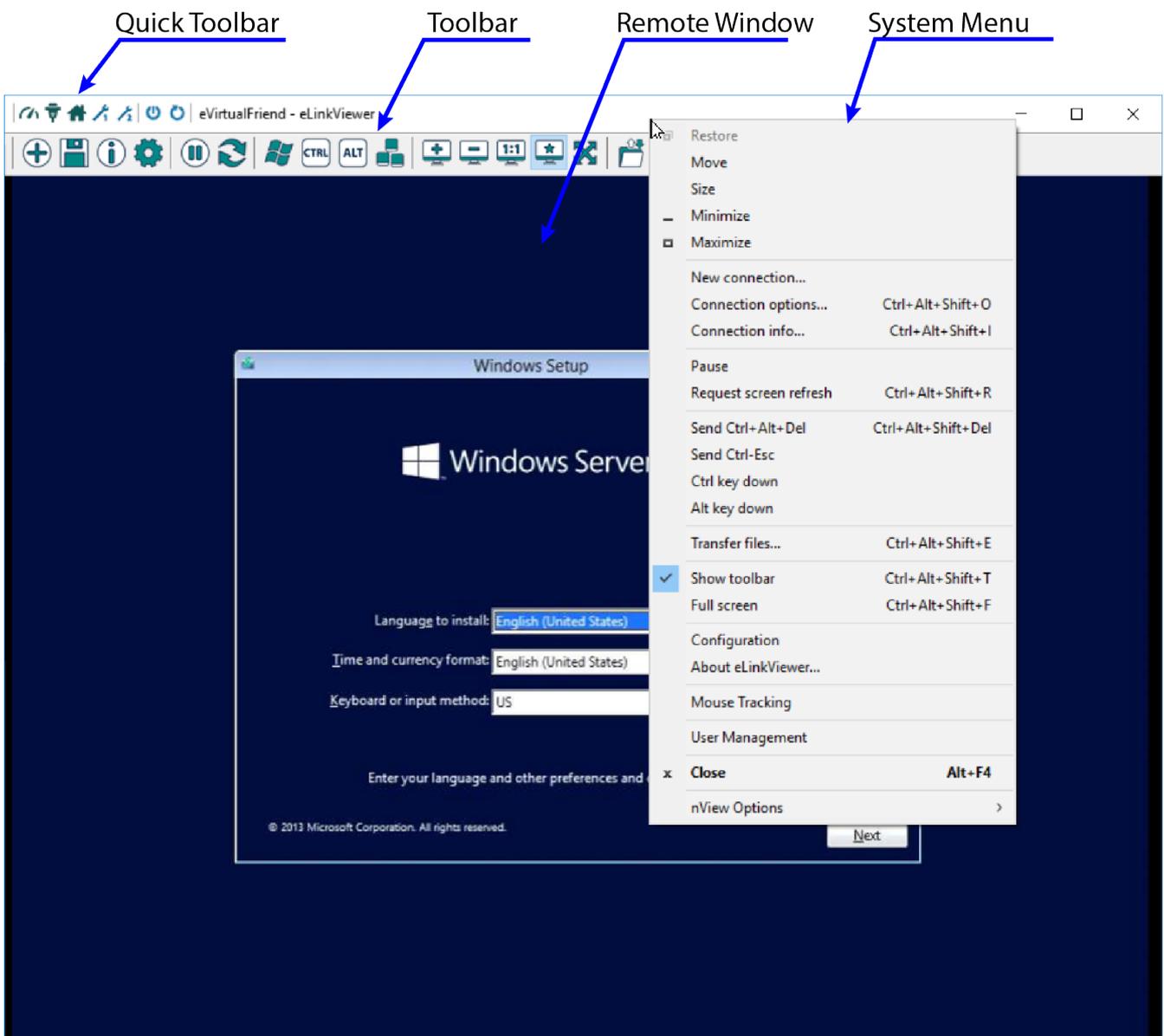
elinkViewer will store the GUI setting such as window size, position, logging option... of remote access session based on IP address of connected device. User may want to change it before connecting again. By clicking on **Configure** button, a below dialog will pop up. Just change the setting and click **OK** to apply the changes.



## 4.2. Access Remote Computer

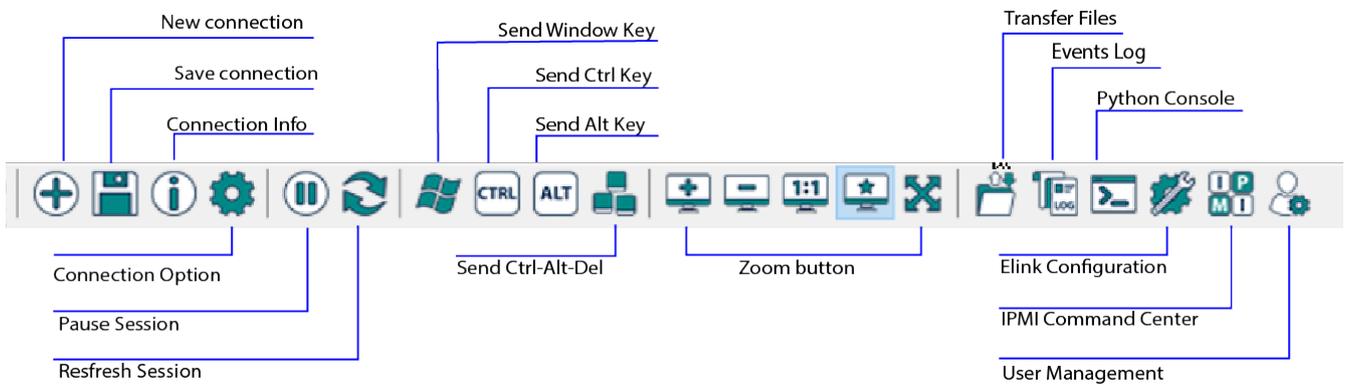
### 4.2.1. Remote Access GUI Layout

After connection is established, eLinkViewer will display GUI to access the remote computer. The layout of main components is shown below:

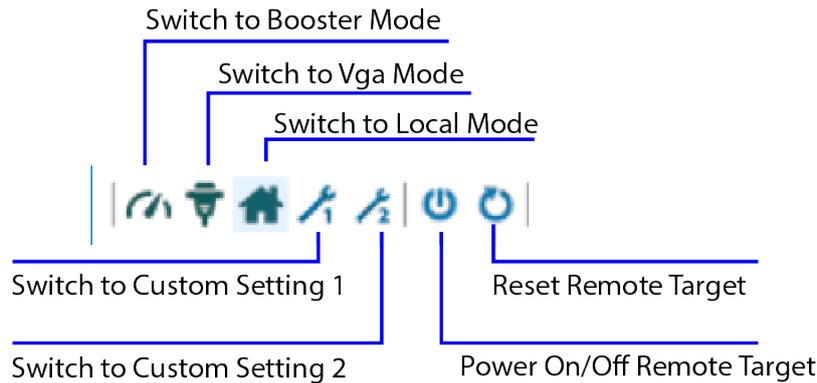


- **Quick Toolbar:** the quick buttons to change remote access mode and power on/off or reset remote target
- **Toolbar:** the buttons to change the setting to remote access session such as zoom in/out, sending special keyboard code, IPMI, user managements...
- **System Menu:** instead of using toolbar, user can select the functionalities on system menu by right click on Application Title Bar.
- **Remote Window:** will display the screen of remote target. When user focuses on this window, all keys event's or mouse events operation on the window will be sent and emulated on the remote target.

Below diagram explains the functionalities of buttons on Toolbar



Below diagram explains the functionalities of buttons on Quick Toolbar



#### 4.2.2. elink Configuration GUI

As evolution of traditional KVM (**K**eyboard,**V**ideo and **M**ouse) device, the main functionality of elinkKVM and the software elinkViewer is to allow user emulate the keyboard and mouse and capture the video data of the remote server. There are few options for user to select.

*The keyboard emulation modes:*

ID	Keyboard Mode	Description
1	USB HID	Emulate as physical USB Keyboard
2	VNC HID	Emulate as software keyboard, this will require an agent software running on targeted server

*The mouse emulation modes:*

ID	Mouse Mode	Description
1	USB HID	Emulate as physical USB Mouse
2	USB ABS HID	Emulate as physical absolute USB Mouse
3	VNC HID	Emulate as software mouse, this will require an agent software running on targeted server

*The video capture mode:*

ID	Video Capture Mode	Description
1	<b>LOCAL</b>	An internal video screen to show the internal status of eLinkKVM and configuration page
2	<b>VGA</b>	The video data is captured from VGA port
3	<b>BOOSTER</b>	The video data is captured from an agent software

The *virtual media feature*: In addition, the user can also select disk images (.iso,.hdd2...) to emulate a **Virtual Media** as it appears locally on targeted server. User can select up to 4 disk images.

The **eLink Configuration** dialog allows user change those settings and quick setting buttons on Quick Toolbar.

By clicking on the **eLink Configuration** button , below dialog will pop up.

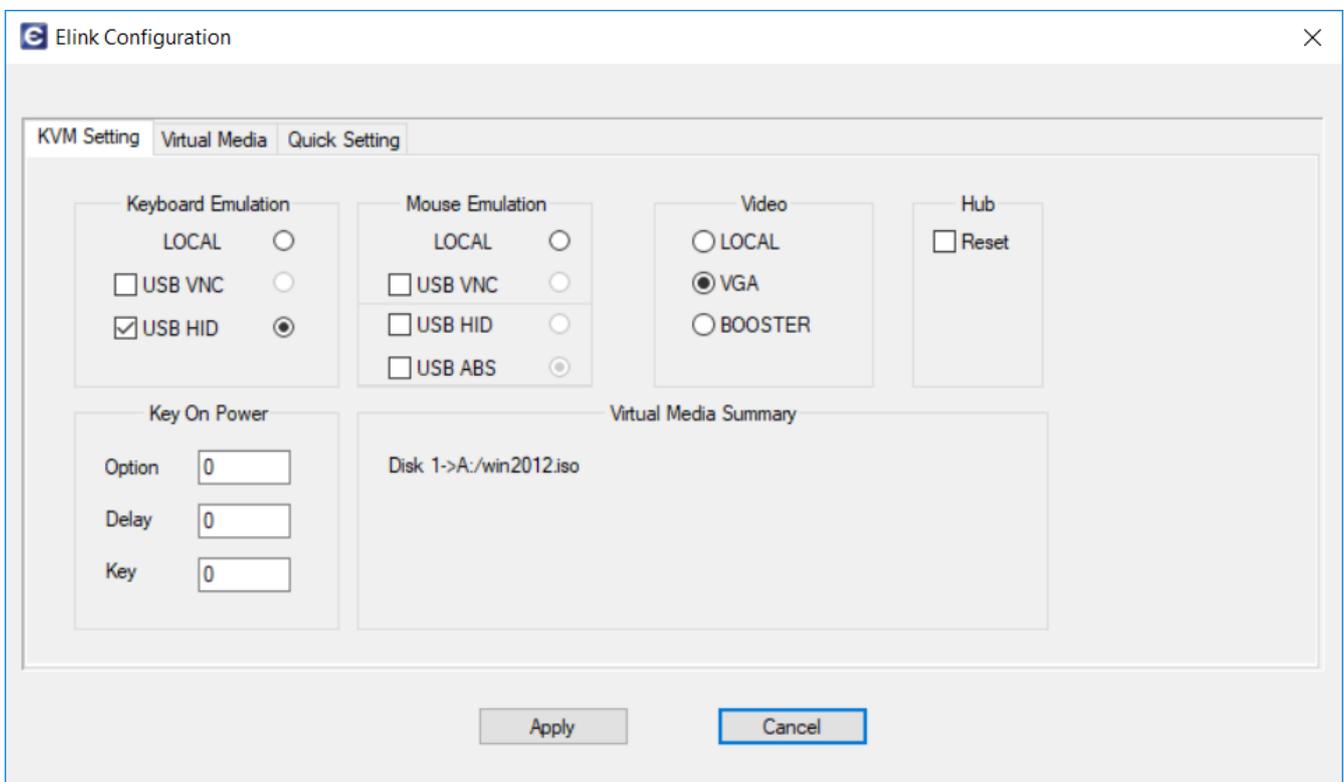


Figure 27. elink Configuration Dialog

**KVM Setting**

To change the KVM setting, select **KVM Setting** tab, change accordingly keyboard, mouse, video... then click **Apply** button.

Optionally, user can request elinkKVM to emulate a hub reset activity by checking **reset** in Hub.



User can configure elinkKVM to send a specific key when the remote server power is on the USB port by inputting the key code, delay time, option in **Key On Power** text box.

- Option: 0 - disable Key On Power, 1- enable Key on Power
- Delay: the delay time in millisecond after the USB resets, the key will be sent.
- Key: the HID key code will be sent.

### Virtual Media

To change the setting of virtual media select **Virtual Media** tab and change the path to the image files (.iso, .hdd, .hdd2 ...) then click **Apply** button to emulate the virtual disks.

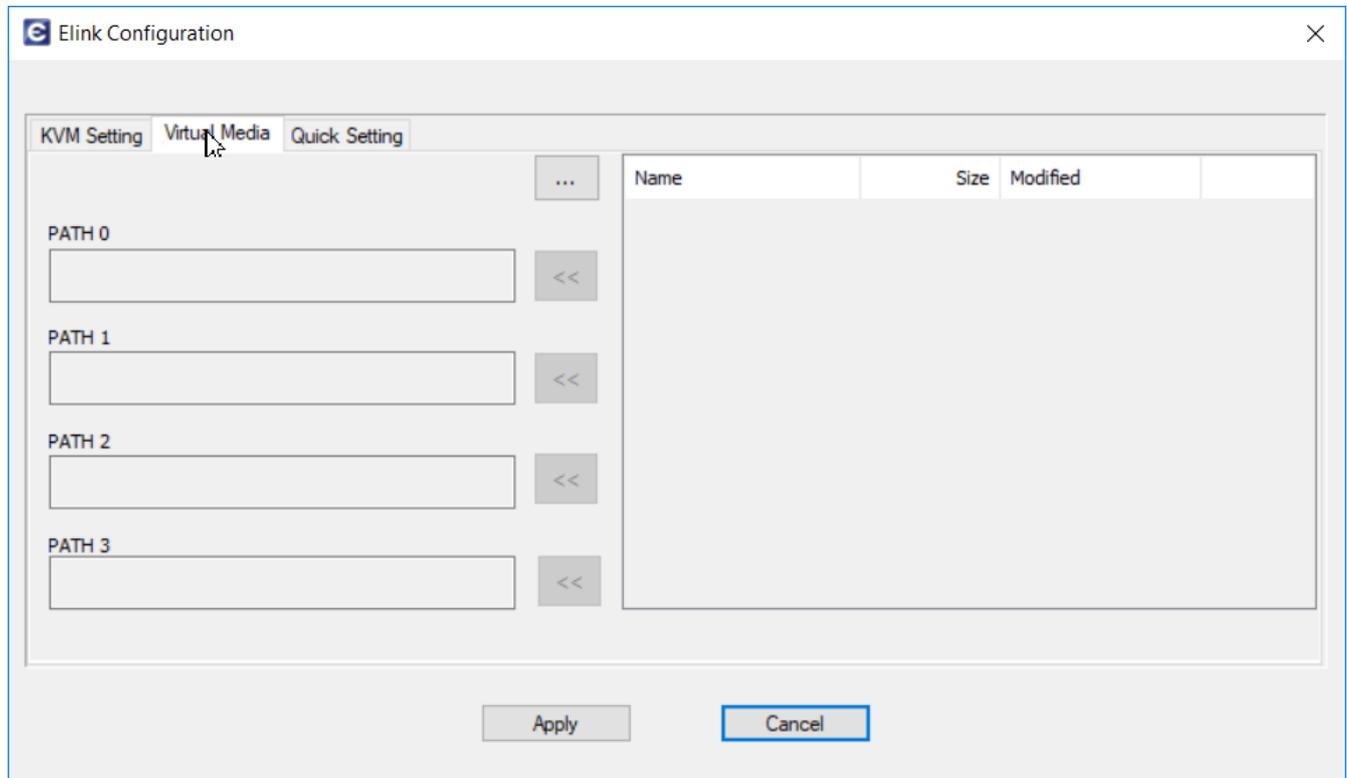


Figure 28. elink Configuration (Virtual Media) Tab

User can browse the image files, which were stored locally in elinkKVM, by clicking on , then local files will be shown on the list box. Select the image file, and then click  to apply to the appropriate PATH

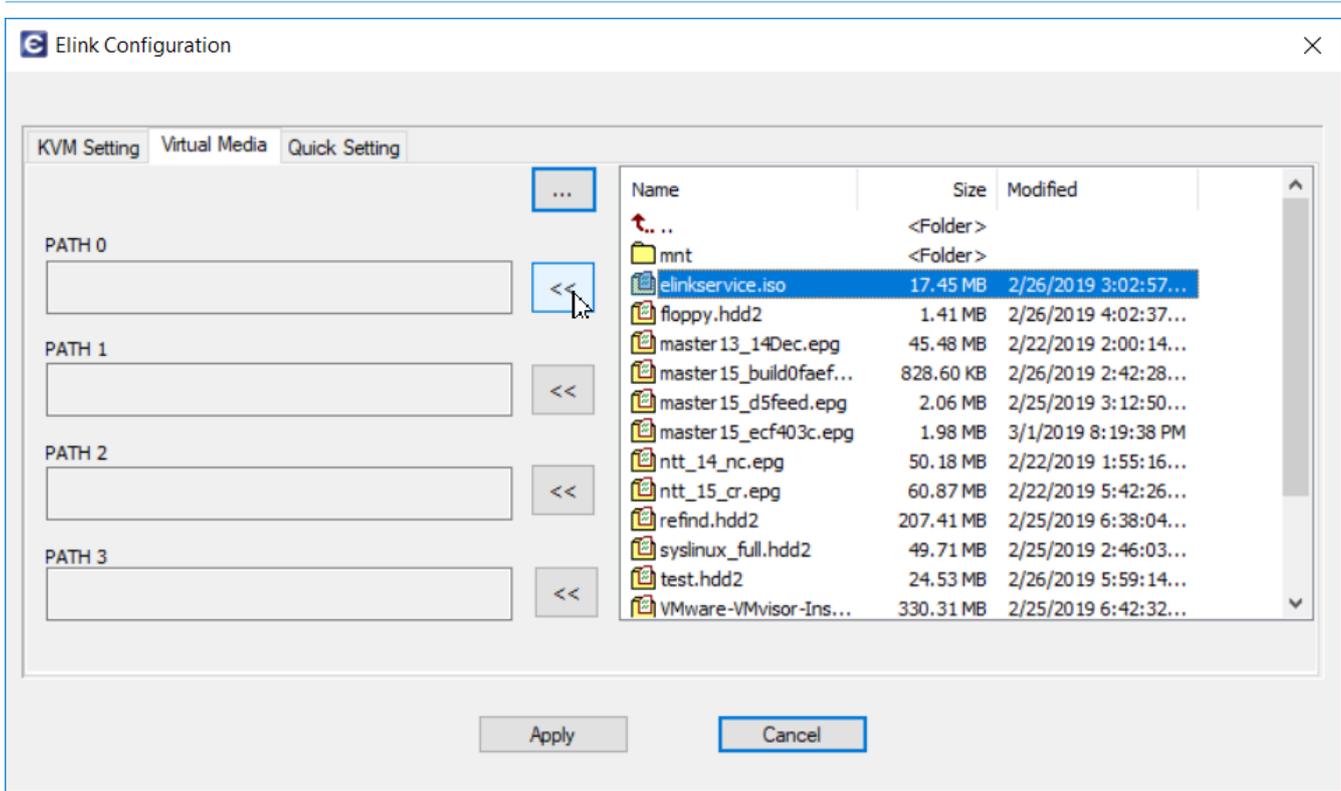


Figure 29. Select media into Path

### Quick Setting

To change the quick setting for button on Quick Toolbar select **Quick Setting** tab and change mode accordingly, and then click **Apply** button.

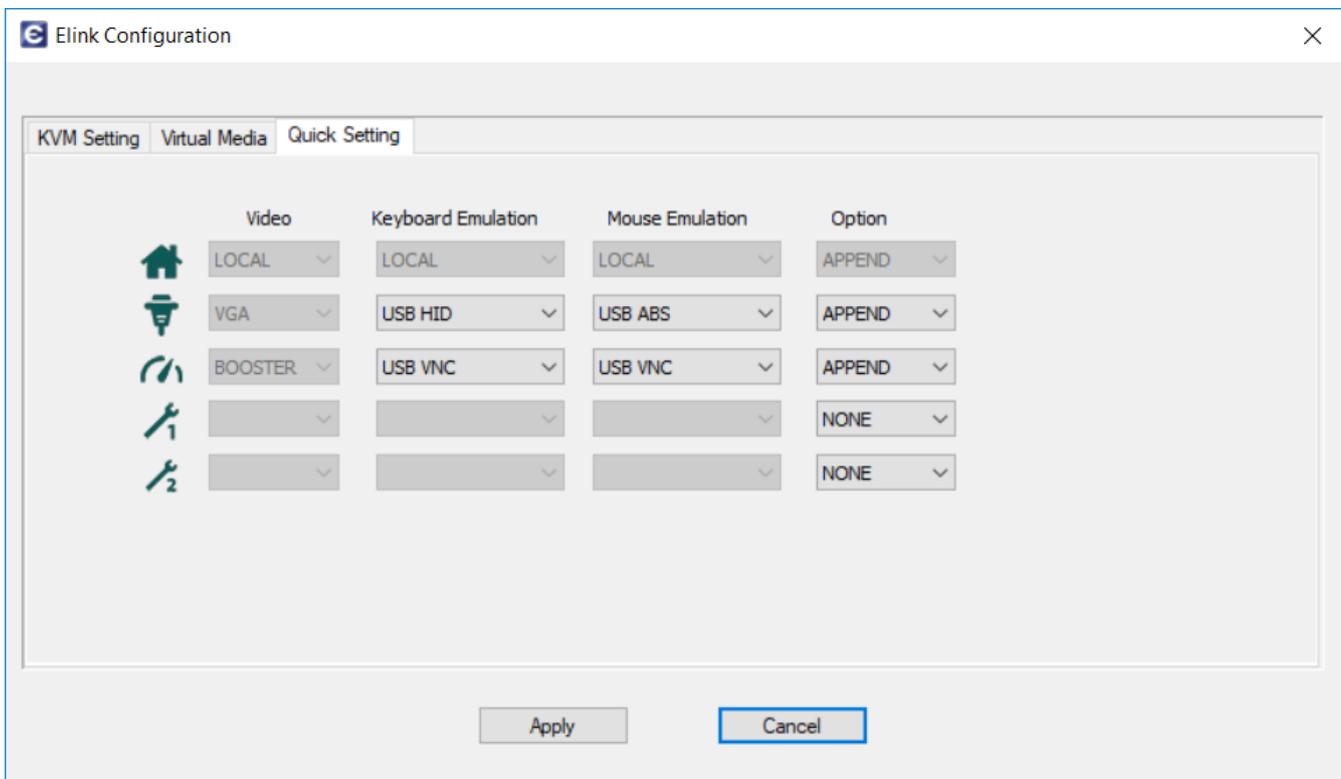


Figure 30. elink Configuration (Quick Setting) Tab

### 4.2.3. Pause and Capture Image

During the remote access session, user may want to capture the whole screen or a portion of the

screen for documentation purpose or can be used as template image for matching process. To capture the image doing below actions:

- Press the **Pause**  button
- After the screen is paused, hold the **Ctrl** button and while holding the **Ctrl** button, hold left mouse button and drag to select a captured region.

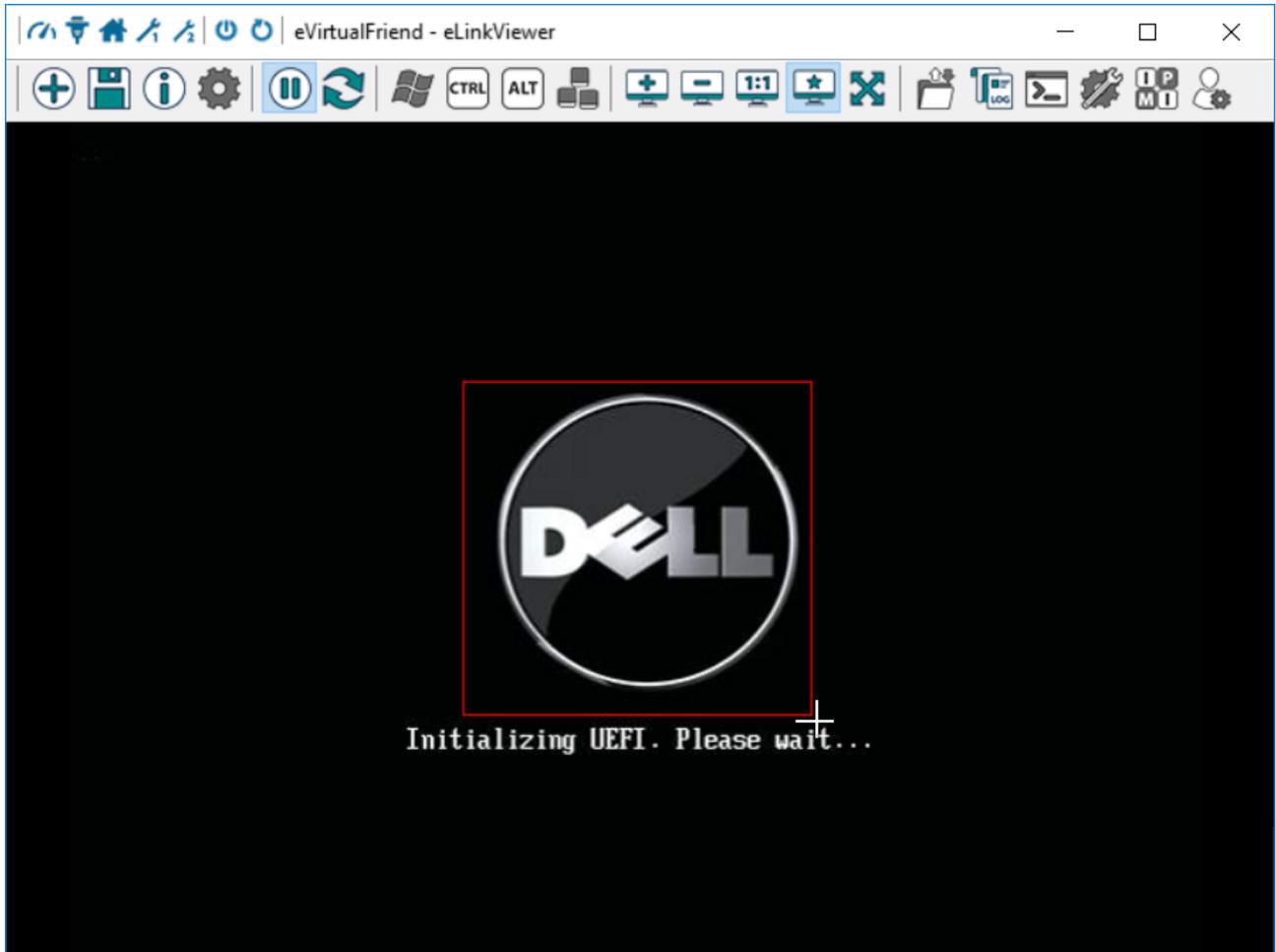


Figure 31. Capture Image

- When the captured region is selected, release the left mouse button. eLinkViewer will then pop up a dialog to ask for location and file name to save. Select the folder and input file name, and then click on **Save** button.

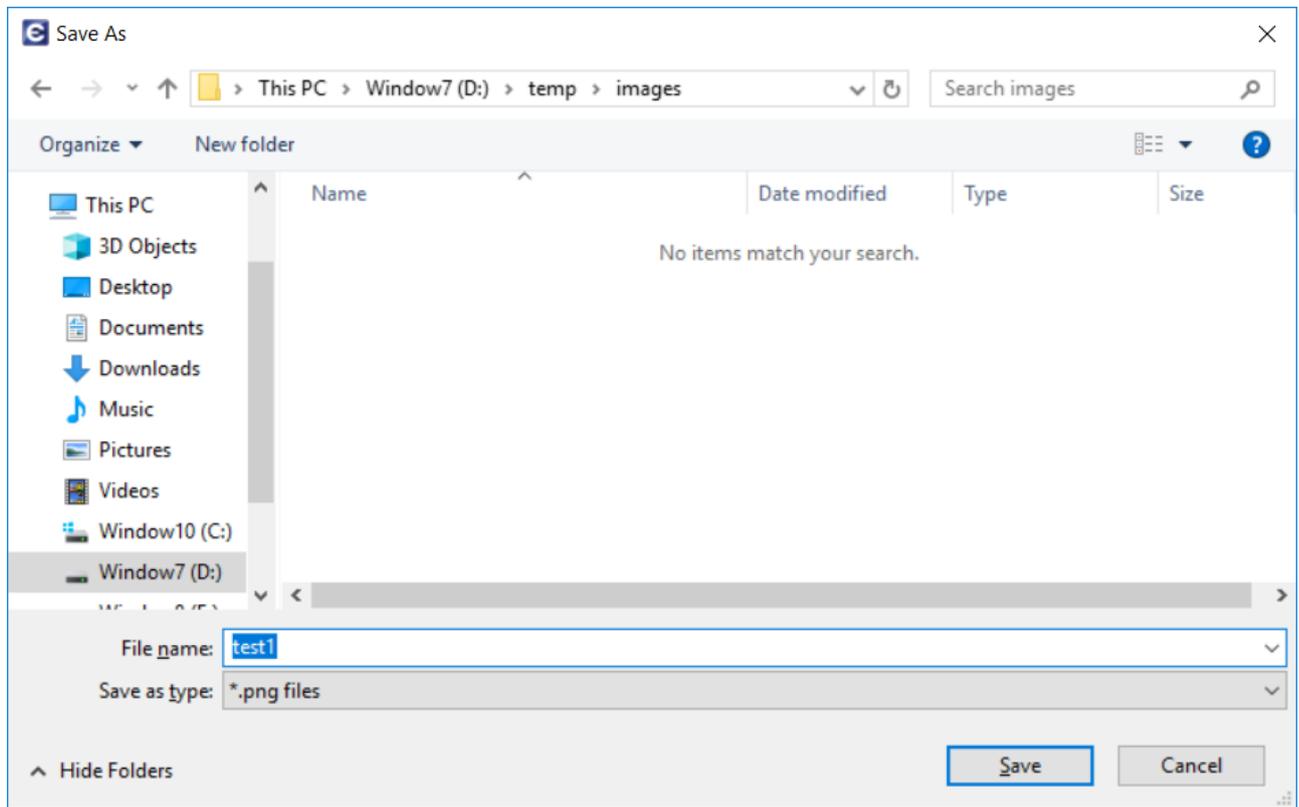


Figure 32. Save Image Dialog

#### 4.2.4. File Transfer GUI

The main purpose of this dialog is to upload the files from remote terminal to elinkKVM and manage files; either delete, rename, make directory...To upload files from remote terminal to elinkKVM, user

can use the File Transfer functionality by clicking on **File Transfer** button



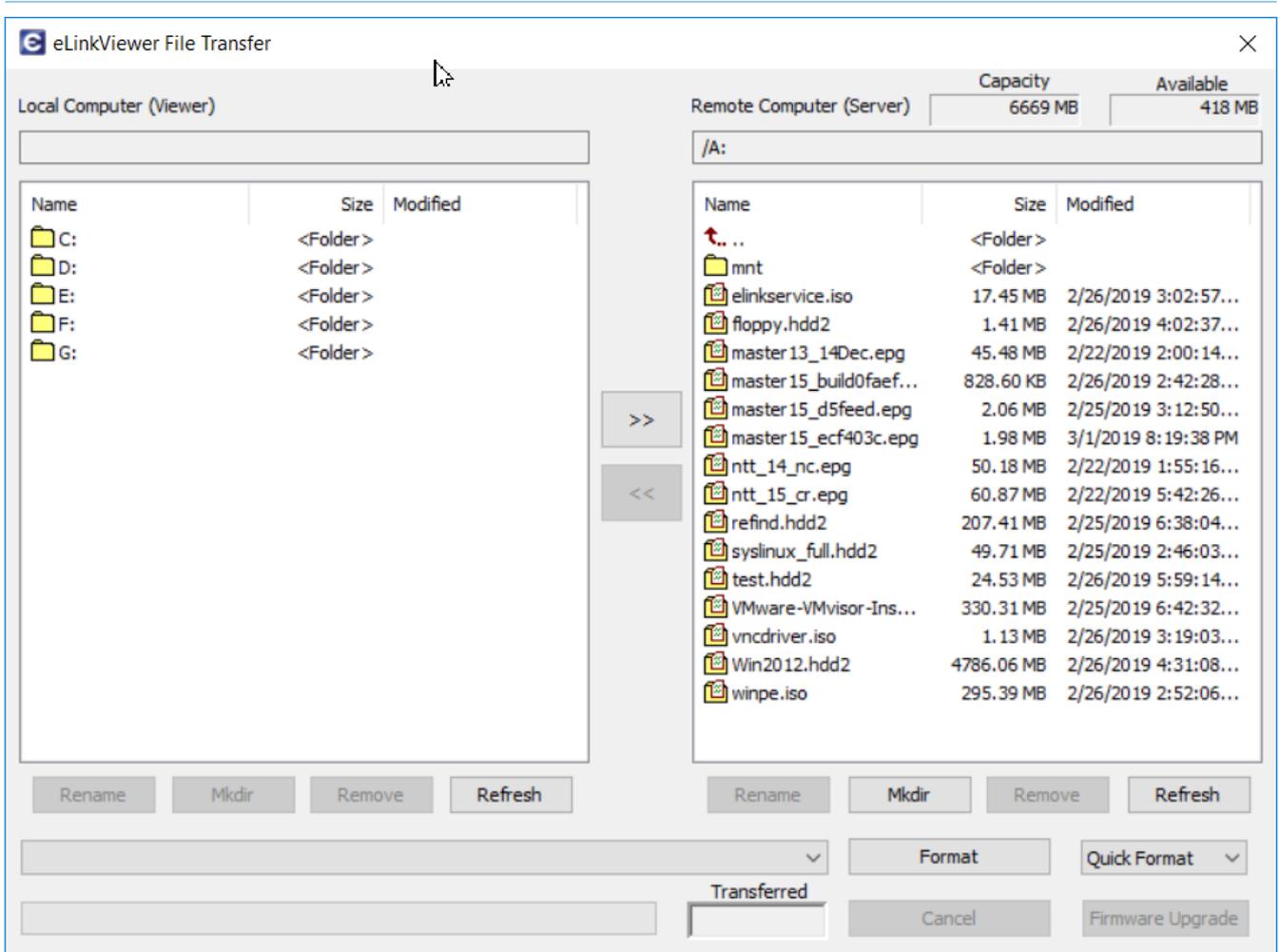


Figure 33. File Transfer Dialog

Browse the files in the left list box, select the folder in right list box, and then click on the upload

button  to upload files.

Additional user can also do below actions in this dialog:

- **Firmware upgrade:** by double clicking on the firmware file (.epg), then clicking on **Firmware Upgrade** button.

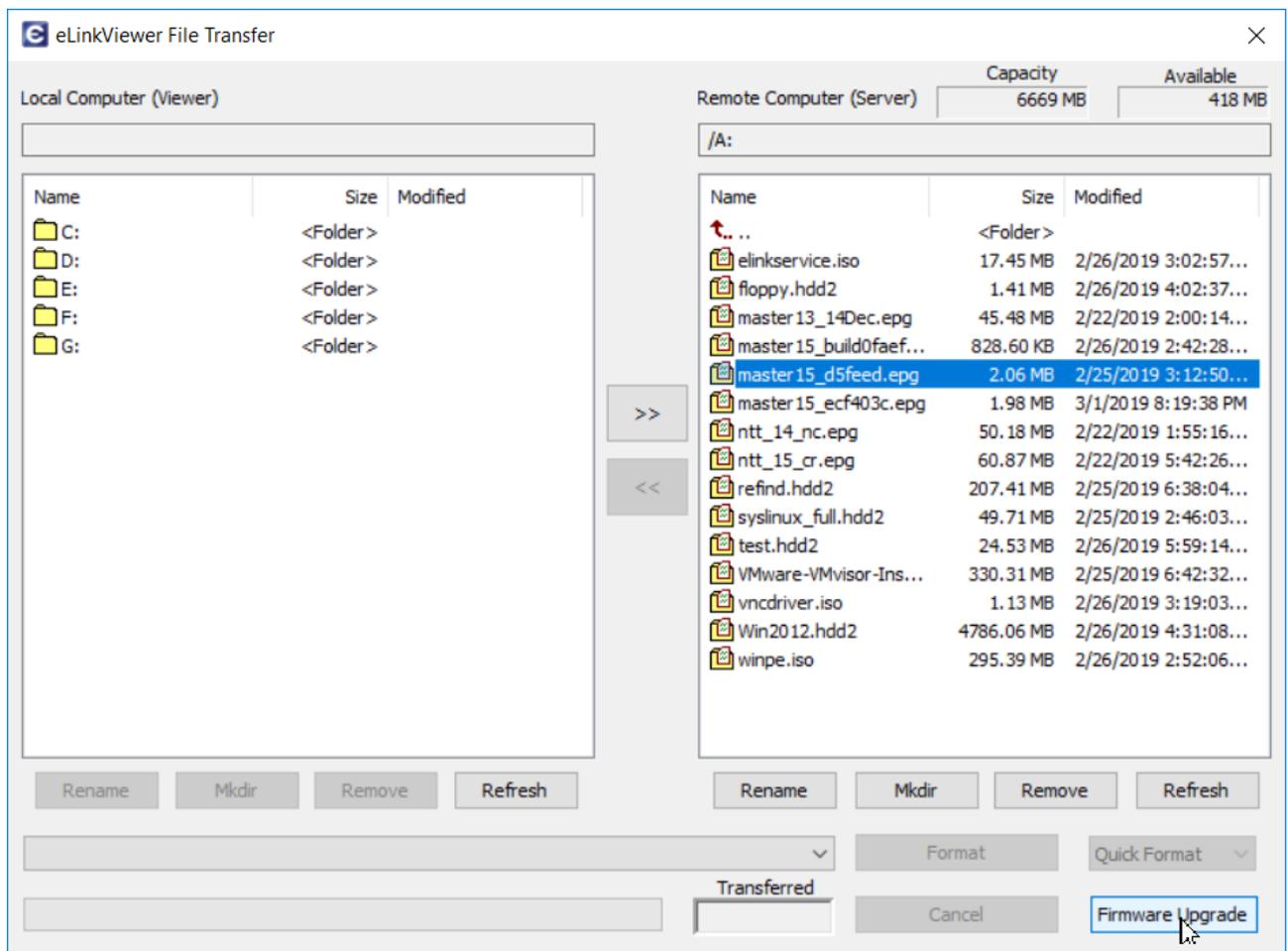


Figure 34. Firmware Upgrade

- **Format Internal Storage:** by selecting the remote disk (either A: or B:), click **Format** button.

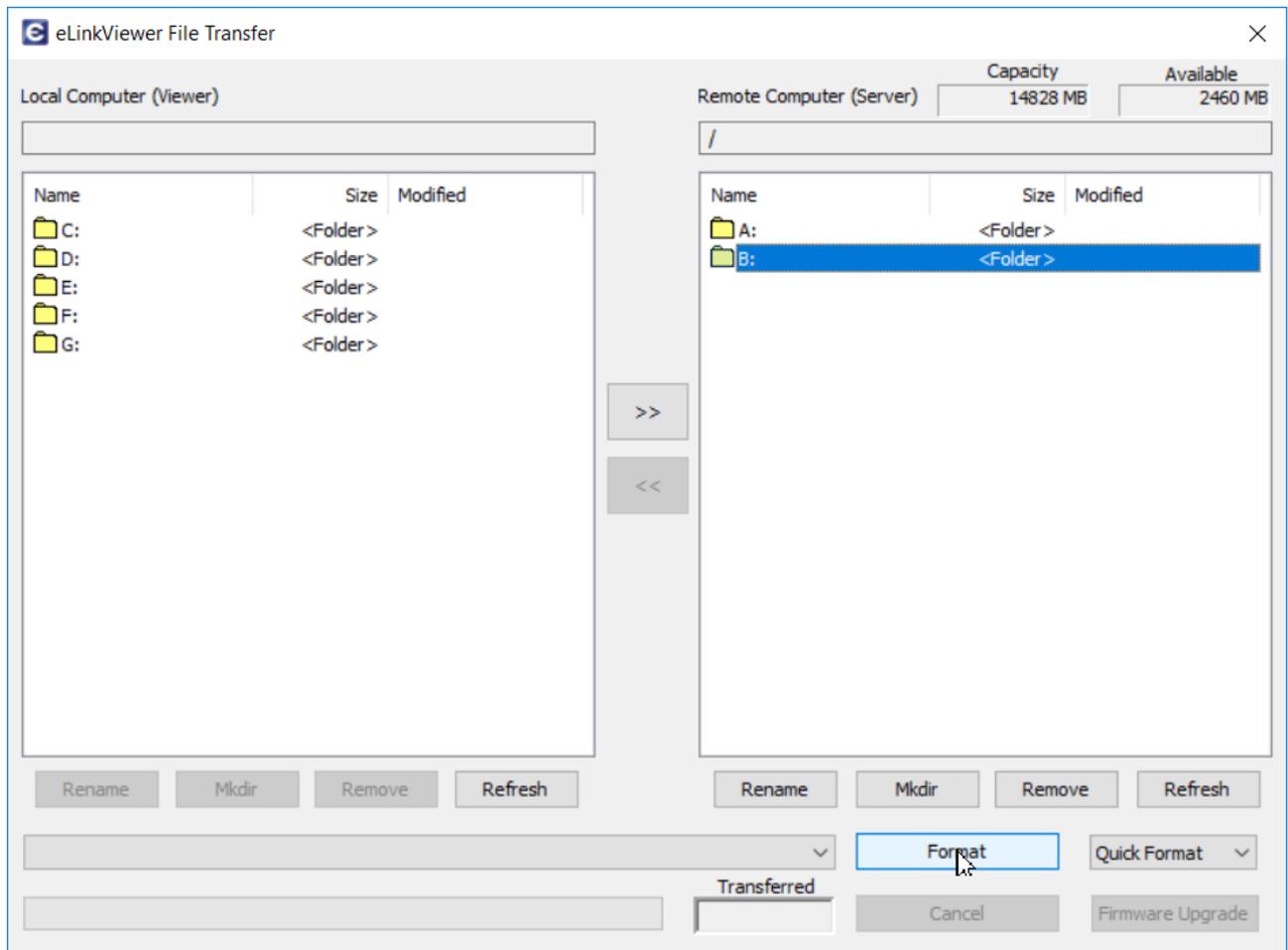


Figure 35. Disk format

#### 4.2.5. IPMI Command Center

The eLinkViewer integrated the IPMI (Intelligent Platform Management Interface) utility allows the viewer to scan and operate (such as Power on/off/reset) on IPMI compliant server that connects to eLinkKVM or within the local network of remote terminal.

By clicking on the **IPMI Command Center** button



the below dialog will pop up.

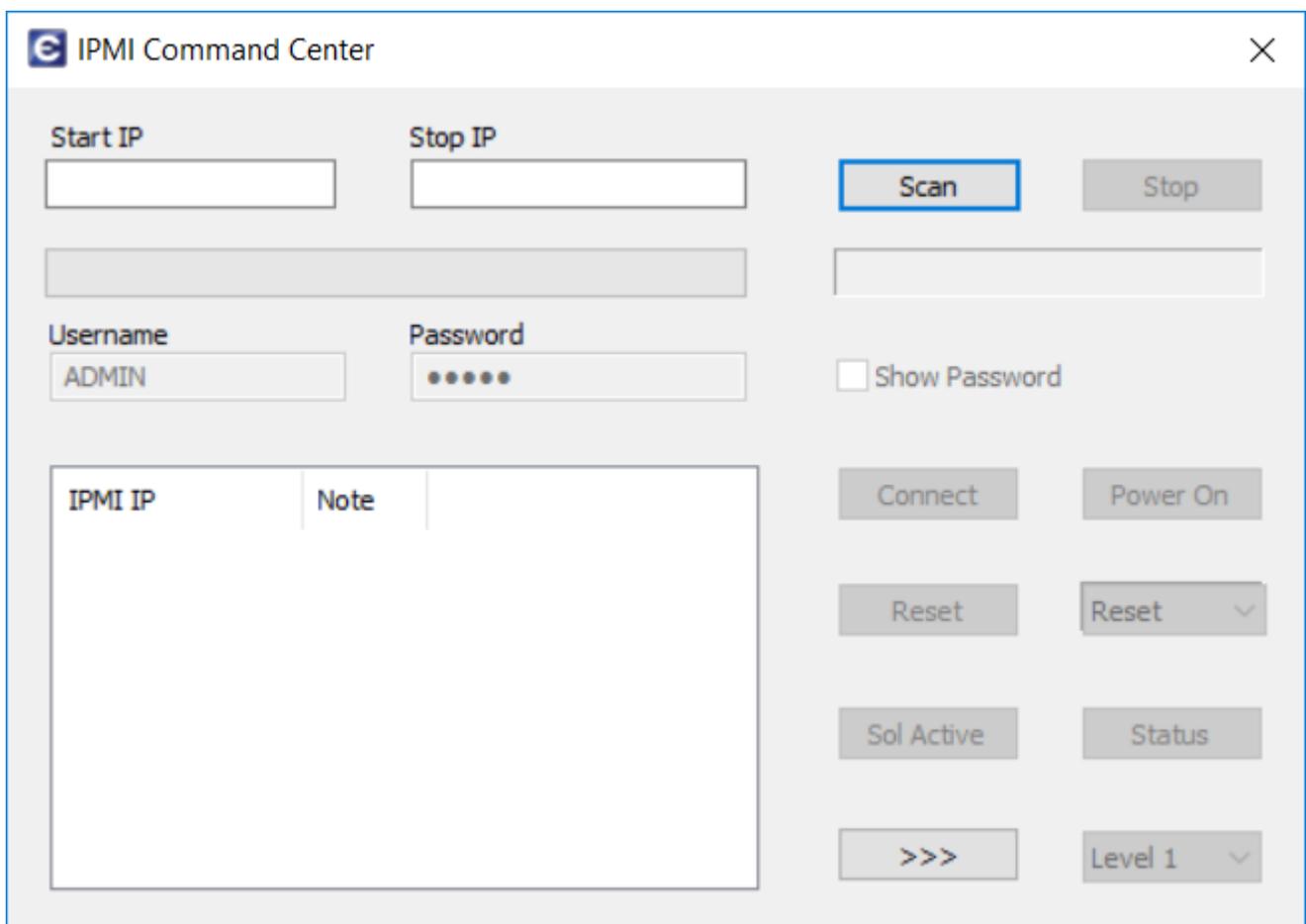


Figure 36. IPMI Command Center

- **Start IP**: start IP to begin scanning
- **Stop IP**: end IP to end scanning
- **Username - Password**: user name and password for each IPMI server. For example, default username and password are ADMIN/ADMIN for a Super Micro server.
- **Scan** : start scanning for an IPMI server. The available IPMI server will be listed in the list box. For those IPMI servers that are connected to elinkKVM, the name will appear as **elink-ipmi**.
- **Stop**: end a current scanning process.
- **Connect**: connects to an IPMI server. When an IPMI server is found, an IP address is displayed, click on the IP address to select, and then enter a Username and Password.

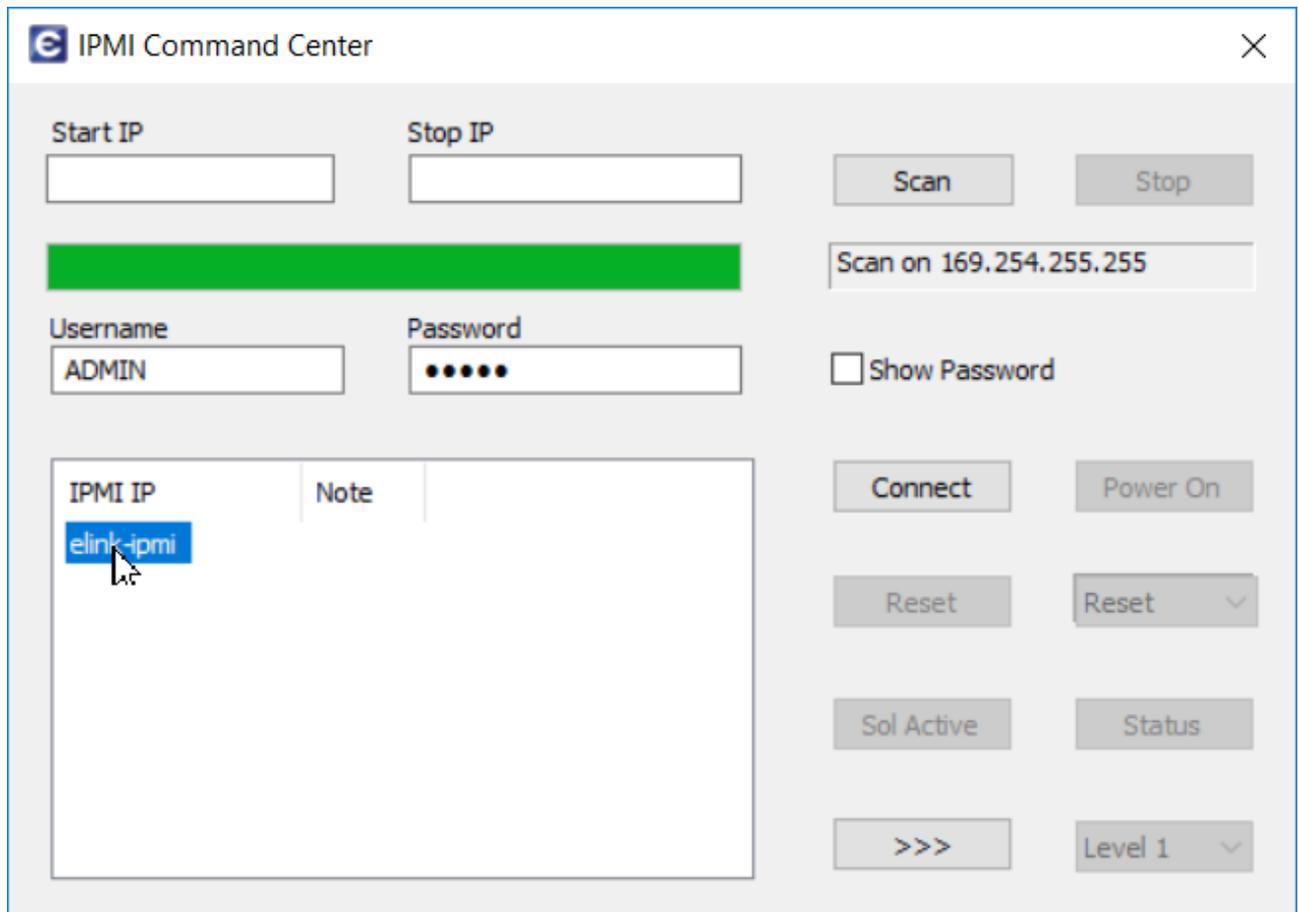


Figure 37. Connect to IPMI Server

- **Power On**: Turn a server on with IPMI.
- **Reset** with options:

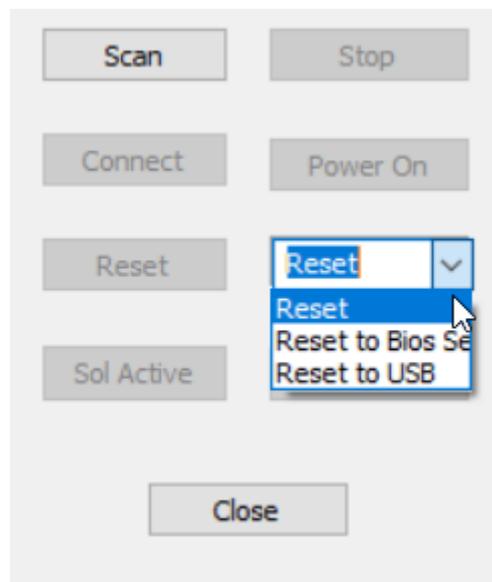


Figure 38. IPMI Reset with options

- **Reset** : restart a server.
- **Reset to Bios Setup**: restart a server and enter BIOS.
- **Reset to USB**: restart a server and boot with a USB device.
- **Sol Active**: enable IPMI management with Serial Over Lan (SOL).

In case the operation is not successful, the user can turn on the debug log by clicking on >>> button to show/hide IPMI log.

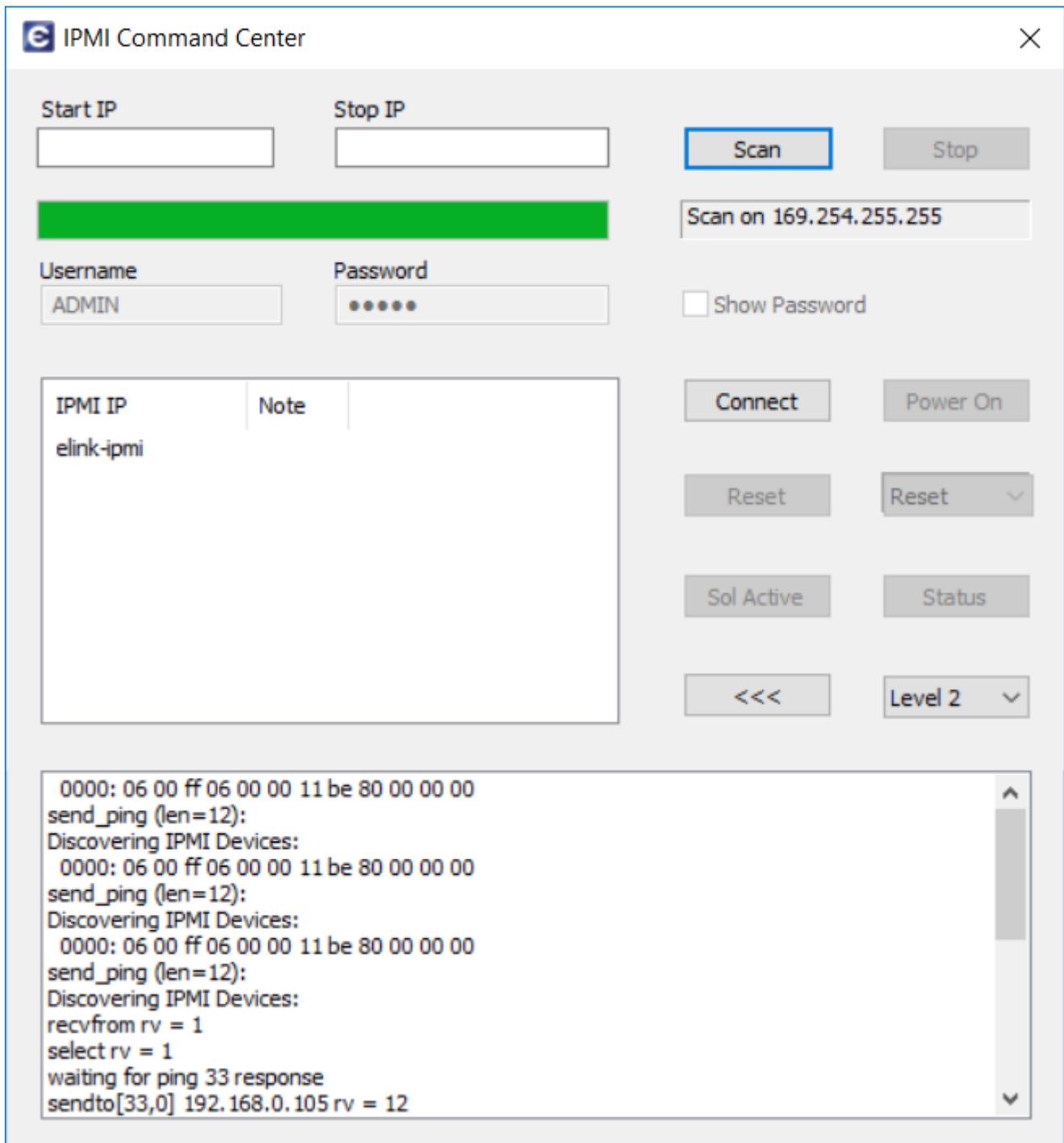


Figure 39. IPMI Command Center With Debug Log

**NOTE**

For IPMI server that connected to elinkKVM, the IP address of server need to set in IPMI tab of local view.

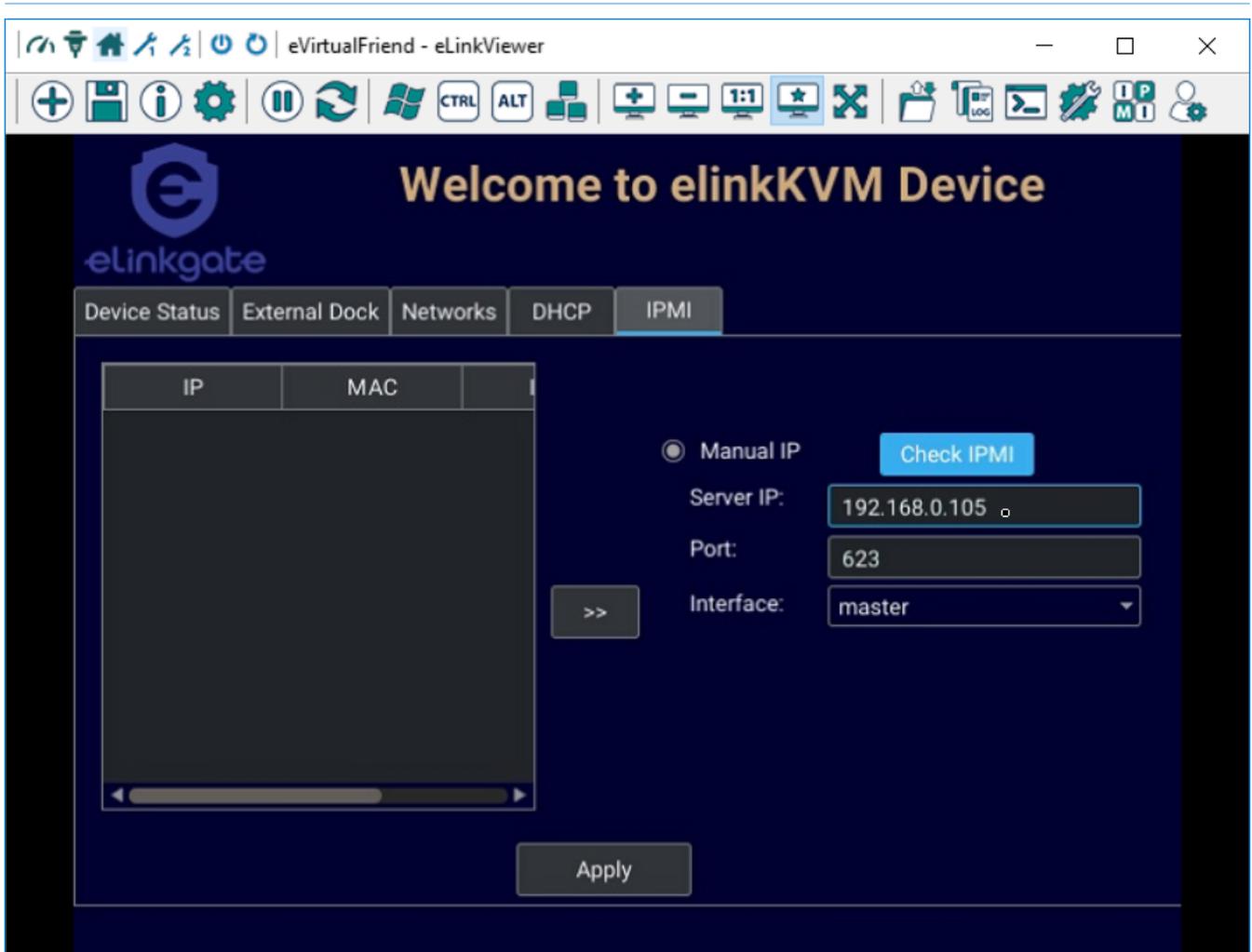


Figure 40. Local view for setting IPMI

#### 4.2.6. User Management

- Supports up to 8 user accounts.
- Multiple level management : The higher level can interrupt connection and modify attribute of lower level user account.
  - Admin : **Top level** account Data center admin account. Can create other lower priority account types like Manager and User.
  - Manager: **Middle level** User manager account. Can create User account.
  - Network: the specific user that can change the network setting but cannot do other things.
  - Operator: this user has only right to operate the remote server.

**NOTE** | the default username and password of elinkKVM is **admin/admin**

- Click on icon **User Management** to open User Management Dialog 

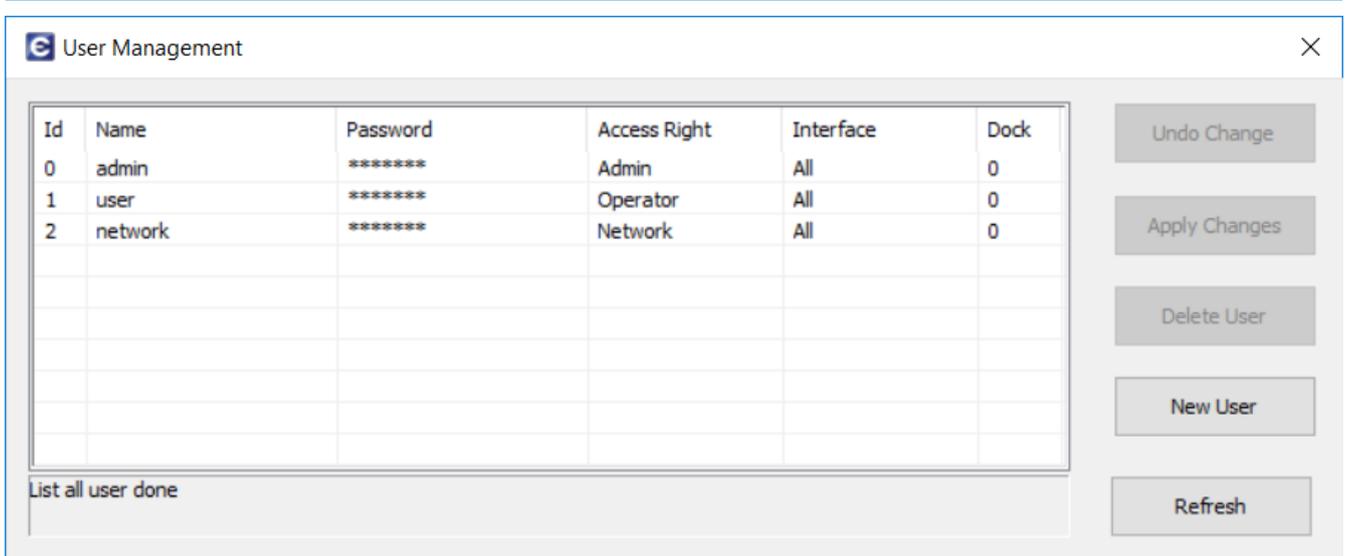


Figure 41. User management dialog

#### • Create new account

- Click **New User** to create new account
- Double click on a cell to edit its attributes.

For example: Double click on password cell of new user to show password configuration dialog.

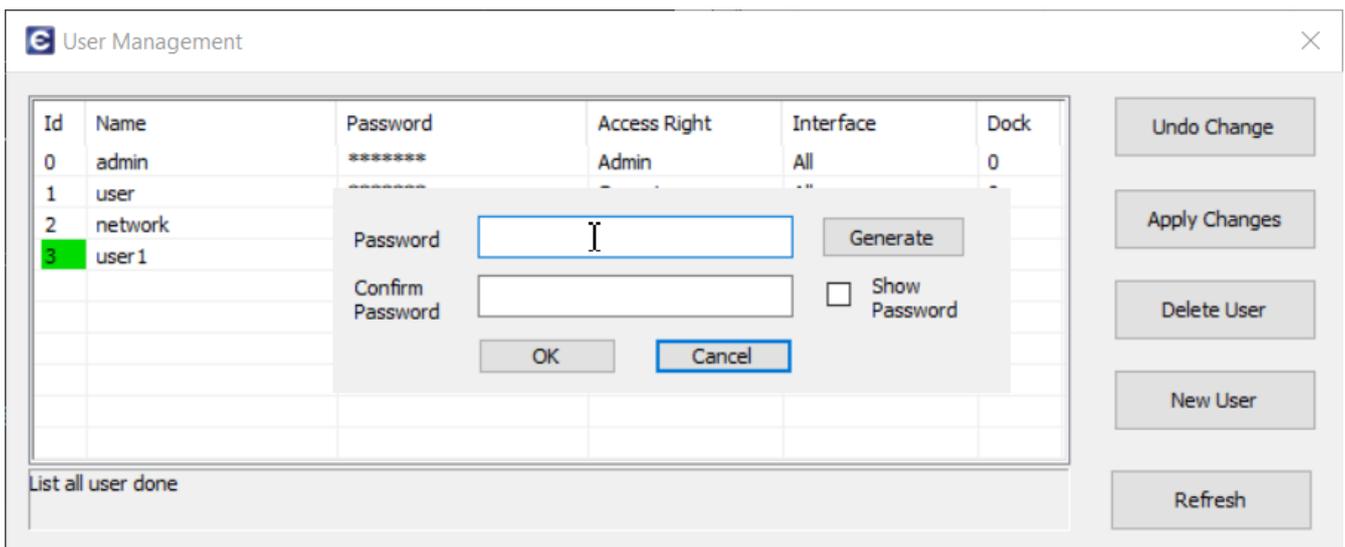


Figure 42. Double click on password field and set password for the new user account

Then input username and password.

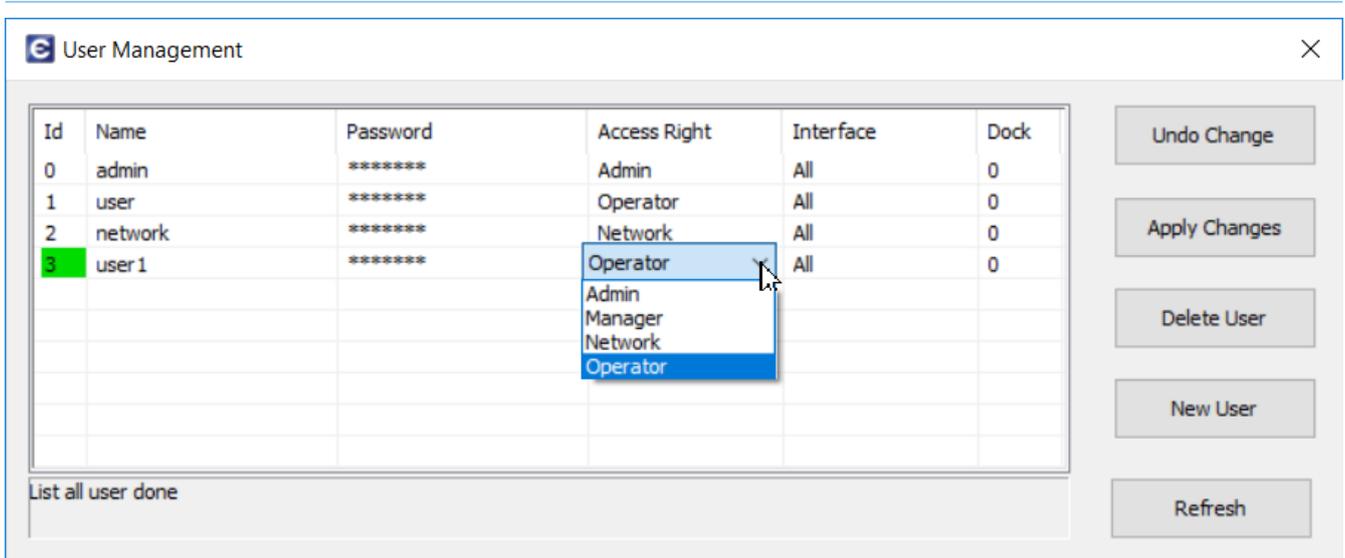


Figure 43. Click to Access Right field to modify the right of an account

Click **Apply Changes** to apply new configurations.

**NOTE**

User can also modify attributes of existing user account by clicking on appropriate field of select account need to modify.

**WARNING**

Only Admin/Account can run User management

#### 4.2.7. elink Event Log

Events are the information sent from elinkKVM to state the changes from remote computer such as USB reset, USB enumeration success...This may be useful while doing scripting and guessing the status of remote computer. The **elink Event Log** displays events and performance metrics related to elinkKVM.

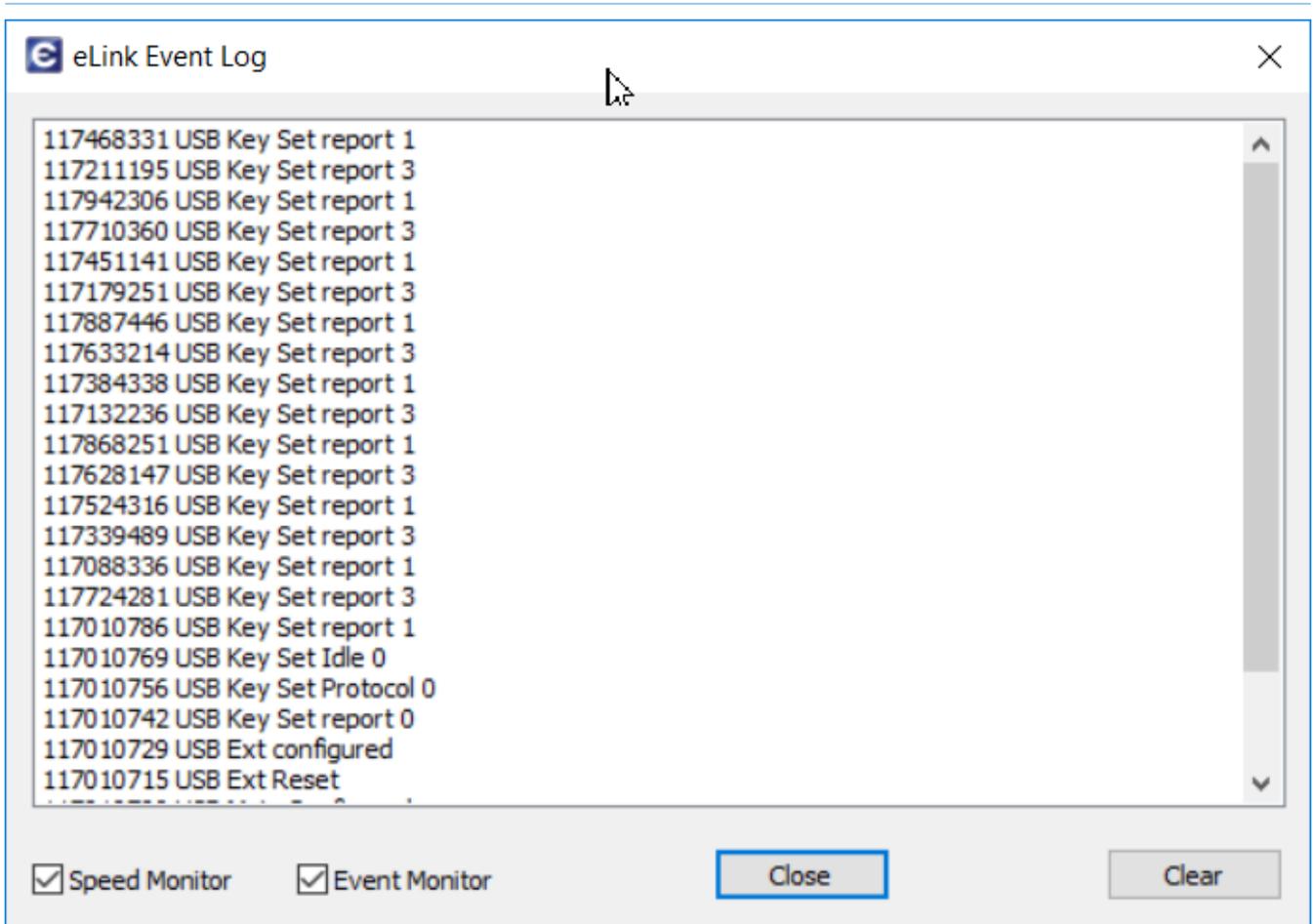


Figure 44. Event Log Dialog

## 4.3. Running Python Script

The elinkViewer exposes a set of API that is compatible with Python. This allows user writing script that automatically sends Keyboard, Mouse events to remote computer or captures screen data and applies image recognition to find the necessary screen...

### 4.3.1. Setup Python Console

The elinkViewer comes with a simple Python Console that allows user to quickly input Python commands and run directly in the console. There are a few ways to open the console:

- By running from command line -cons option

```

elinkviewer.exe -cons
elinkviewer.exe -cons=setup_win2012.py # to run the script directly

```

- By clicking on **Python Console** button . To hide the console, we can just click on the button again.

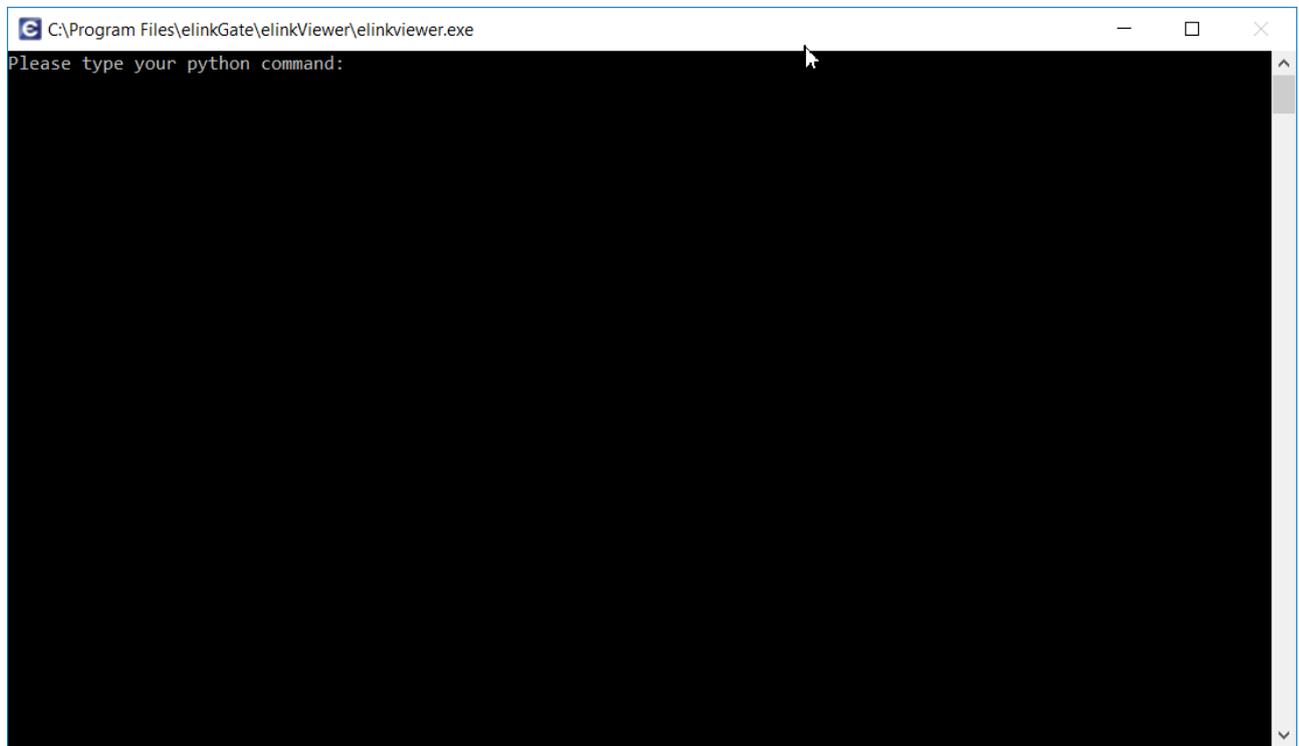


Figure 45. Python Console

In addition, elinkViewer can be recognized by others Python IDE such as IDLE, PyCharm, Visual Studio ... To setup the python dependencies for those IDE, user can either:

- Setup the environment manually by adding the path of elinkviewer.exe to PYTHONPATH

```
PYTHONPATH=%PYTHONPATH%;\PathTo_elinkViewer\
```

- Or by running elinkViewer at command line with `-add_python_env` option

```
elinkviewer.exe -add_python_env
elinkviewer.exe -add_python_env=system # this to add python PATH for
all users
```

To remove the environment, just run the command.

```
elinkviewer.exe -del_python_env
elinkviewer.exe -del_python_env=system
```

#### NOTE

The elinkViewer can itself run all basic python commands such as while, for... But if you want to make use of the power of other Python module, you will need to install Python36 from Python website.

### 4.3.2. How to call elinkViewer Python API

In the python console, either with elinkViewer or Python IDE, type below command:

```
import elink
```

Afterwards, user can call all APIs by elinkViewer. Below is a simple example:

```
elinkviewer -cons
```

Then in the console, typing below command

```
import elink
#connect to eLinkKVM with IP "10.42.02"
vnc = elink.newConnection("10.42.0.2","admin","admin")
vnc.info()
['01.00.01.01', 'eVirtualFriend', 'RFB 003.008\n', 640, 480]
```

#### 4.3.3. elinkViewer Python API

##### **elink Module**

The elink module defines the following functions:

##### **elink.newConnection(ipaddres,username,password)**

Create a VNC connection to elinkKVM or standard VNC server. This function will return a VNC object.

*Example*

```
#Connect to eLinkKVM with IP "10.42.02"
vnc = elink.newConnection("10.42.0.2","admin","admin")
```

##### **elink.getConnection()**

Return list of VNC Objects, which the elinkViewer is connected to.

*Example*

```
g = elink.getConnection()
for vnc in g:
    vnc.info()
```

The example produces below output.

```
['01.06.01.01', 'eVirtualFriend', 'RFB 003.008\n', 640, 480]
```

##### **elink.scanDevice()**

Scan for elinkKVM in local network

*Example*

```
elink.scanDevice()
```

The example produce below output

```
[['10.42.0.2', 'Elink-KVM-44:a3:bd'], ['10.42.0.3', 'Elink-KVM-44:a3:ef']]
```

**elink.exec()**

Execute a script

*Example*

```
elink.exec("setup_win2012.py")
```

**VNC Object**

The VNC object defines following functions:

**vnc.info()**

Return the info of vnc object

*Example*

```
g = elink.getConnection()
for vnc in g:
    vnc.info()
```

The example produces below output.

```
['01.06.01.01', 'eVirtualFriend', 'RFB 003.008\n', 640, 480]
```

*Note*

```
# <'01.00.01.02'>      : firmware version
# <'eVirtualFriend'   : elinkKVM name
# <'RFB 3.03'>        : Protocol
# <1366, 768>         : Width-Height of current screen
```

**vnc.close()**

Close vnc object

*Example*

```
vnc.close()
```

**vnc.sendString(string)**

Send a string of keyboard events to remote computer

*Example*

```
#eLinkKVM sends string hello world to server
vnc.sendString("hello world")
```

**vnc.sendKey(keyCode,releaseMode = 3)**

Send a key to server

**keyCode**

- The key code of keyboard event can be referenced in the **HID Usage Tables** from [https://www.usb.org/sites/default/files/documents/hut1\\_12v2.pdf](https://www.usb.org/sites/default/files/documents/hut1_12v2.pdf).
- Or user can use the mnemonic name: Backslash, CapsLock, Comma, Delete, DeleteForward, Down, End, Enter, Equals, Escape, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Grave, Help, Home, Insert, KPDivide, KPEnter, KPEquals, KPMultiply, KPNumLock, KPPoint, KPSubtract, Left, LeftAlt, LeftBracket, LeftControl, LeftGUI, LeftShift, Menu, Minus,NonUSBackslash, PageDown, PageUp, Pause, Period, PrintScreen, Quote, Right, RightAlt, RightBracket, RightControl, RightGUI, RightShift, ScrollLock, Semicolon, Slash, Space, Tab.

**releaseMode**

- 0: release key
- 1: press key
- 3: auto press, and then release key

*Example*

```
# eLinkKVM send key <LeftShift> lèn server
vnc.sendKey("LeftShift")    # Press Left Shift and release
vnc.sendKey("LeftShift",1)  # Press and hold Left Shift
vnc.sendKey("LeftShift",0)  # Release Left Shift
vnc.sendKey("A")            # Send Key 'A' and release
vnc.sendKey(0x4)           # Send Key 'a' and release
```

**vnc.sendKeyEx(keyList,downPeriod = 10)**

Send combination keys.

**keyList** is the list of key [key1,key2,key3...]

**downPeriod**: is the period of down key in milliseconds Default value is 10 milliseconds

*Example*

```
vnc.sendKeyEx(["LeftCtrl", "LeftShift", "Del"]) # send Ctrl-Alt-Key
```

**vnc.sendMouse(clickType,x,y)**

Send mouse event with **clickType** for moving x, y position to remote computer. Depending on the mouse mode (absolute or relative), the x,y is the absolute or relative movement.

**clickType** is combined of below value

- RDOWN: right button down
- LDOWN: left button down

- MDOWN: middle button down
- WUP: the wheel up
- WDOWN: the wheel down
- CLICK: mouse click
- DCLICK: mouse double click

*Example*

```
vnc.sendMouse(0, 100, 100) # move mouse to 100,100
vnc.sendMouse("RDOWN", 100, 100) # right mouse down at 100,100
vnc.sendMouse("RDOWN|CLICK", 100, 100) # right mouse click at 100,100 (or
0x40)
vnc.sendMouse("RDOWN|DCLICK", 100, 100) # right mouse double click at
100,100 (or 0x80)
```

### **vnc.setUsbMode(usbMode,usbOption,virtualMediaList)**

Set USB emulation mode

**usbMode** defines the USB peripheral that will be emulated. It will be combination of below values.

- USB\_MODE\_KEY = 0x0001: configure elinkKVM as USB Keyboard
- USB\_MODE\_MOUSE = 0x0002: configure elinkKVM as USB Mouse
- USB\_MODE\_MSC = 0x0004: configure eLinkKVM as USB Mass storage
- USB\_MODE\_VNC\_HID = 0x0008: configure elinkKVM as USB HID device which is designed for booster mode
- USB\_MODE\_MOUSE\_ABS = 0x0040: configure elinkKVM as USB Absolute Mouse

**usbOption** This value is always 0

**virtualMediaList**

This is a list of virtual media images in format (.hdd, .hdd2, .iso ....). When this list is not empty, the USB\_MODE\_MSC is set automatically.

*Example*

```
vnc.setUsbMode("USB_MODE_KEY|USB_MODE_VNC_HID",0,["A:\win10.hdd2"])
vnc.setUsbMode(0x09,0,["A:\win10.hdd2","B:\hiren.iso"])
vnc.setUsbMode(0,0,["A:\win10.hdd2","A:\ubuntu.hdd2"])
```

### **vnc.setVncMode(vncMode)**

Set VNC mode

**vncMode**

- MODE\_VNC\_DUMMY = 0: set VNC mode to local mode

- `MODE_VNC_RGB = 1`: set VNC mode to VGA mode
- `MODE_VNC_BOOSTER = 2`: set VNC mode to Booster mode

*Example*

```
vnc.setVncMode("MODE_VNC_RGB") #Set VGA mode
vnc.setVncMode("MODE_VNC_DUMMY") #Set Local mode
vnc.setVncMode("MODE_VNC_BOOSTER") #Set Booster mode
```

#### **vnc.setKeyMode(keyboardMode)**

Set keyboard emulation mode

##### **keyboardMode**

- `KEY_INTF_NONE = 0`: local keyboard mode
- `KEY_INTF_HID = 1`: USB HID keyboard mode
- `KEY_INTF_VNC = 2`: Keyboard VNC HID mode (This is for Booster)

*Example*

```
vnc.setKeyMode("KEY_INTF_HID")
vnc.setKeyMode(1)
vnc.setKeyMode("KEY_INTF_VNC")
vnc.setKeyMode(2)
```

#### **vnc.setMouseMode(mouseMode)**

Set mouse emulation mode.

##### **mouseMode**

- `POINT_INTF_NONE = 0`: local mouse mode
- `POINT_INTF_HID = 1`: USB HID Mouse mode
- `POINT_INTF_VNC = 2`: USB VNC HID mode (this is for Booster)
- `POINT_INTF_HID_ABS = 3`: USB HID Absolute Mouse mode

*Example*

```
vnc.setMouseMode("POINT_INTF_HID")
vnc.setMouseMode(1)
vnc.setMouseMode("POINT_INTF_VNC")
vnc.setMouseMode(2)
vnc.setMouseMode("POINT_INTF_HID_ABS")
vnc.setMouseMode(3)
```

#### **vnc.setKeyIdle(minIdleTime)**

Set minimum idle time between 2 keys. This feature is meant to handle the case if network jams and many keyboard events arrive elinkKVM at same time, then the device will delay at least this

minimum time before emulating a keyboard event.

### `minIdleTime`

The idle time is in milliseconds.

#### *Example*

```
vnc.setKeyIdle(200) # 200ms delay between 2 keys send
# Each character in string "hello world" will be send with delay time is
200ms
vnc.sendString("hello world")
```

### `vnc.setVncIdle(idleTime)`

Configure eLinkKVM send Screen Idle event when there is no screen change in specific time (Idle time).

### `idleTime`

The period is in milliseconds that elinkKVM will send Screen Idle event if there is no screen change.

#### *Example*

```
vnc.setVncIdle(200) # 200 ms timeout for idle event
```

### `vnc.ipmiConnect(ipmiServer,username,password)`

Connect to IPMI server

*ipmiServer is the address of IPMI*

This can be IP address or `elink-ipmi`, which represents that the IPMI is connected to elinkKVM.

### `username`

The username to login IPMI server

### `password`

The password to login IPMI Server

#### *Example*

```
#Connect to IPMI direct 10.42.0.100
#User name: "ADMIN"
#Password: "ADMIN"
vnc.ipmiConnect("10.42.0.100", "ADMIN", "ADMIN")
vnc.ipmiConnect("elink-ipmi", "ADMIN", "ADMIN")
```

### `vnc.ipmiPower(action)`

This will power on/off server through IPMI

`action` is the power action

- action = 0: power off server

- action = 1: power on server

*Example*

```
vnc.ipmiConnect("elink-ipmi", "ADMIN", "ADMIN")
vnc.ipmiPower(1) # power on server
vnc.ipmiPower(0) # power off server
```

#### **vnc.ipmiReset(option)**

Hard reset remote server through IPMI

##### **option**

- 0: normal reset
- 1: hard reset to SVC
- 2: hard reset to EFI
- 3: hard reset to PXE
- 4: hard reset to DVD/CD Media
- 5: hard reset to hard disk
- 6: hard reset to Bios setup
- 7: hard reset to floppy/removable disk

*Example*

```
vnc.ipmiReset(0) # normal reset server
vnc.ipmiReset(6) # reset to Bios Setup
```

#### **vnc.ipmiStatus()**

Return the status of IPMI server [connectedState,powerState,message]

##### **connectedState**

- 0: the IPMI server is not connected
- 1: the IPMI server is connected

##### **powerState**

- 0: the server is power off
- 1: the server is power on

##### **message**

The IPMI status message

*Example*

```
vnc.ipmiConnect("elink-ipmi", "ADMIN", "ADMIN")
vnc.ipmiStatus()
```

The example may produce below output

```
[1, 0, 'Ipmi is connected']
```

### **vnc.remoteFileList(remotePath)**

List the files and folder at remote path. This will return a list of files or folder

**remotePath** The path in format "A:\\folder1\\folder2..."

*Example*

```
vnc.remoteFileList()
vnc.remoteFileList("A:\\")
```

This example may produce below output

```
[['A:', 1, 0], ['B:', 1, 0]]
[['master_slave1010.epg', 0, 52139048], ['elinksetup.iso', 0, 75694080],
['win2012.iso', 0, 247324672], ['syslinux_full.hdd2', 0, 52126208]]
```

### **vnc.remoteFileDelete(remoteFilePath)**

Delete the remote file on elinkKVM

**remoteFilePath**

The path in format "A:\\folder1\\file1"

*Example*

```
vnc.remoteFileDelete("A:\\win2012.iso")
```

### **vnc.remoteFileRename(remoteFileOrg,remoteFileDst)**

Rename the remote file on elinkKVM

**remoteFileOrg**

The original name of file

**remoteFileDst**

The new name and location of file will be renamed

*Example*

```
vnc.remoteFileRename("B:\\win2012.iso", "A:\\win2012.iso")
```

### **vnc.remoteFileUpload(localFile,remoteDstPath)**

Upload the file from local folder of remote terminal to elinkKVM.

**localFile**

The path of source file, which is located on remote terminal computer

#### `remoteDstPath`

The folder on elinkKVM, which will store the file.

#### *Example*

```
vnc.remoteFileUpload("C:\\images\\win2012.iso","A:\\") # this will upload
file win2012.iso to A: on elinkKVM
```

#### **`vnc.remoteFileCopy(remoteSrcFile,remoteDstFile)`**

Copy the file on elinkKVM

#### `remoteSrcFile`

The source file on elinkKVM

#### `remoteDstFile`

The destination file on elinkKVM

#### *Example*

```
vnc.remoteFileCopy("B:\\win2012.iso","A:\\win2012.iso")
```

#### **`vnc.matchScreen(matchingImage,matchingScore,attentionPeriod)`**

This will get the remote computer screen and do matching with template image. This function will return the matching window and matching score `[[x,y,w,h], matching_score]`

#### `matchingImage`

The template image file name, which is stored on the remote terminal computer

#### `matchingScore`

The score ranges from 0 to 1.0. If there is 100% matching the score will be 1.0.

#### `attentionPeriod`

This will set the period in milliseconds highlighting of matching window screen if the matching score is met.

#### *Example*

```
matchData = vnc.matchScreen("test1.png",score=0.9,attentionPeriod = 500)
```

Implementation of function `waitImage`

```
def waitImage(vnc, img, score=0.9, attentionPeriod=3000):
    print("wait for " + repr(img))
    while True:
        loc = vnc.matchScreen(img, score, attentionPeriod)
        if loc[0] != None:
            break
        sleep(2)
    sleep((attentionPeriod + 500) / 1000)
    return loc[0]
```

```
matchingData = waitImage(vnc, "test1.png")
x = int(matchingData[0])
y = int(matchingData[1])
w = int(matchingData[2])
h = int(matchingData[3])
print("Found {} in screen at x:{} y:{} width:{}
height:{}".format("test1.png", x, y, w, h))
```

### **vnc.getEvent()**

The elinkKVM will return events if there are changes in USB signal such as USB Reset, USB enumeration, USB Set Protocol... This signal will be useful to know about the booting status. This function will return the Event Object that represents those events.

Below are the event codes:

```
EVT_USB_EXT_RESET = 1
EVT_USB_EXT_CONFIGURED = 2
EVT_EXT_BUFFER_FULL = 3
EVT_KEY_PHANTOM = 4
EVT_USB_KEY_SET_PROTOCOL = 5
EVT_USB_KEY_SET_REPORT = 6
EVT_USB_KEY_SET_IDLE = 7
EVT_USB_MAIN_RESET = 8
EVT_USB_MAIN_CONFIGURED = 9
EVT_USB_MAIN_CONNECTED = 11
EVT_USB_MAIN_DISCONNECTED = 12
EVT_STORAGE_FIRST_READ = 13
EVT_STORAGE_FIRST_READ2 = 14
EVT_STORAGE_FIRST_WRITE = 15
EVT_STORAGE_FIRST_WRITE2 = 16
EVT_RESET_MAX_REACH = 17
EVT_FILE1_ERROR = 18
EVT_FILE2_ERROR = 19
EVT_FILE_VNC_ERROR = 20
EVT_NAND_ERROR = 21
EVT_VNC_HIP_CONNECTED = 22
EVT_STORAGE_IDLE = 23
EVT_STORAGE_REGION_COUNT = 24
EVT_KEY_TRIGGER_ON_RESET = 25
EVT_KEY_ON_POWER_DONE = 26
EVT_DISK_ON_POWER_DONE = 27
```

#### *Example*

```
# below is process to check if event 5 existed => Press key F12
while True:
    e = vnc.getEvent() #get event object
    if e.getIdCode() == 5:
        #check event 5 (EVT_USB_KEY_SET_PROTOCOL) is existed
        vnc.sendKey("F12") # send key F12 to server
    break
```

#### **vnc.clrEvent()**

Clear all the pending events in queue

*Example*

```
#Clear all existed event in queue and waiting for new event 5
vnc.clrEvent()
# below is process for waiting event 5 existed => Press key F12
while True:
    e = vnc.getEvent() #get event object
    if e.getIdCode() == 5:
        #check event 5 (EVT_USB_KEY_SET_PROTOCOL) is existed
        vnc.sendKey("F12") # send key F12 to server
        break
```

**elink Event Object**

The elink Event Object defines below API

**event.getIdCode()**

Return the id code of event

**event.getData(fieldName)**

Return the data that is associated with event.

*Example*

```
vnc.clrEvent()
vnc.probeTemp()
while True:
    e = vnc.getEvent()
    if e.getIdCode() == elink.EV_LOG_MASTER_EX:
        print("temperature ",e.getData("temp"))
        break
```

# CHAPTER 5. ELINKKVM ADVANCE CONFIGURATION

## 5.1. Local Mode

When user changes VNC mode to local mode, the **Remote Screen** will show the internal status of elinkKVM. User can, through this view change/monitor the network settings, DHCP server and IPMI.



Figure 46. elinkKVM Local Sreen

### 5.1.1. Network Tab

In local screen, select the **Networks** Tab and change the network setting for Slave and Master port. Click on **Apply** button to apply the changes.

- The Slave port can be changed to DHCP Client or Manual IP address
- The Master port can be changed to DHCP Client, DHCP Server or Manual IP address



Figure 47. elinkKVM Network Setting

### 5.1.2. DHCP Tab

In local screen, by selecting the **DHCP** Tab, user can monitor the DHCP clients, which are currently connected to elinkKVM.

To flush the Cache, click on **Flush Cache** button.

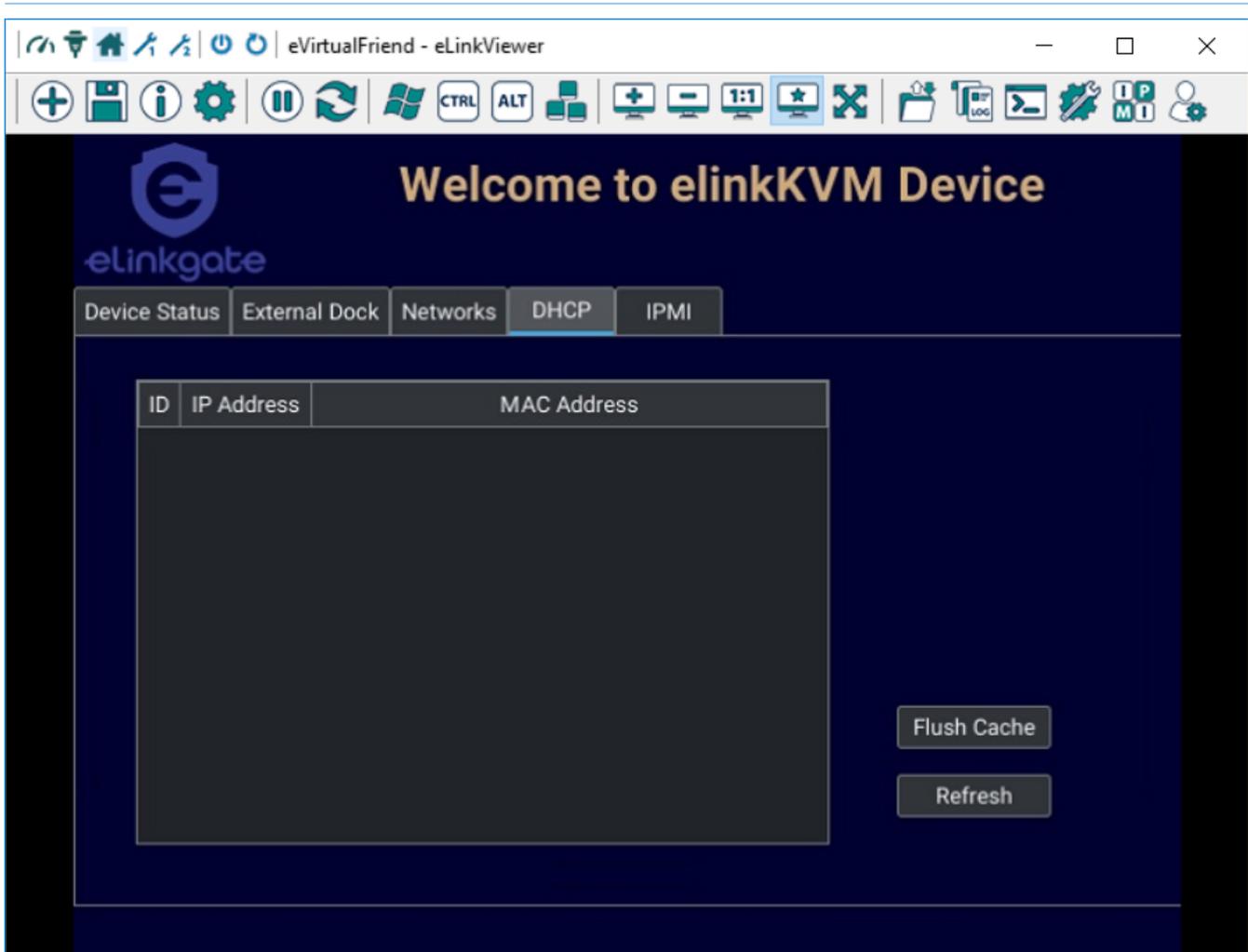


Figure 48. eLinkKVM DHCP Setting

### 5.1.3. IPMI Tab

In local screen, by selecting the **IPMI** Tab and user change the setting of IPMI server which is connected to eLinkKVM.

Input the IP and port of IPMI server in **Server IP** and **Port**. User will need to set the port either Master or Slave the IPMI is connected to.

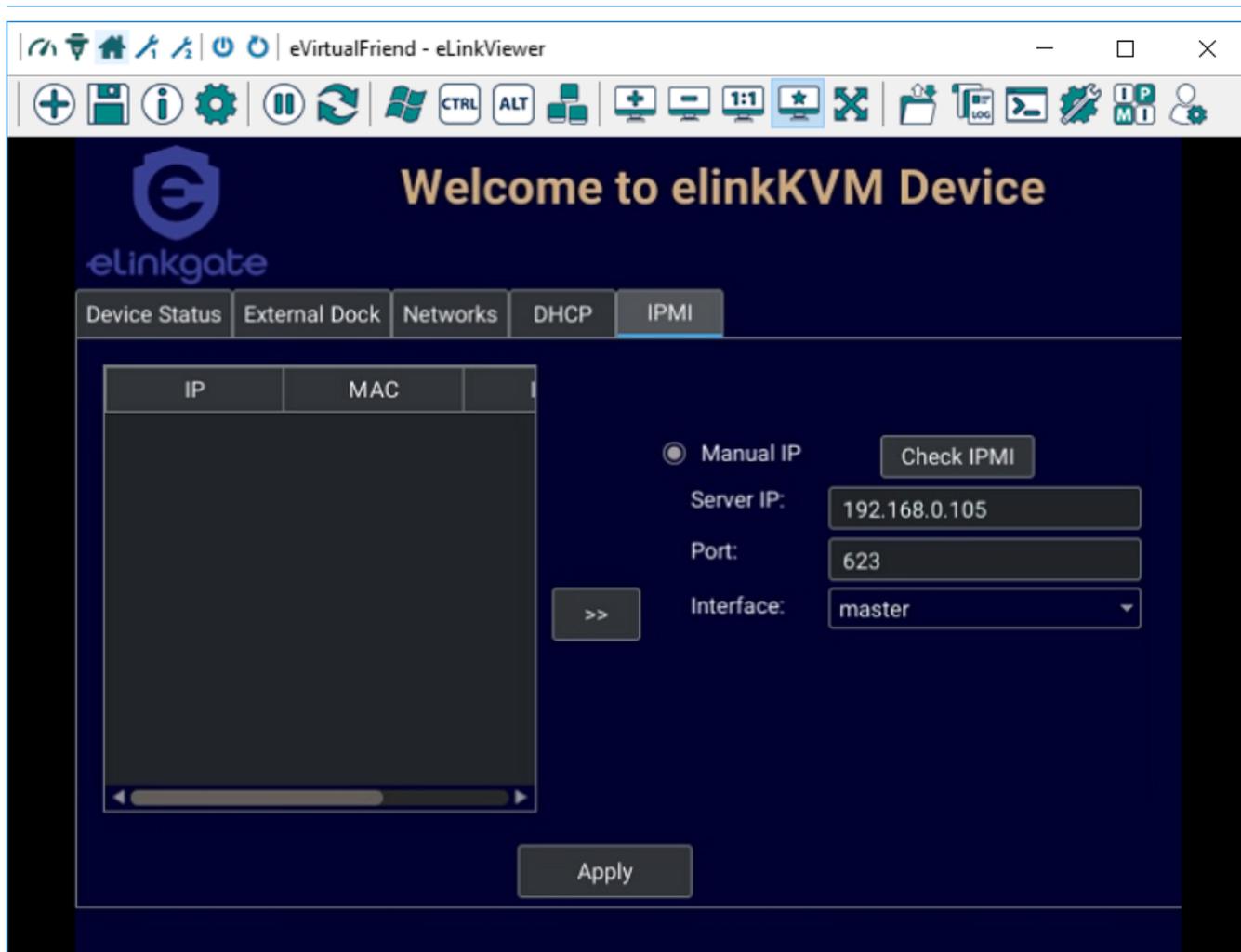


Figure 49. eLinkKVM IPMI Setting

To check whether the setting is correct or not, click on **Check IPMI**. The screen will display whether it is valid IPMI server.

Click on **Apply** button to apply the changes.

## 5.2. Offline Configuration

eLinkKVM supports offline configuration.

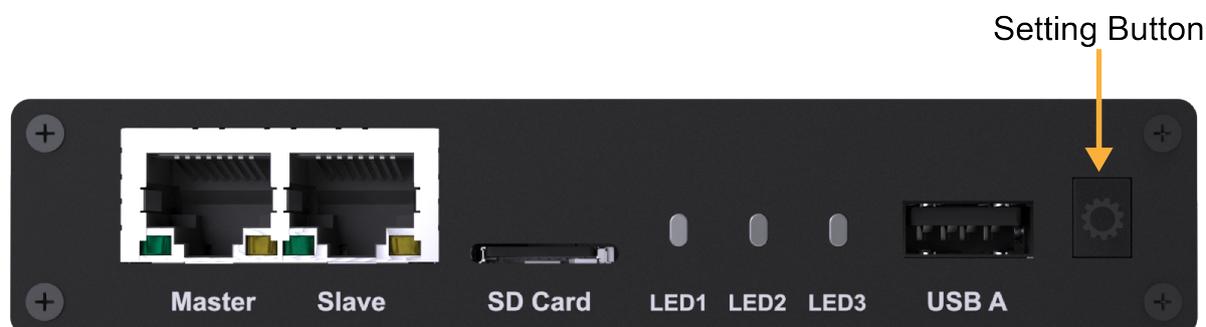
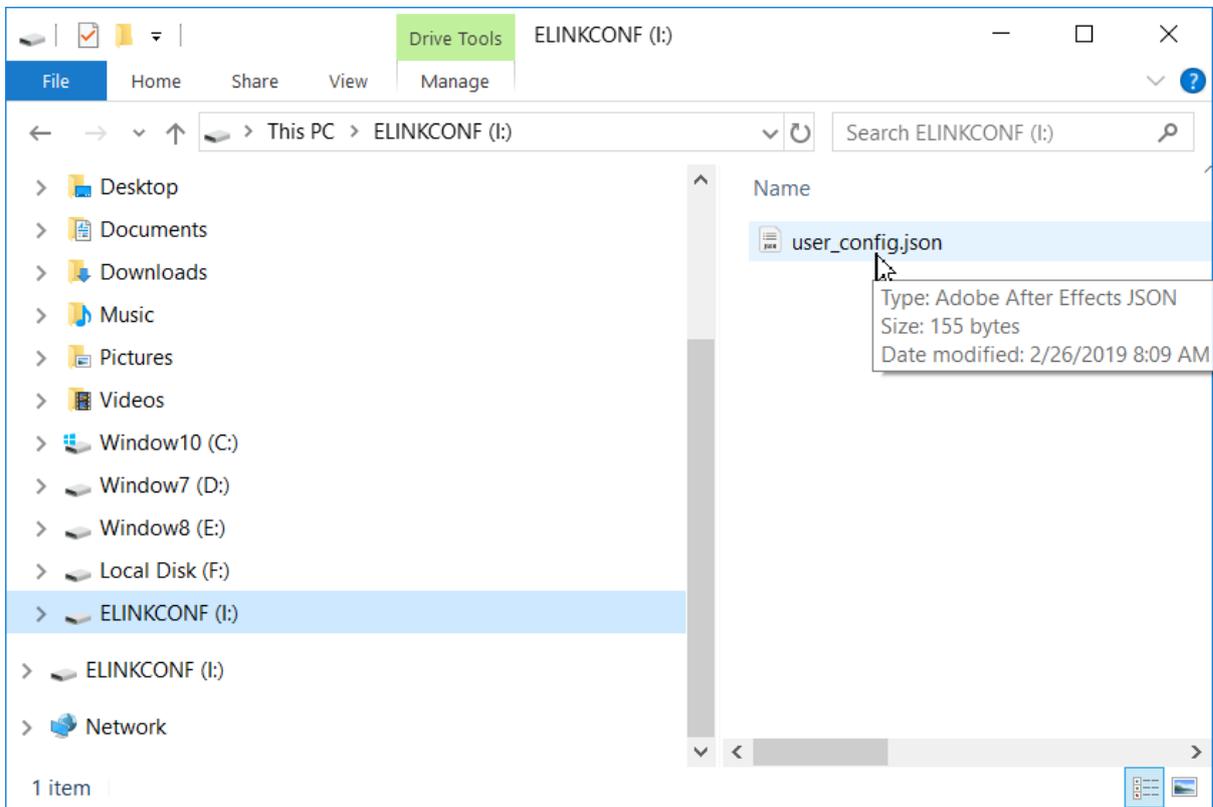


Figure 50. Setting Button of eLinkKVM

1. When LED1 of eLinkKVM turns green, short pressing the Setting Button will result eLinkKVM to appear as storage to attached computer.



2. Open the drive **ELINKCONF**.
3. In **ELINKCONF** drive, there is a configuration file **user\_config.json** that can be edited by any text editor.

```

user_config.json
1  {
2    "ipconfig.slave.dhcp" : false,
3    "ipconfig.slave.ipaddr" : "10.42.0.2",
4    "ipconfig.slave.netmask" : "255.255.255.0",
5    "ipconfig.slave.gateway" : "10.42.0.1",
6    "accessright.admin.netconfig" : true
7  }
8

```

Figure 51. Sample contents of Button of elinkKVM

4. Edit the file following eLinkKVM configuration syntax. Store the file and reset eLinkKVM to use the new configuration.
5. Press **Enter configuration** button again to exit configuration mode.

**NOTE**

elinkKVM provides a dedicated mode of user management for Datacenter where only network user can change network setting. To turn on this mode, change "accessright.admin.netconfig": false

## 5.3. Factory Reset

Support Hardware factory reset

1. Power on elinkKVM and wait until the LED1 is green.
2. Press and hold button for 3 seconds till all 3 LEDs in orange color. Then release the button.

3. Press and hold button (for 3 seconds) again till 3 LED blinks in RED color then release to confirm action factory reset, the board factory resets and then reboots.
4. Wait until the LED1 is green again.

# CHAPTER 6. BOOSTER

**Booster** is a patented technology for eLinkKVM and related ElinkGate products. **Booster** includes software agents for each major OS platform (Windows, Linux and Mac OS) that are installed on controlled computers. When an eLinkKVM device is attached to a controlled computer, the installed software agent is activated to cooperate with a eLinkKVM device using a specialized proprietary protocol. The software agent then utilizes the available computing resources on the remote host to greatly accelerate video input and output processing. As a result, eLinkKVM can deliver higher performance at higher resolutions for an extremely economical price.

To enable **Booster** technology, a user only needs to install the accompanied software components on an existing operating-system installation. For a fresh computer that need a new operating-system installation, a tool called `elinkSetuptool` is provided to help creating install disk images with embedded **Booster** from existing installation ISO files, which is covered in the next chapter.

These disk images can be loaded on eLinkKVM internal storage for remote operating-system installations on new computers with **Booster** enabled through the whole installation process. After the installation process is done, the freshly installed operating system is also pre-installed with **Booster** software to allow a user to continue using **Booster** without any interruption.

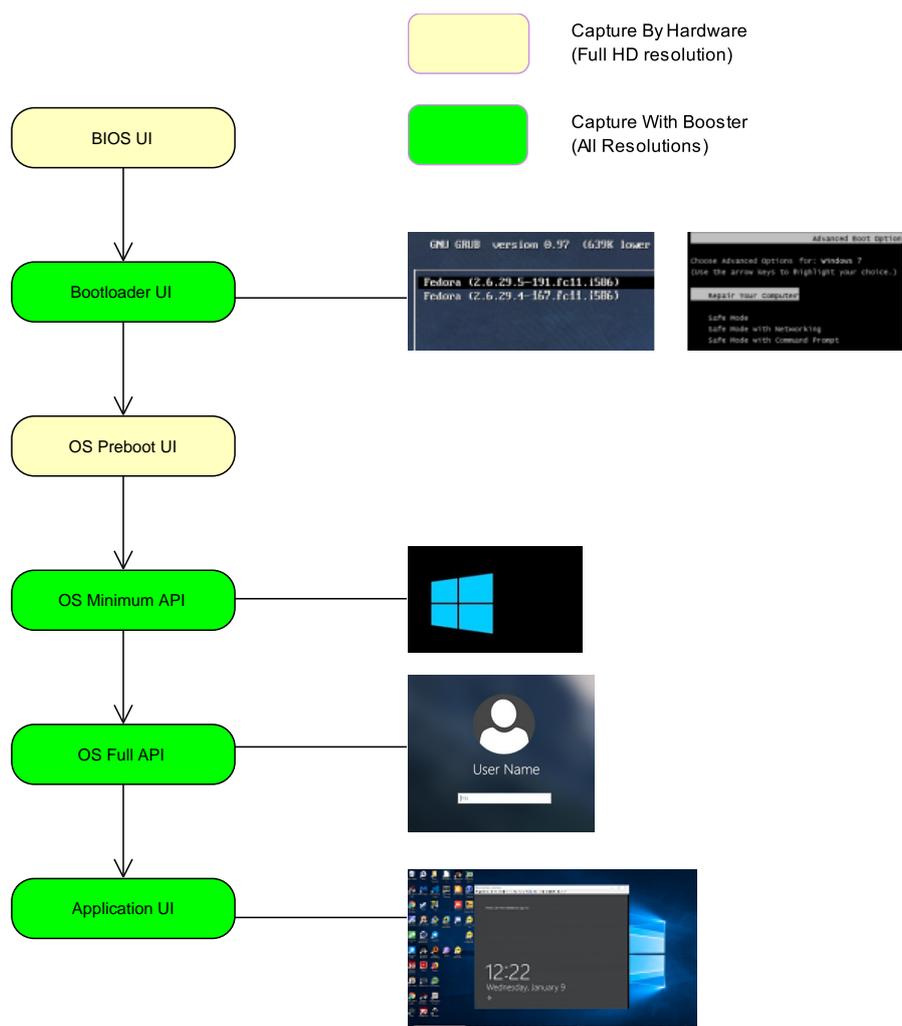


Figure 52. Booster operating scope

Once a **Booster** agent is installed on a respective operating system, simply click the **Booster** icon to start getting remote screen with Booster.

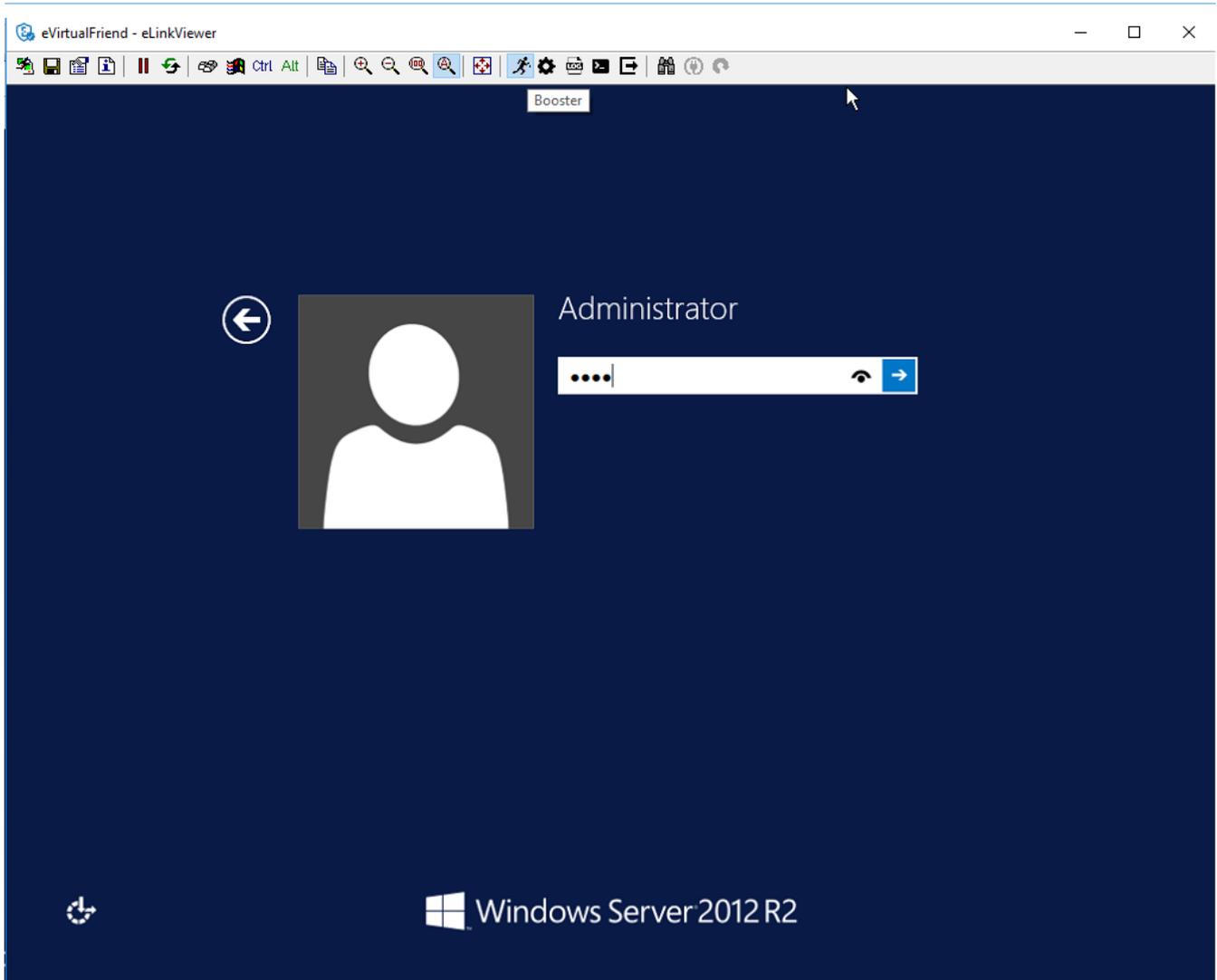


Figure 53. Booster enabled

Screen quality may improve slightly or significantly compared to VGA hardware mode, depends on the quality of the VGA cable. Remote keyboard and mouse interactions become much more responsive.

## 6.1. Booster Configuration

To make it convenient to switch between **Booster** and other modes, eLinkViewer allows a user to customize Auto **Booster** Toggle button. Pressing this button allows a user to switch between a pre-configured Booster mode and the current mode.

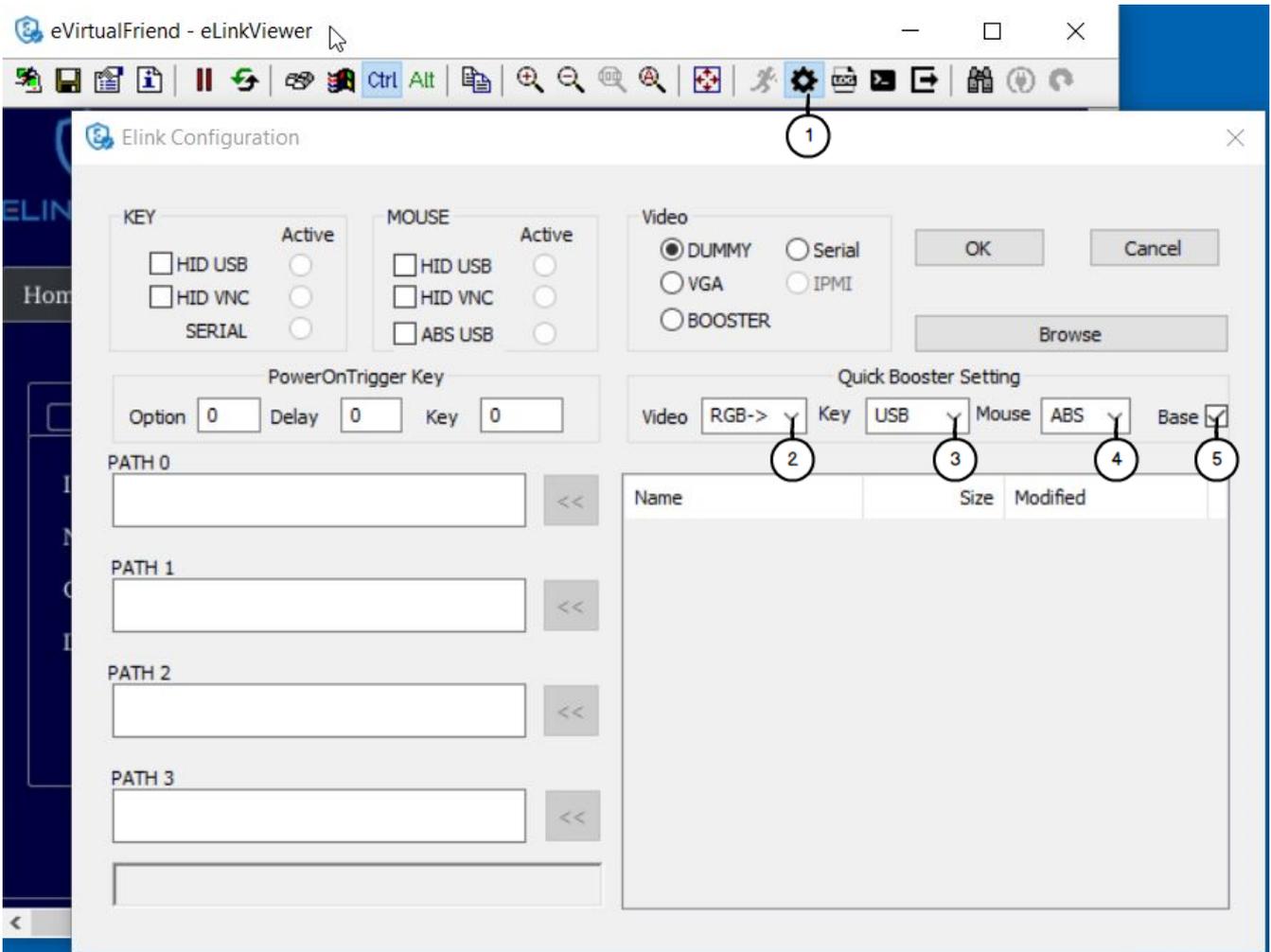


Figure 54. Booster Configuration UI

Steps to configure **Booster** mode:

1. Click **ELink Configuration** button.
2. From **ELink Configuration** select a remote display mode.
3. Similarly, select a Key mode
4. Next, select a Mouse mode
5. Select **Base** to confirm configuration. This configuration is used whenever **Auto Booster Mode** button is clicked.
6. Press OK to save the configuration.
7. Press OK to confirm.

## 6.2. Booster for Windows

Follow these steps to install and use **Booster** on Windows:

- Logged into the computer to be a remote host.
- Download `setup_elinekagent.exe` on the website.
- Click the exe file, follow the installer instructions to install the agent.

- After the installation, run `elinkserver.exe` to enable **Booster** on the remote host.
- On the remote terminal, connect the remmote host with `eLinkViewer`.
- Click **Elink Configuration** → **Browse** then select `A:\floppy.hdd2`.
- Clicking **Booster** to activate remote with Booster.

## 6.3. Booster for Linux (Non supported yet)

Follow these steps to install and use **Booster** on Linu:

- Logged into the computer to be a remote host.
- Download setup package for a distro, .e.g. `.deb` for Ubuntu, `.rpm` for Fedora.
- Install the packages.
- On Ubuntu: `sudo apt-get install elinkagent.deb`
- On Feodra: `sudo apt-get install elinkagent.rpm`
- After the installation, run `elinkserver` to enable **Booster** on the remote host.
- On the remote terminal, connect the remmote host with `eLinkViewer`.
- Click **Elink Configuration** → **Browse** then select `A:\floppy.hdd2`.
- Clicking **Booster** to activate remote with Booster.

## 6.4. Booster for UEFI

By default, when installing the setup packages for an appropriate operating system, **Booster** is also installed for UEFI boltloader. Whenever an operating system enters a non-graphical environment in UEFI, **Booster** can be used without any restriction.

## 6.5. Embbed Booster into operating system setup images with eLinkSetupTool

**Booster** can run in a setup environment of an operating system when it is being installed on a computer. To use this feature, the setup disk images must be recreated with an appropriate **Booster** agent embedded, using `eLinkSetupTool`, a disk creation image tool. **Booster** is enabled as soon as the setup image is loaded by the remote host computer.

### 6.5.1. eLinkSetupTool instalation

#### On Windows

- Download `setuptool.exe`.
- Click the installer and follows the instructions.

#### On Ubuntu

- Download `setuptool.deb`.
- Install it:

```
sudo apt-get install setuptool.deb
```

### On Fedora

- Download [setuptool.rpm](#):

```
sudo dnf install setuptool.rpm
```

After the installation, `vfimg` command should be available globally to be used in a terminal program, e.g. `cmd.exe` on Windows.

**NOTE** | the setup files is above, e.g. [setuptool.exe](#) should be clickable to download

## 6.6. eLinkSetupTool usage

### 6.6.1. Create OS setup image

To create a new disk image, simply run the following command:

```
elinkimg /create-image Win2012.hdd2 /iso Win2012.iso
```

The command produces the following output:

```
Initializing environment...Done.
Start Analyzing image...Found a x86_64 Windows 2012 ISO. Done.
Format HDD image to FAT32...Done formatting.
Generate elinkme_dummy.dat...
- Start sector: 18136, end sector: 19164.
Copy files from ISO to HDD...Done copying.
Installing vfservice to boot.wim...Done copying.
Installing UEFI vfservice...
- Copy vfUEFI bootloader
- Copy vfUEFI driver
- Done copying.
Generate Embedded Hddx at the end of HDD file...Done.
```

The above command creates a new image `Win2012.hdd2` with **Booster** agent embedded from the original Windows setup image `Win2012.iso`.

### Once the new image is created, upload it to eLinkKVM with [File Transfer](#). To use the new image

- Click [Elink Configuration](#) → [Browse](#).
- Browse to the uploaded `Win2012.hdd2` and select it.
- `Win2012.hdd2` is now exposed to the remote host computer as a USB drive and is selectable as a boot device in the BIOS.

### 6.6.2. Create a minimal floppy image

eLinkKVM is already bundled with the minimal image `floppy.hdd2` in its internal storage that can be mounted as a floppy disk drive. For some reason, if the disk is deleted, a user can recreate and reupload the image. To create the floppy image, run the following command:

```
elinkimg /make-floppy floppy.hdd2
```

### 6.6.3. Install UEFI Booster agent to an existing disk image

Aside from OS setup images, there are disk images that contain troubleshooting tools running in the UEFI environment. To create new images with UEFI **Booster** agent, run the following command:

```
elinkimg /install-uefi img.hdd2
```

### 6.6.4. Display version information:

To show the current setuptool version, run the following command:

```
elinkimg /version
```