How to Install ADISRA SmartView Runtime on Linux

Document Information

Software Version:	4.1.2.1
Creation Date:	10 November 24
Last Edit Date:	18 June 25
Version:	V 1.2

ADISRA · 6001 W. Palmer Lane, Suite 350-285 · Austin, TX 78727
Phone: 1-833-5ADISRA (1-833-523-4772) ·. 1-512-993-2949
www.ADISRA.com

Table of Contents

1. Scope 4

2. Summary 4

3. Architecture 5

4. Hardware Requirements 7

5. Software Requirements 8

5.1. Operating System Compatibility 8

5.2. NET Runtime 8

5.3. Web Server 9

6. Modules Available on Linux Runtime 10

6.1. Modules Migration Report 10

6.2. Database Migration Report 11

6.3. Drivers Migration Report 11

7. Linux Distributions 13

8. Installation Prerequisites 14

8.1. Apache installation and configuration 14

8.2. Dongle Library installation and configuration 15

8.3. Linux Alpine distribution - Missing package you need to install manually 16

8.3.1. Installing the prerequisites 16

8.3.2. Installing the ADISRA Runtime 17

8.3.3. Running the ADISRA Remote Tools Service 17

9. ADISRA SmartView Installation 18

9.1. ADISRA SmartView Linux Runtime Deployment 18

10. Remote Management Tool 21

10.1. Overview 21

10.2. Target Tab 23

10.3. Project Tab 24

10.4. License Tab 25

10.5. Log Tab 26

10.6. RemoteToolsServiceLnx execution 26

10.7. Remote Management Tool execution and connection. 27

11. Licensing the RunTimeLnx using the Remote Management Tool 29

11.1. Types of Licenses on Linux 30

11.2. Registering a License 30
11.2.1. Softkey 31
11.2.2. Hard key 32
11.3. Deactivating a License 34
11.4. Certificate of authenticity 36
12. Application Deployment and Execution 37
12.1. Transfer Application 37
12.1.1. Uploading ADISRA SmartView projects Manually 37
12.1.2. Uploading ADISRA SmartView projects to Linux using the Remote Management Tool 37
12.1.3. Uploading web pages to Linux (for Web Viewer access) using the Remote Management Tool 38
12.2. Execute the Runtime Process 39
12.2.1. Linux Runtime execution (command line) 39
12.2.2. Linux Runtime execution (Remote Management Tool) 40
12.2.3. Linux Runtime execution (As a service) 41
12.2.3.1. Configure the appsettings.json 41
12.2.3.2. Create the Systemd service file 41
12.2.3.3. Configure Systemd 42
12.2.3.4. Validate the Systemd Configuration 42
12.2.3.5. Systemd useful commands 43
12.3. Ports 44
12.4. Viewer Connection 44
12.5. Web Connection 45
12.6. Data Watcher Connection 46
13. Linux Runtime troubleshooting 48
13.1. How to check if the Runtime executables have permission to be executed 48
13.2. Permission denied in Web Viewer 48
13.3. Runtime Logs on Linux 48
13.3.1. Analyzing Logs using the Remote Management Tool 49
13.3.2. Analyzing Logs saved locally on the Linux machine. 49

1. Scope

ADISRA SmartView Runtime is the server process responsible for PLC communications, Database connections, Service and Trigger execution, and other functions such as broadcasting data to the client interfaces (Viewer or the Web Clients) in a multithreaded environment.

2. Summary

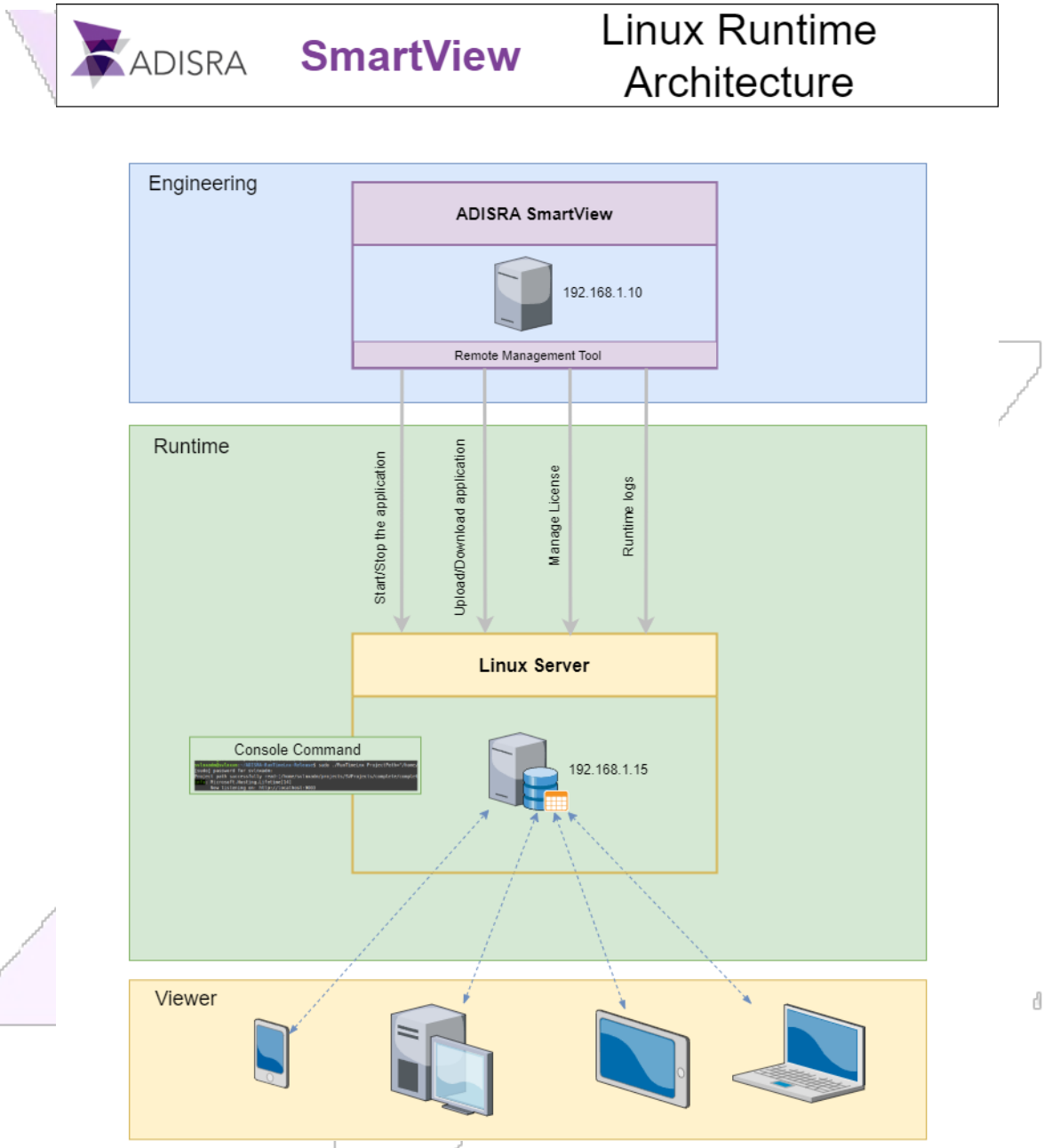
This document provides an overview of the ADISRA SmartView Runtime on Linux. This feature is available from version 4.1.0.0 to the latest versions.

You will learn how to deploy, configure, and execute the ADISRA SmartView Runtime on multiple Linux distributions. In the following chapters you will learn the hardware requirements along with the procedures to install ADISRA SmartView Runtime on Linux distributions, license it, and execute applications using the remote management tool or via command line.

> ***Note:*** *Applications for the ADISRA SmartView Runtime are created in the ADISRA SmartView engineering environment. The ADISRA SmartView engineering environment is only available on Windows and not covered by this document.*

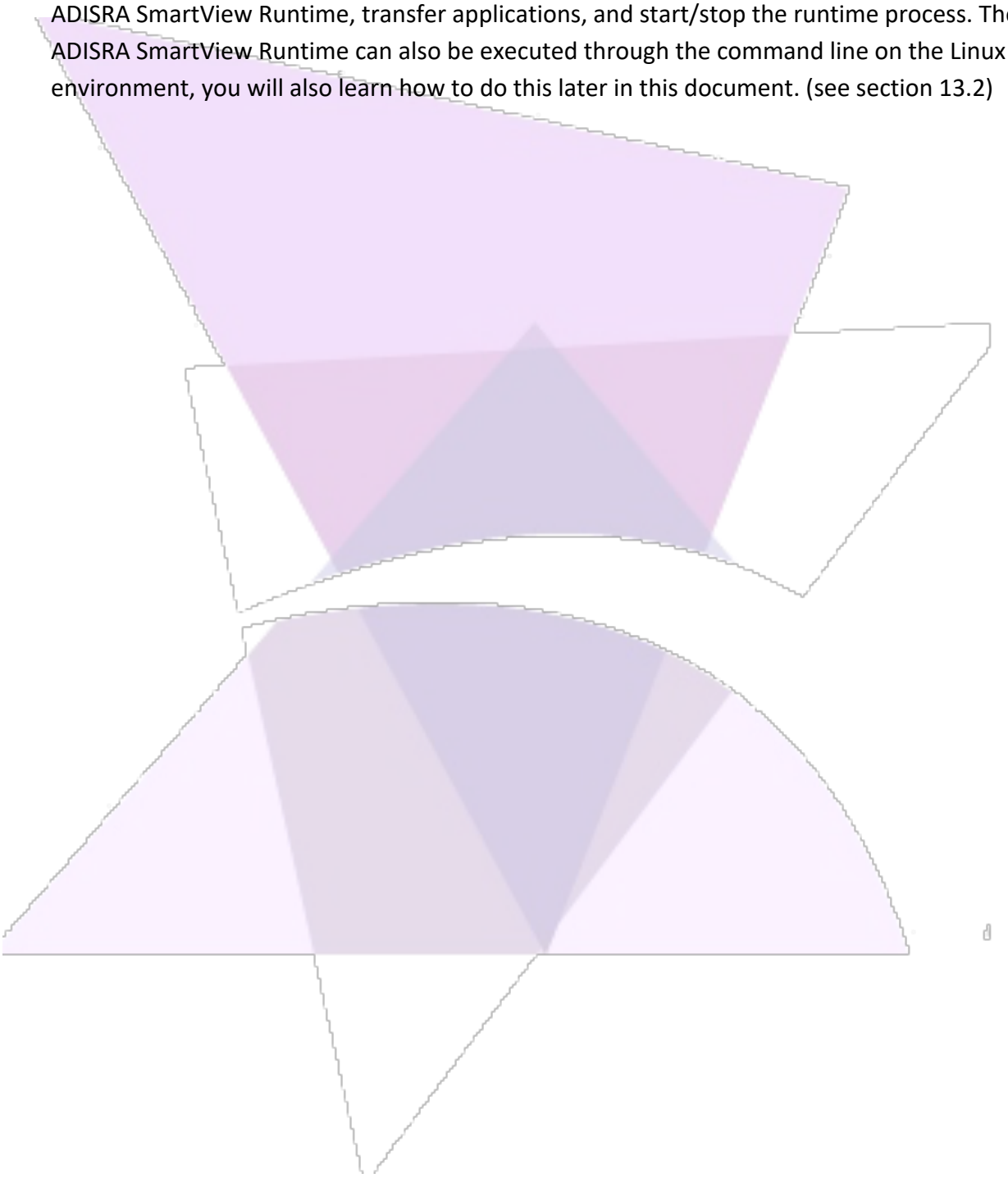
3. Architecture

Let's look at the architecture below before proceeding with the document.



The image above shows the 3 processes levels for ADISRA SmartView. The first block represents the Engineering environment which only runs on Windows operating systems. It has the Remote Management Tool that can be used to manage the ADISRA SmartView Runtime environment on Linux. The installation of ADISRA SmartView Runtime on Linux is

an offline process and will be detailed in the following chapters. As soon as ADISRA SmartView Runtime is installed, and the Remote Management Tool Service is running, you can connect to it from the ADISRA SmartView Engineering environment to license the ADISRA SmartView Runtime, transfer applications, and start/stop the runtime process. The ADISRA SmartView Runtime can also be executed through the command line on the Linux environment, you will also learn how to do this later in this document. (see section 13.2)



4. Hardware Requirements

The hardware requirements will vary depending on the complexity of the application. The ADISRA SmartView Runtime process has many responsibilities such as PLC communication, Database connections, Service and Trigger execution, and other functions such as broadcasting data to the client interfaces (Viewer or the Web Clients) in a multithreaded environment. The following requirements are recommendations from small to mid-applications.

Processor (CPU):

Minimum:

- CPU: 1.44 GHz
- RAM: 4 GB
- Hard Disk (min): 8 GB

Recommended:

- CPU: 1.7 GHz
- RAM: 6 GB
- Hard Disk (min): 20 GB

Network:

- Ethernet or other network connectivity for communication with Remote Viewer, PLC or other servers and devices.

5. Software Requirements

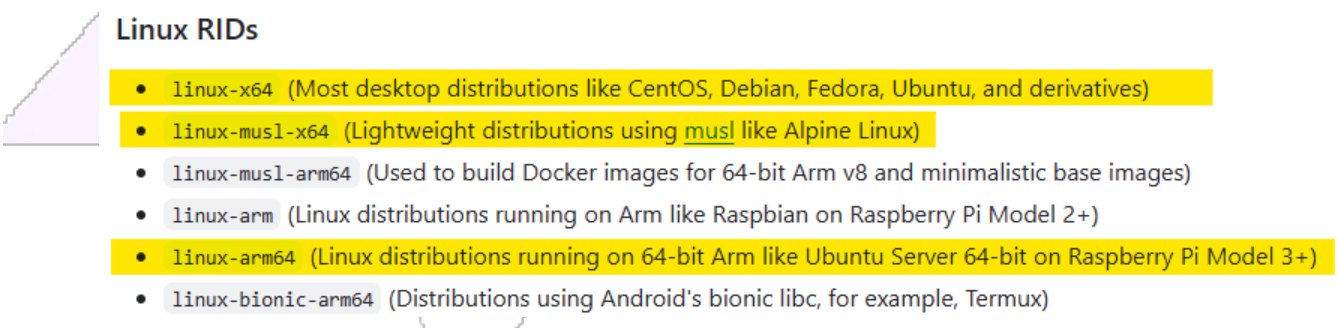
5.1. Operating System Compatibility

The ADISRA SmartView Runtime is designed to run on various Linux distributions, including but not limited to:

- Arch Linux (CLI)
- Elementary OS (GUI)
- Zorin OS (GUI)
- Lubuntu (GUI) - Ubuntu-based
- Pop!_OS (GUI) - Ubuntu-based
- Alpine Linux (CLI)
- Alpine Standard (CLI)
- Alpine Virt (CLI)
- CentOS (Both CLI and GUI)
- Debian (GUI)
- Linux Mint (GUI)
- Ubuntu Desktop (GUI)
- Ubuntu Server (CLI)

5.2. NET Runtime

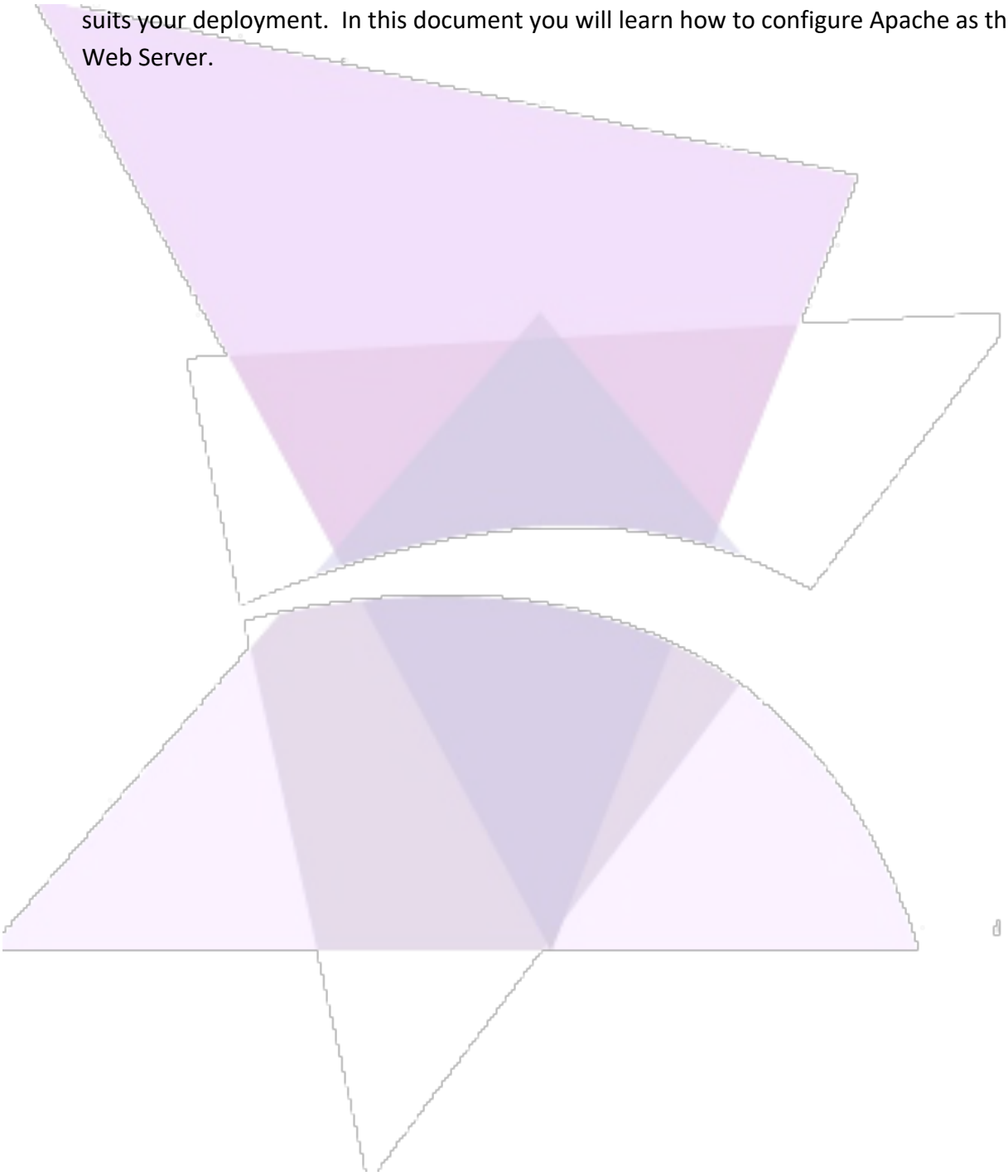
The ADISRA SmartView Runtime for Linux has different builds for different architectures, and each build contains its relevant .net6 runtime dependencies. The highlighted runtime identifiers below illustrate the current available builds for ADISRA SmartView.



ADISRA SmartView Runtime for Linux is a self-contained .net6 application, which means there is no need to install the .net6 on the Linux distribution, it will be already packed in the tarball provided by ADISRA, keeping the installation simple and platform specific.

5.3. Web Server

For the ADISRA SmartView application to serve web pages (so users will have access to the interface through the web client), it is necessary to configure a web server that best suits your deployment. In this document you will learn how to configure Apache as the Web Server.



6. Modules Available on Linux Runtime

This section provides an overview of the runtime's modules and functionalities that have been migrated to Linux, as well as those that remain unimplemented or not supported.

6.1. Modules Migration Report

Modules	Windows	Linux
Alarms	✓	✓
Audit	✓	✓
Database	✓	✓
Drivers	✓	✓
Language	✓	✗
Recipe	✓	✓
Recipe [Smart Recipe]	✓	✗
Redundancy	✓	✗
Remote Tools	✓	✓
Reports	✓	✗
Security	✓	✓
Services	✓	✓
Simulation	✓	✓
System Function Library	✓	✓
Tag History	✓	✓
Triggers	✓	✓
Tunneling	✓	✓
User Function Library	✓	✓
Viewer [Native] - Running on Linux	✓	✗

Viewer [Remote] - Running on Windows	✓	✓
Viewer [Web] - Running on Browser	✓	✓

6.2. Database Migration Report

Database	Windows	Linux
Microsoft Access Database File	✓	✗
Microsoft ODBC Data Source	✓	✗
Microsoft SQL Server	✓	✓
Oracle Database	✓	✓
PostgreSQL	✓	✓

6.3. Drivers Migration Report

Driver	Windows	Linux
Allen Bradley	✓	✓
Bacnet	✓	✓
Mitsubishi	✓	UNDER BETA
Modbus-FC23	✓	✓
Modbus	✓	✓
Mqtt	✓	✓
Omron-CIP	✓	UNDER BETA
Omron-FINS	✓	UNDER BETA
OPC DA Client	✓	✗
OPC UA Client	✓	✓

OPC UA Server	✓	✓
OpenWeather	✓	✓
Siemens-TIA	✓	✗
Siemens	✓	✓
Snmp	✓	✓
TwinCat-ADS	✓	UNDER BETA
TxRx	✓	✓
WebApi	✓	✓

7. Linux Distributions

This section provides an overview of the Linux Distribution and their respective ADISRA Distributions that have been adapted on Linux.

Linux Distribution	ADISRA Distribution
ArchLinux	RHEL
ElementaryOS	Debian
Lubuntu	Debian
POP	Debian
Zorin	Debian
Alpine Extended	Alpine
Alpine Standard	Alpine
Alpine Virt	Alpine
CentOs	RHEL
Debian	Debian
Linux Mint	Debian
Ubuntu Desktop	Debian
Ubuntu Live Server	Debian

8. Installation Prerequisites

This topic provides information to ensure your system meets the necessary prerequisites before installation. If your project does not require web pages, you can skip the Apache installation.

8.1. Apache installation and configuration

To use the Web Viewer to connect to the ADISRA SmartView Runtime on a Linux workstation, it's necessary to install and configure a web server. As an example, we will use the Apache2 Web Server.

To install Apache2 on Ubuntu/Debian Linux, open a terminal and type the following command:

```
sudo apt update
sudo apt install apache2
```

To install Apache2 on CentOS/Fedora Linux, open a terminal and type the following command:

```
sudo apt update
sudo yum install httpd
```

Or

```
sudo apt update
sudo dnf install httpd
```

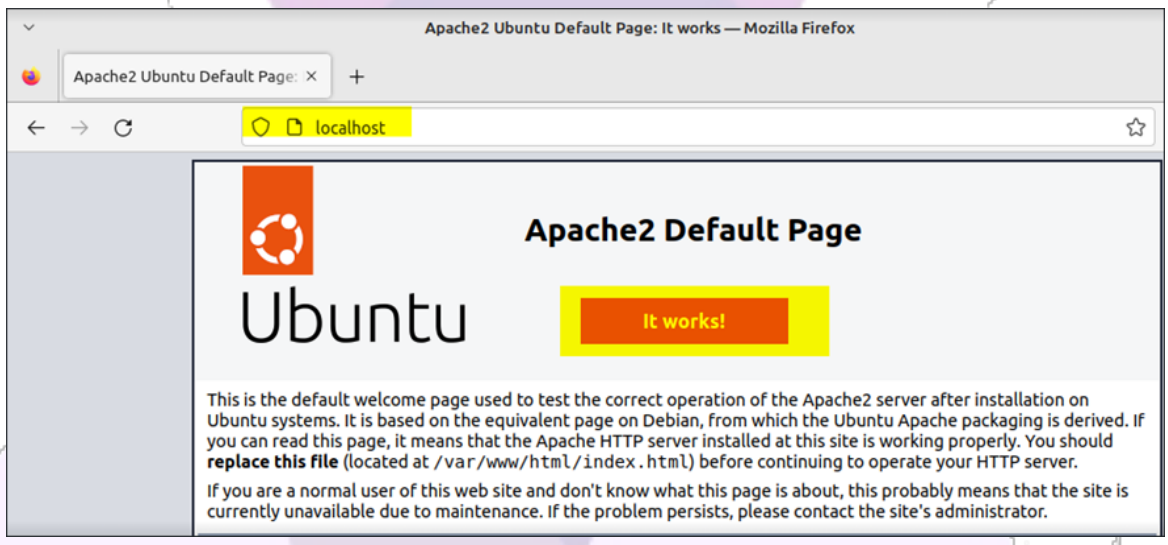
To install Apache2 on Alpine Linux, open a terminal and type the following command:

```
sudo apt update
sudo apk add apache2
```

Once Apache is installed, you can test it by accessing the following URL in your browser:

```
http://localhost/
```

This will display the default Apache landing page, indicating that Apache is successfully installed and running. In case it is a Linux server, you can test it using the network replacing localhost for the appropriate IP address.



8.2. Dongle Library installation and configuration

If udev or systemd-udev is not already installed, the following commands will install the library necessary for enabling a hard key on Linux environments:

```
Debian-based (Debian, Ubuntu):
```

```
sudo apt update
sudo apt install udev

Alpine Linux:
apk update
apk add udev

Red Hat-based (CentOS, Fedora):
sudo yum update
sudo yum install udev
```

If you will be using a softkey instead, you don't need to install udev.

8.3. Linux Alpine distribution - Missing package you need to install manually

While executing the ADISRA SmartView Runtime Linux on an Alpine distribution, you may notice issues related to the Alarms module. The problem is solved by installing a package, which, in other Linux distributions, seems to be already installed. Since Alpine is a lighter distribution, it was necessary to include it manually.

Package: tzdata

Therefore, please pay attention to the commands below (9.3.1):

8.3.1. Installing the prerequisites

Run the commands below to install the tzdata package.

```
su
apk update
apk add bash icu-libs krb5-libs libgcc libintl libstdc++ zlib udev tzdata
```

8.3.2. Installing the ADISRA Runtime

Note: *These commands can also be found in section 9, so if you prefer, you can skip it to the next chapter.*

```
cat ADISRA-SmartView-4100-Alpine.tar.gz | gunzip | tar -xvf -
cd ADISRA-SmartView-4100-Alpine
chmod +x RemoteToolsServiceLnx
chmod +x RunTimeLnx
```

8.3.3. Running the ADISRA Remote Tools Service

Note: *These commands can also be found in section 10, so if you prefer, you can skip it to the next chapter.*

```
sudo ./RemoteToolsServiceLnx
```

9. ADISRA SmartView Installation

Considering that not all production Linux hosts have access to the internet, the ADISRA SmartView Linux Runtime will be offered as a tarball (tar.gz) and a set of instructions, but it will be explained in the next chapter.

9.1. ADISRA SmartView Linux Runtime Deployment

Follow the steps below to install ADISRA SmartView Linux RunTime in a Linux target machine.

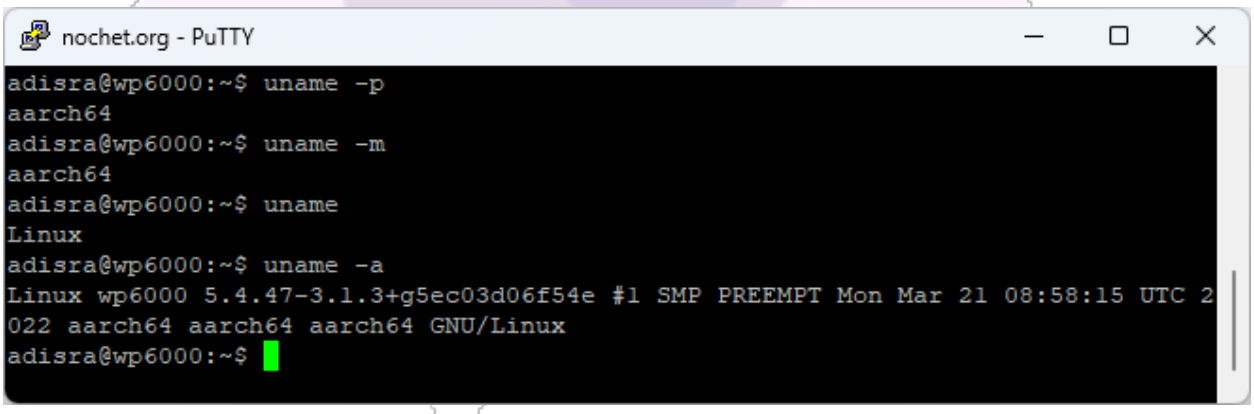
1. Download the tarball (ADISRA-RunTimeLnx-Release.tar.gz) from the ADISRA Web site:

   It is important to identify the correct architecture:

   Before proceeding with the installation of the ADISRA Runtime, it's important to determine the architecture of the Linux operating system.

   Note: *The "arch" command is a good way to determine the host architecture.*

   The example below shows an ARM64 architecture Linux distribution. You can get the architecture by running the command “uname - a”. It returns aarch64 (ARM64).



Identifying the architecture (ARM64, X64), you are ready to download the correct ADISRA SmartView Linux Runtime and proceed with the installation. The download page also provides a different installer for the Alpine distribution.

2. Copy the tarball to the Linux environment

   First let's define the installation folder "/home/svlnxadm/". You will need to review the commands below to make sure they match your paths and file name.

   Make sure that the tarball is available in the Linux workstation. You need to download it from ADISRA SmartView download page as mentioned in the previous topic:

```
/home/svlnxadm/ADISRA-RunTimeLnx-Release.tar.gz
```

3. Unpack the tarball:

```
cd /home/svlnxadm

cat ADISRA-RunTimeLnx-Release.tar.gz | gunzip | tar -xvf -
```

4. Make sure that the following directory has been created:

```
ls -ld /home/svlnxadm/ADISRA-RunTimeLnx-Release
```

```
svlnxadm@svlnxvm:~$ ls -ld /home/svlnxadm/ADISRA-RunTimeLnx-Release
drwxrwxr-x 19 svlnxadm svlnxadm 40960 Mar 25 09:48 /home/svlnxadm/ADISRA-RunTimeLnx-Release
```

5. Make sure that the following files have permission to be executed:

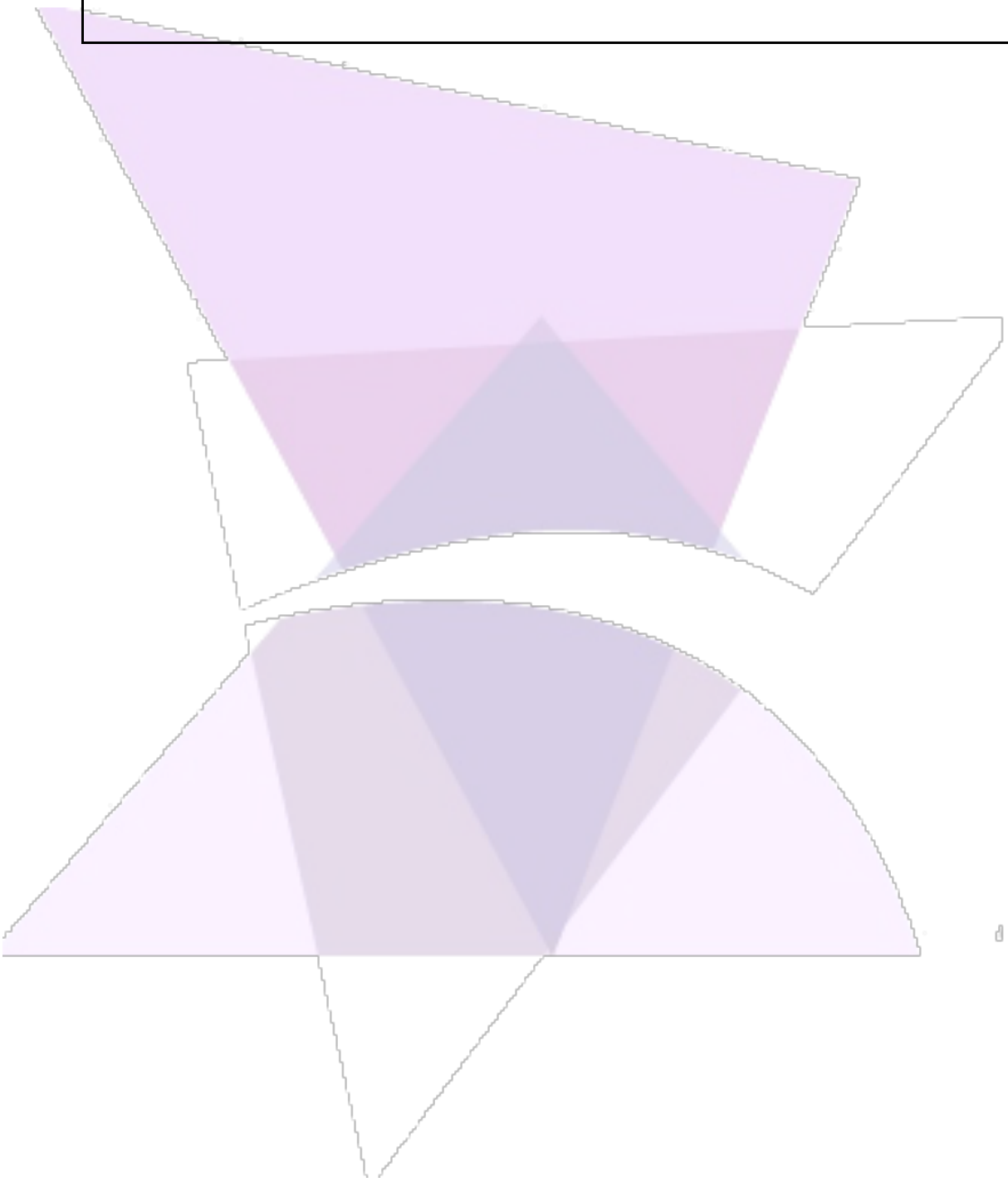
```
cd /home/svlnxadm/ADISRA-RunTimeLnx-Release

ls -l RemoteToolsServiceLnx RunTimeLnx
```

```
svlnxadm@svlnxvm:~/ADISRA-RunTimeLnx-Release$ ls -l RemoteToolsServiceLnx RunTimeLnx
-rwxrwxr-x 1 svlnxadm svlnxadm 142840 Mar 25 09:39 RemoteToolsServiceLnx
-rwxrwxr-x 1 svlnxadm svlnxadm 142840 Mar 25 09:39 RunTimeLnx
```

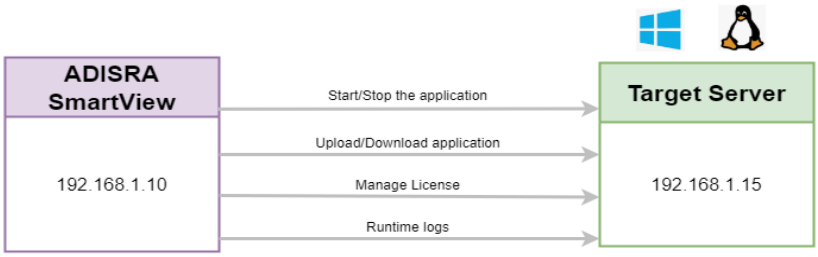
6. If they don't have permission to be executed, issue the following command:

```
chmod 775 RemoteToolsServiceLnx RunTimeLnx
```



10. Remote Management Tool

Now that ADISRA SmartView Linux Runtime and its prerequisites have been installed, it's time to execute it. But before that, let's review the Remote Management Tool (RMT). This tool is part of the ADISRA SmartView Engineering environment and allows the users to create a connection between two computers running ADISRA SmartView. The source computer must be a Windows machine, and the target machine can be Windows or Linux machine.



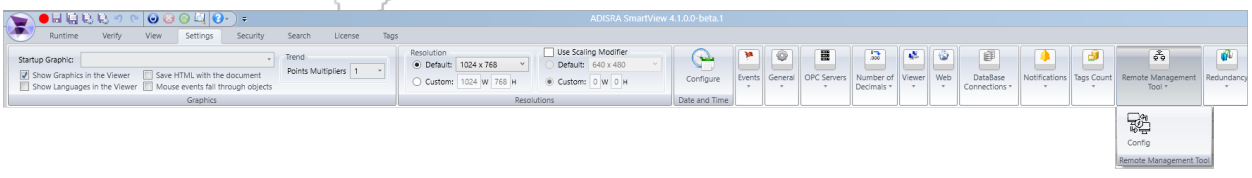
To create a connection, it is mandatory to execute a service in the Linux target machine called RemoteToolsServiceLnx. It opens a safe communication channel with the ADISRA SmartView Engineering environment. After the connection is created, you can transfer applications, start/stop them remotely, license the ADISRA SmartView on the Linux machine and some other functionalities.

10.1. Overview

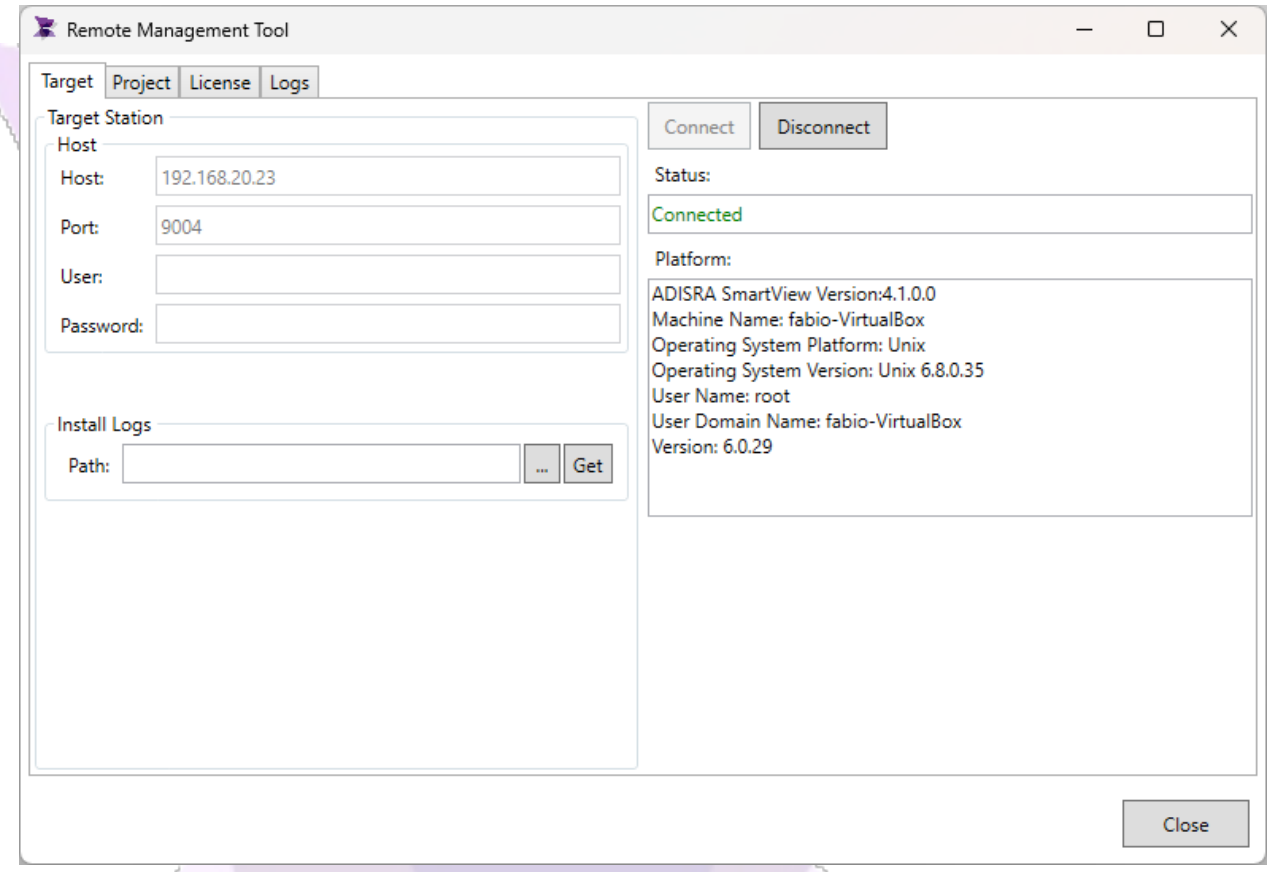
1. Run the RemoteToolsServiceLnx on the Linux target machine

```
svlnxadm@svlnxvm:~/ADISRA-RunTimeLnx-Release$ sudo ./RemoteToolsServiceLnx
```

2. Open ADISRA SmartView Engineering environment and locate the Remote Management Tool (Top Menu-> Settings-> Remote Management Tool)



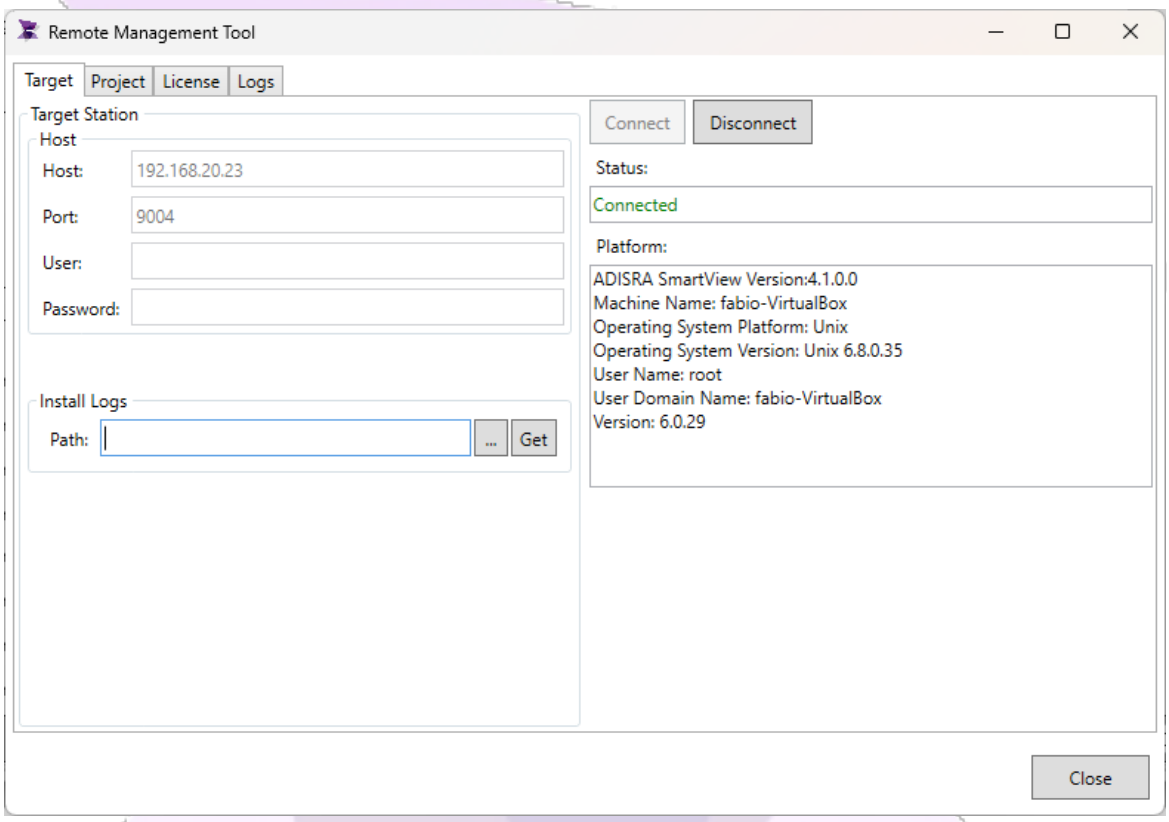
3. Add the target system IP address and the port number exposed in the RemoteToolsServiceLnx and click connect.



4. After a connection is established, you will have access to different functionalities such as transfer application, license management, remote logs, start/stop execution of the remote application. Please refer to the next sections for more information.

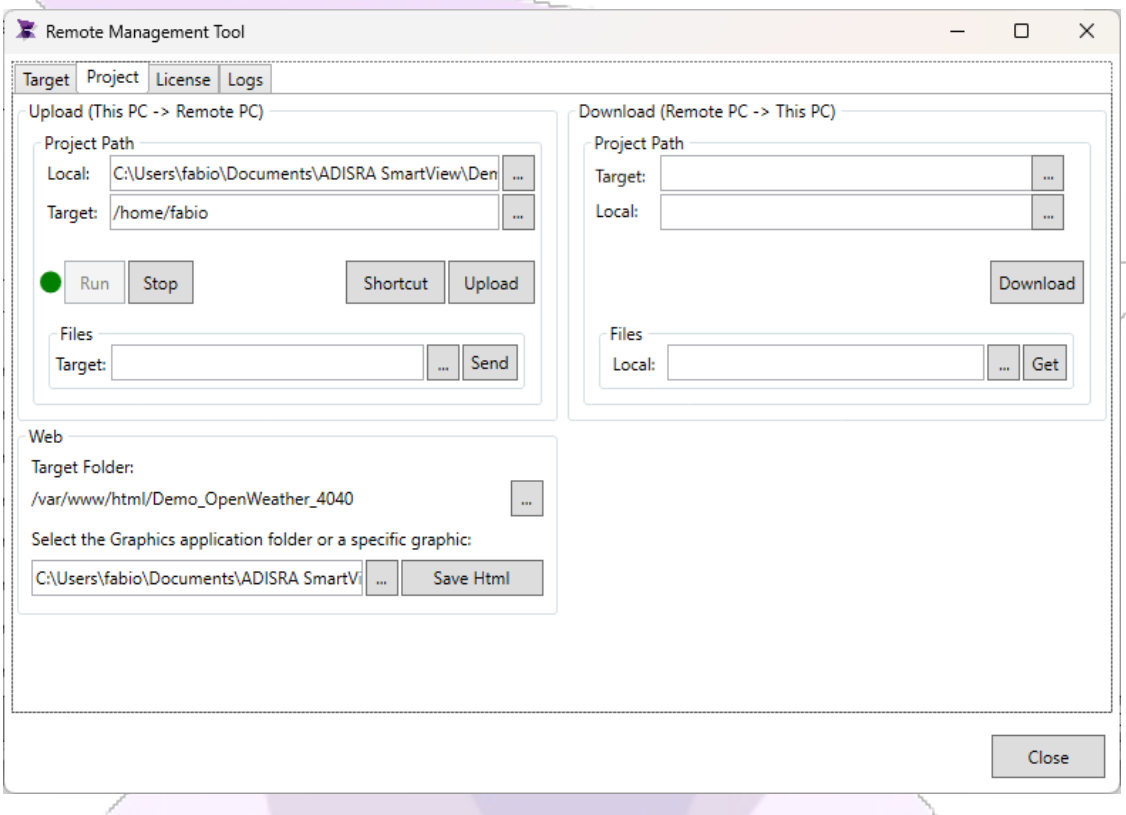
10.2. Target Tab

This is the initial tab, which allows the user to connect to a remote ADISRA SmartView system. As soon as it connects, the other tabs are enabled. At any point, if the Remote Management Tool window is closed, it closes the connection with the remote management tool service.



10.3. Project Tab

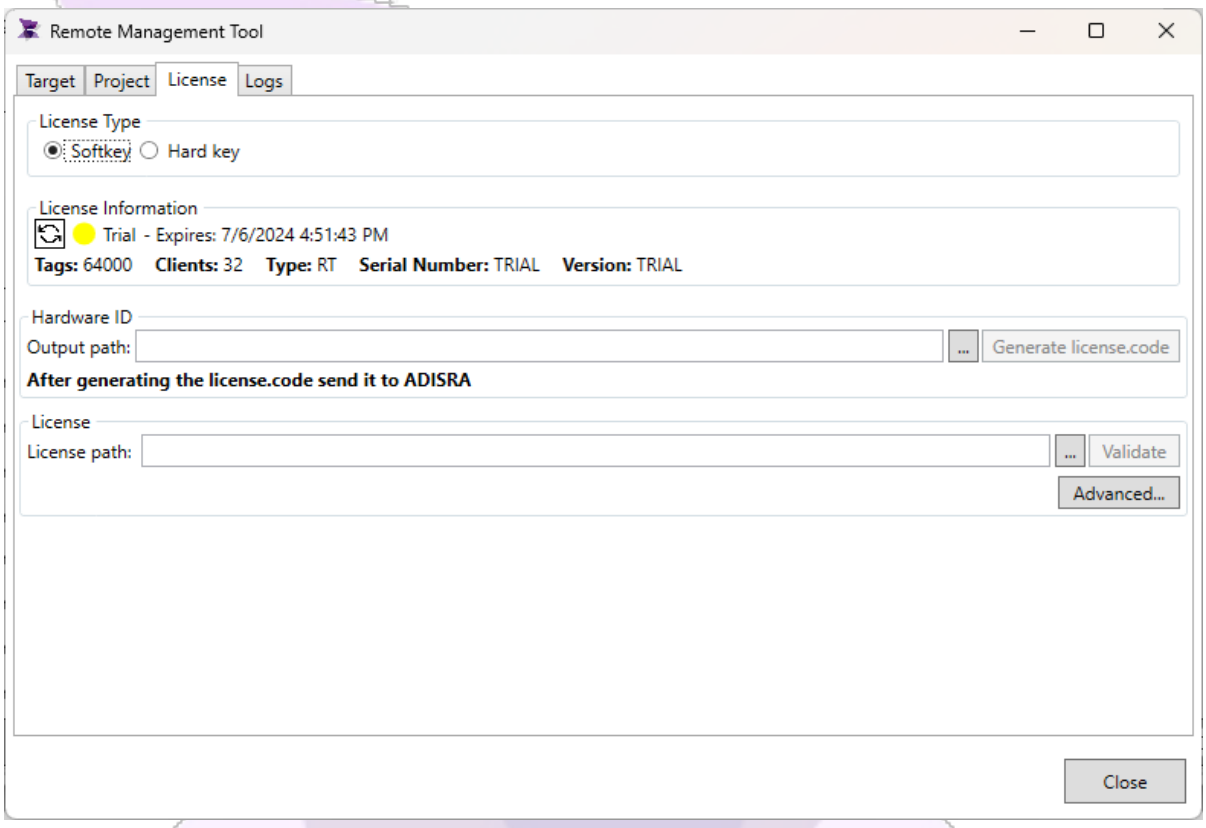
To send the application to the target Linux machine, you will just need to select the target path and it will create a new folder containing the application. This tab also allows sending files and html pages to the target system. In the example below the target folder is used by Apache to serve the web pages.



Another option available is to download applications or files located in the Linux system.

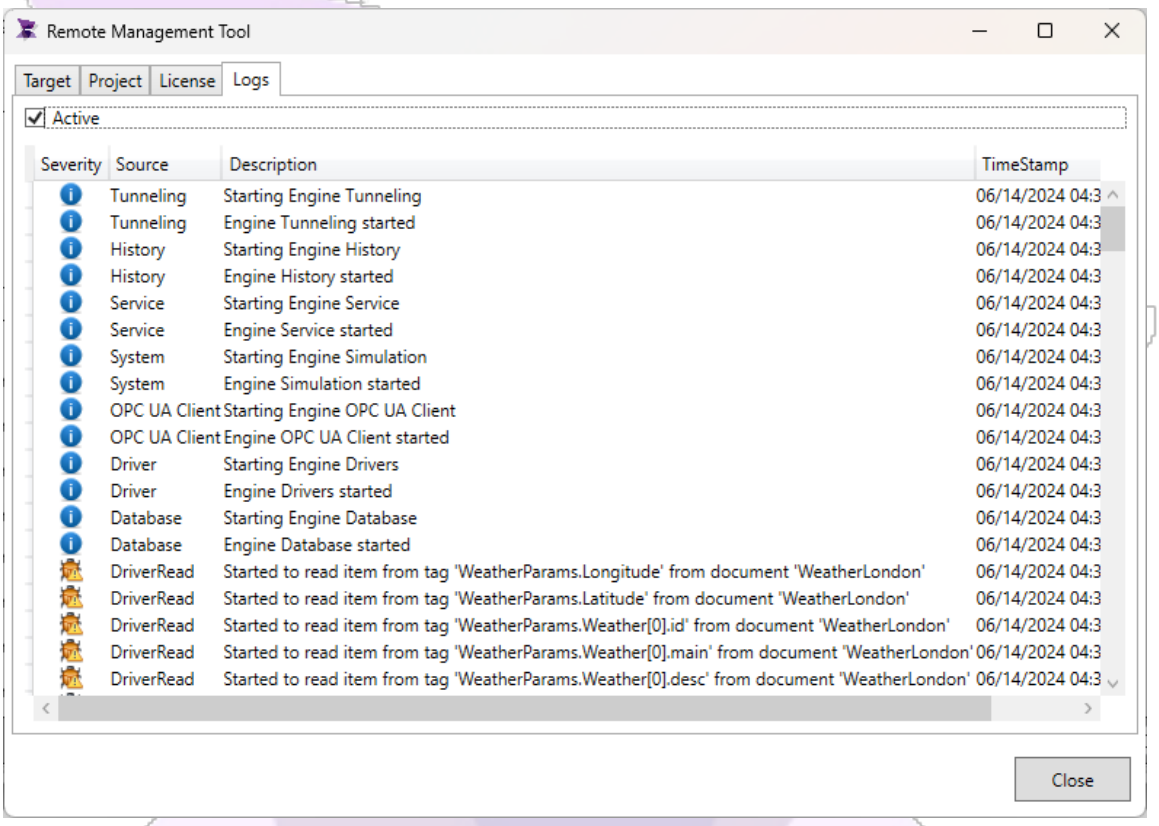
10.4. License Tab

You will use this tab to manage the ADISRA Linux Runtime licenses. It remotely connects to the security system allowing you to license your application or change the license type for example. (see section 12)



10.5. Log Tab

You can debug the runtime application using the Log tab, but it's important to mention that the log filters, whenever changed, require the application to be re-uploaded to the Linux machine.



10.6. RemoteToolsServiceLnx execution

To execute the ADISRA SmartView Linux Runtime using the Remote Management Tool (remote windows machine) we need to connect to the Remote Tools Service. To do so, let's go to the folder containing all the ADISRA SmartView Runtime Linux binaries and execute the RemoteToolsServiceLnx as in the example below. Please change the path to match your installation path.

```
cd /home/svlnxadm/ADISRA-RunTimeLnx-Release
sudo ./RemoteToolsServiceLnx
```

```
svlnxadm@svlnxvm:~/ADISRA-RunTimeLnx-Release$ sudo ./RemoteToolsServiceLnx
[sudo] password for svlnxadm:
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://0.0.0.0:9004
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/svlnxadm/ADISRA-RunTimeLnx-Release/
```

If the above information is displayed, you should be able to connect to it using the Remote Management Tool on ADISRA SmartView engineering environment.

10.7. Remote Management Tool execution and connection.

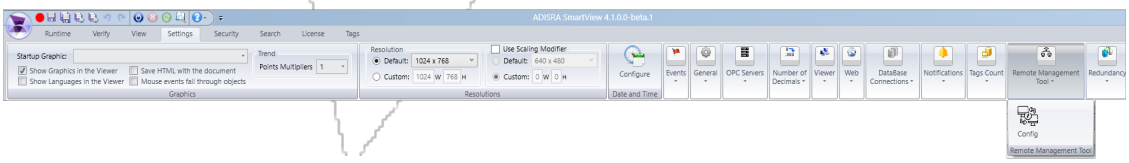
1. Before launching the Remote Management Tool, use the following command in the Linux Workstation to get its IP address:

hostname -I

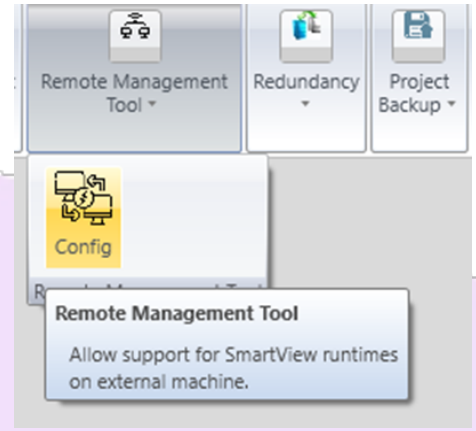
```
svlnxadm@svlnxvm:~/ADISRA-RunTimeLnx-Release$ hostname -I
192.168.0.145 2804:14d:4280:5988::1001 2804:14d:4280:5988:8813:6862:9acc:6477 2804:14d:4280:5988:7642:6b9a:1039:a783
```

2. In the Engineering Workstation (Windows), where ADISRA SmartView is installed and running, execute the RemoteManagementTool:

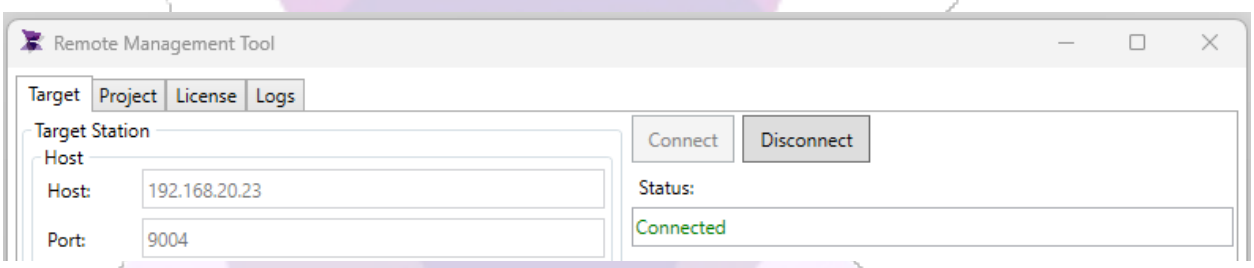
3. Open ADISRA SmartView engineering environment and locate the Remote Management Tool (Top Menu-> Settings-> Remote Management Tool)



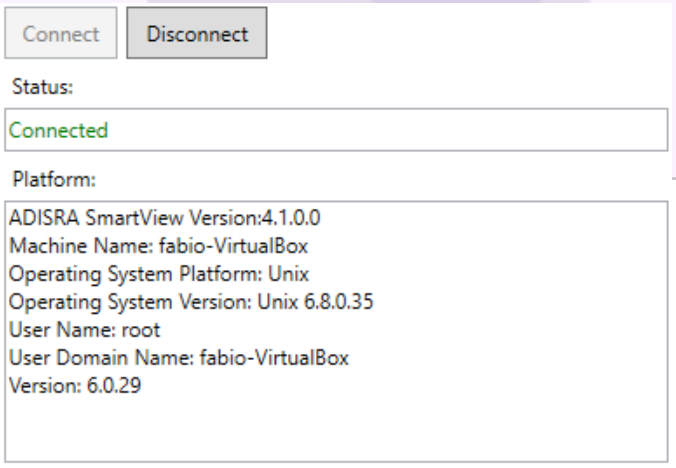
4. Click on the Config button to open the RMT.



5. Select the Target tab and fill the “Target Station Host” with the Linux server IP address and hit the “Connect” button. Please confirm the port number since it can be configured to a different port:



The image below shows the expected result:



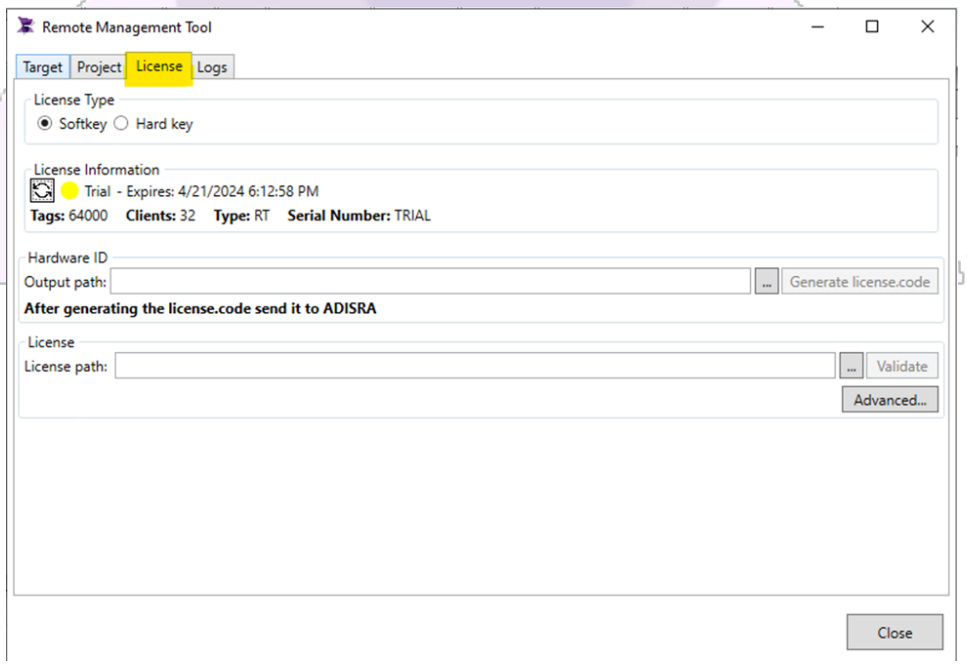
11. Licensing the RunTimeLnx using the Remote Management Tool

The ADISRA SmartView license authorizes a single machine to execute the ADISRA SmartView software. The single license cannot be used on multiple machines since it is associated with the unique components of the specific machine that issued the licensing code.

The license code is generated using a machine's hardware identification, which is based on 3 hardware components: CPU, HDD, and Mother Board. The component-based license ensures that a single license will only enable the software to work on a specific machine. If the user plans to change any of the three hardware components, we recommend the user contact ADISRA to deactivate the license and then have it reactivated after the hardware component has been replaced.

The ADISRA team securely generates each license based on a license code generated by the customer. The license generation steps are explained in detail later in this document. A license can be deactivated at any time, but if it is deactivated, the customer must contact ADISRA to reactivate the license.

Information about your license is available in the Remote Management Tool on the License Tab. The information available is the license status, license type, the maximum number of clients and tags, and the license serial number. The license status is represented by a circle color: green means activated, yellow means trial version, and orange means there are five days remaining until trial expiration.



11.1. Types of Licenses on Linux

The License System has two license types on Linux:

Trial License (TRIAL)

This license is provided upon installation of ADISRA SmartView. After 30 days, the license will expire, requiring a permanent Runtime License. It is possible to run the application in Runtime mode with a Trial License. However, there is a time limit of 2 hours, that is, the Runtime mode will be closed if it exceeds the time of 2 hours.

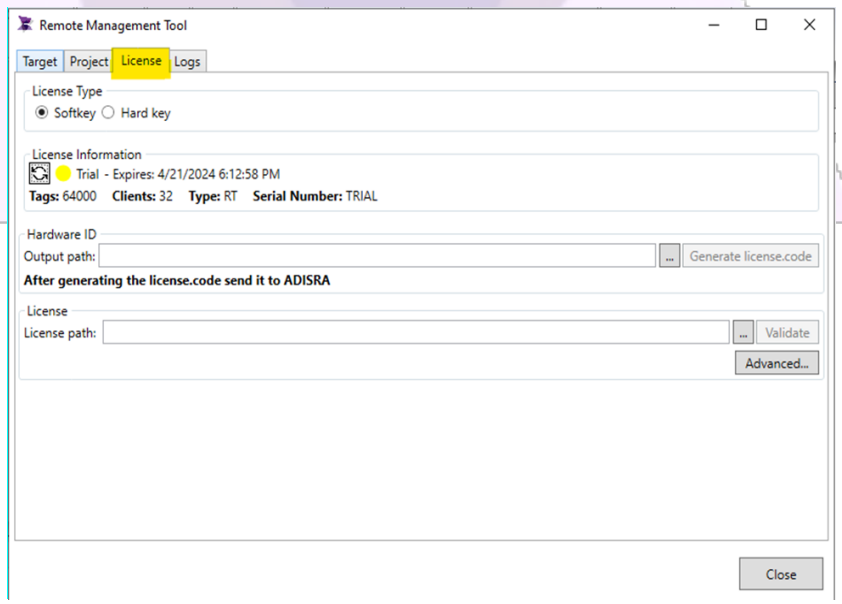
Runtime License (RT)

This license does not allow access to the ADISRA SmartView Engineering environment to develop the application, just run the application in Runtime mode with unlimited time.

11.2. Registering a License

ADISRA SmartView gives the user a 30-day development trial when it is first installed. When the 30 days expire, a permanent license will need to be purchased to run the software. The instructions below will show the steps to register a new license:

1. Select the License tab in the Remote Management Tool:

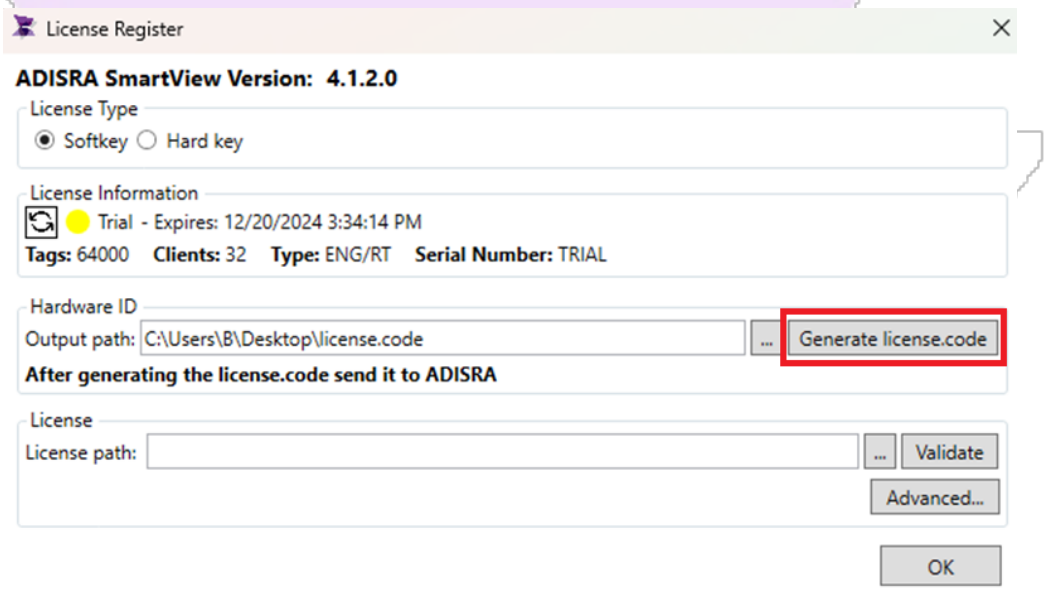


There are two license types: a Softkey or a Hard Key.

11.2.1. Softkey

The softkey involves receiving a digital license, a ".lic" file, from ADISRA. Follow the steps below to generate and validate the softkey license.

1. Click the button labeled "Generate license.code" to generate file containing this machine's unique Hardware ID information. The file location will be displayed in the text box called Output Path.



Note: *The user can change the Output Path by clicking on the ... button next to the Output Path text box. Please do not change the file name.*

2. Attach the text file named "license.code" to an email and send it to info@adisra.com.

The ADISRA will verify your purchase and send back a License Key File that matches the Hardware ID. Download and save the license file to your storage drive. It is important to remember the license file location.

3. Enter the license key file location in License Path text box or browse to locate it by clicking on the ... button next to the License path and then click the Validate button.

4. You will be prompted to confirm the operation once the program accepts or validates your License Key.

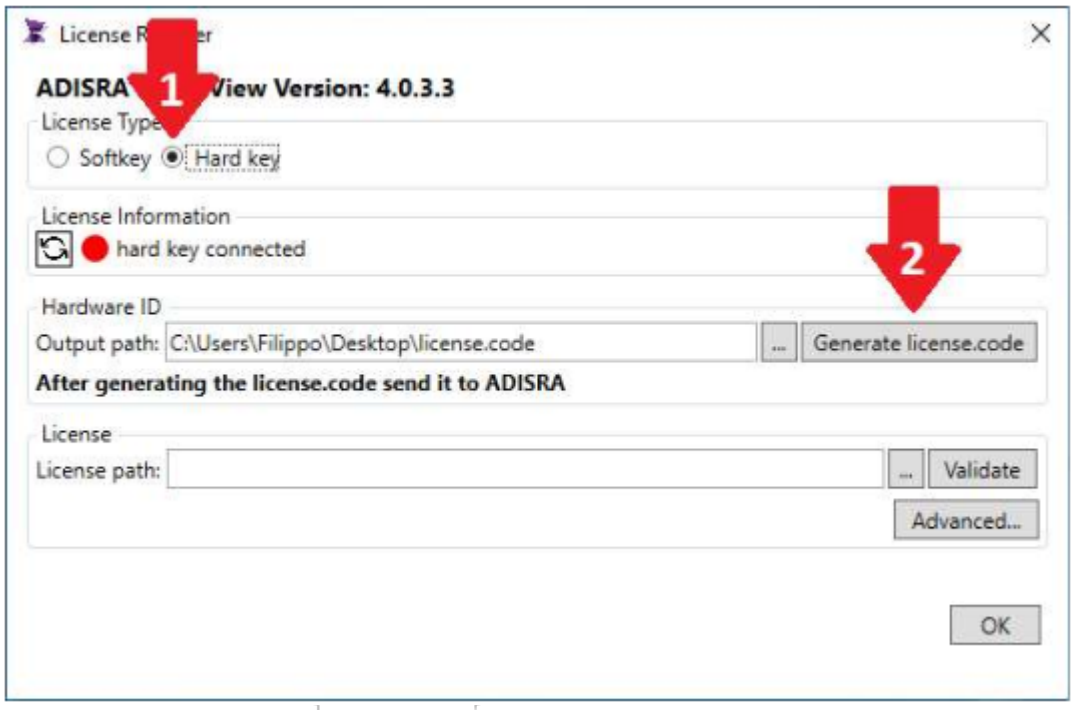
Your machine is now licensed and ready to use ADISRA SmartView

Close the Register License window and run ADISRA SmartView again.

11.2.2. Hard key

The hard key license is used for a license located on an external removable device called a dongle. Follow the steps below to generate and validate the hard key license.

1. Connect the dongle to the machine's USB port;
2. Select the License tab in the Remote Management Tool.
3. In the "License Type" field, select the "Hard key" option and then click on the "Generate license.code" button.



Note: *If you do not have a dongle attached to the machine the License Information will show "No hard key connected".*

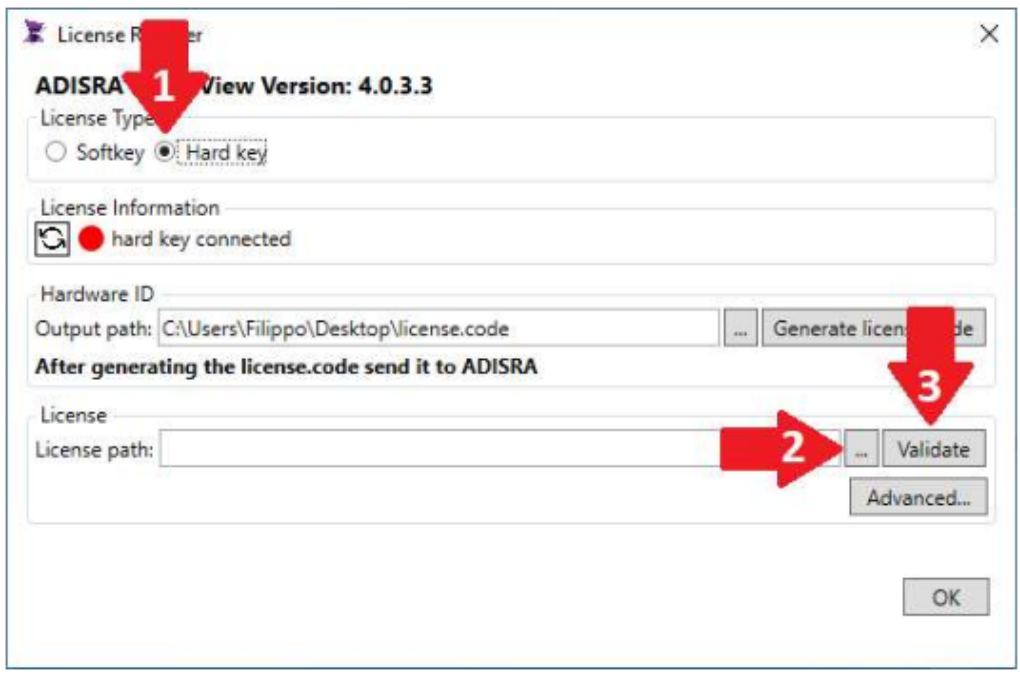
4. A file will be generated called “license.code”

5. Send an email to “info@adisra.com” attaching the Hard key activation file (license.code). In the body of the email, identify the current version of ADISRA SmartView and the Serial Number of the license, which can be found in the certificate of authenticity. (see section 12.4)

6. Once this is done, the ADISRA team will activate the Hard key license. A file called “license.lic” will be sent to your email, in which you have the Hard key activation code.

7. Now with the “license.lic” file in your possession, open the License tab in the Remote Management Tool again.

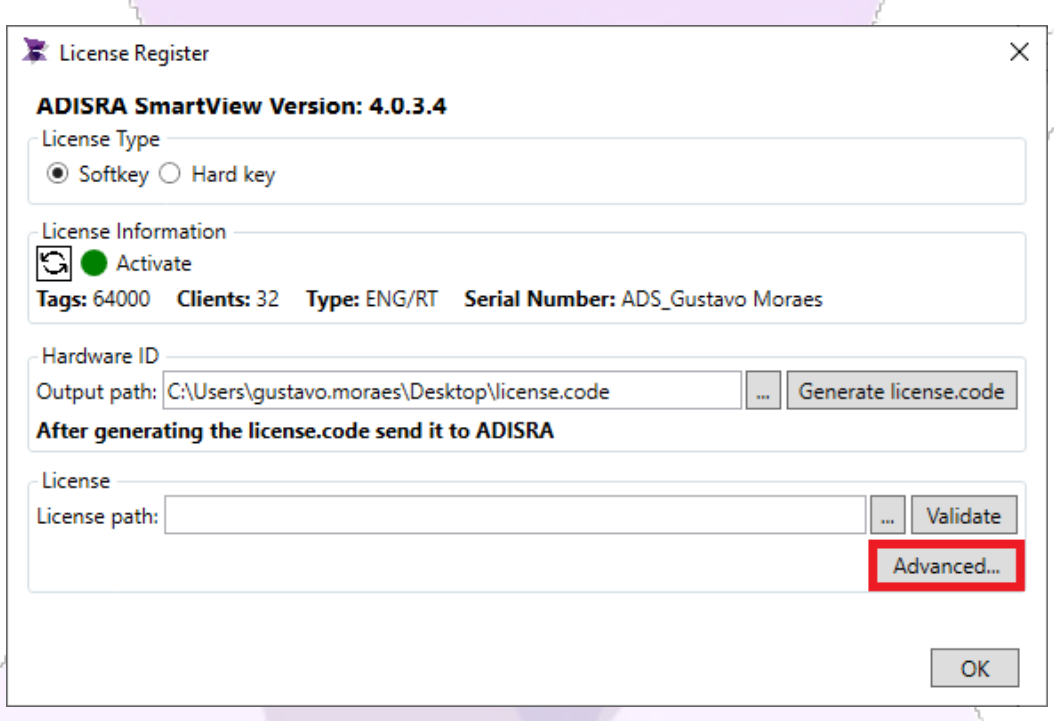
8. In the “License Type” field, select the Hard key option. Then click on the “...” button and point to the “license.lic” file. After clicking the "Validate" button, your license is ready to use. You have successfully validated your Hard key.



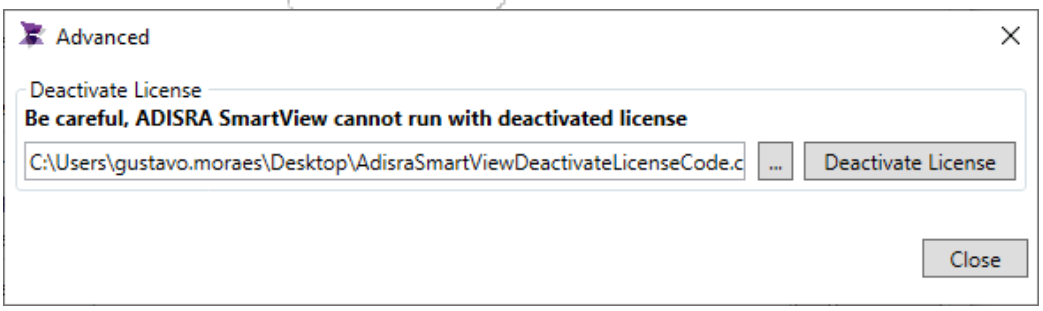
11.3. Deactivating a License

Each ADISRA SmartView license is registered to a specific machine. To use the Runtime license on another Windows or Linux machine, it will be necessary to deactivate the current license. To execute the deactivation procedure, please follow the steps below:

1. In the License tab in the Remote Management Tool, press the "Advanced…" button:



2. It will open a new window so you may decide where the deactivated license file will be generated:

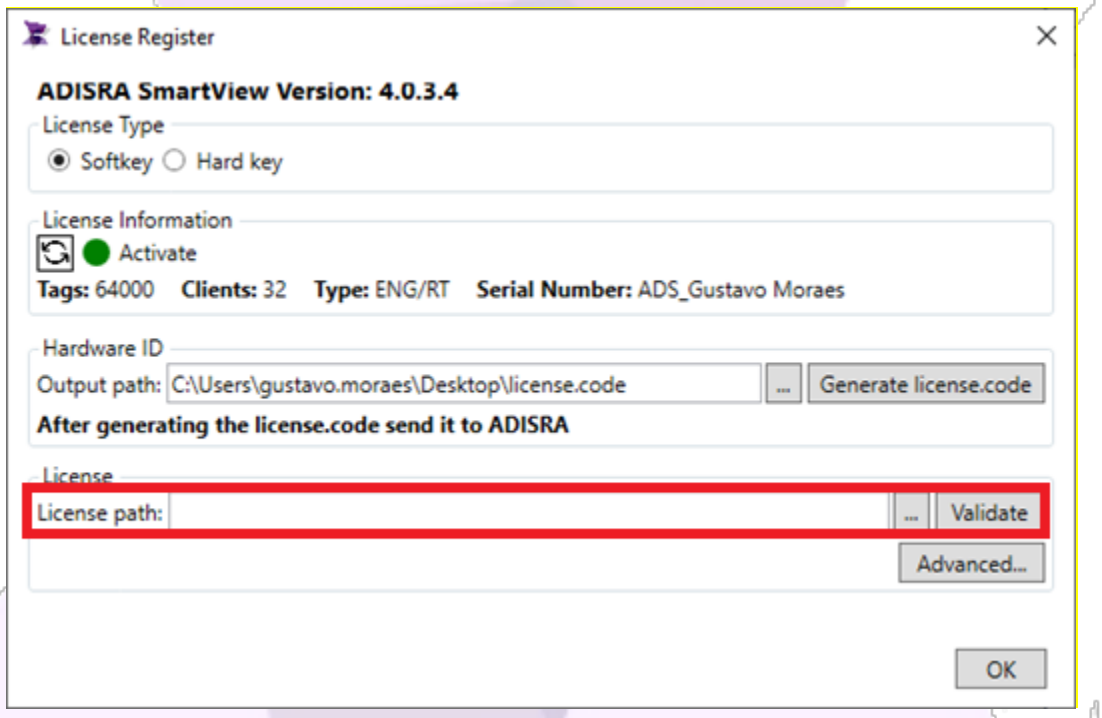


Note: *The user can change the deactivated license file location by clicking on the ... button next to the "Deactivate License" button.*

3. Click the button "Deactivate License" to generate the file.

4. Once deactivated, ADISRA SmartView will ask for the license again and will not work until a regenerated license is validated again.

To validate a regenerated license, send the deactivated license file to info@adisra.com and ADISRA will send a regenerated license back which you can save to the file location in License Path text box or browse to locate it by clicking on the ... button next to the License path and then click the Validate button.



11.4. Certificate of authenticity

When the user purchases a license for ADISRA SmartView, the software vendor/ADISRA will generate a certificate of authenticity for that license, this certificate contains the owner's name, or the company name, the email, along with the license's unique serial number and the hardware ID.

See below an example of a certificate of authenticity:

12. Application Deployment and Execution

In this topic we will explain how to execute and interact with Linux Runtime.

12.1. Transfer Application

12.1.1. Uploading ADISRA SmartView projects Manually

Applications can be manually transferred by directly moving the application folder from one location to another, eliminating the need for additional configuration.

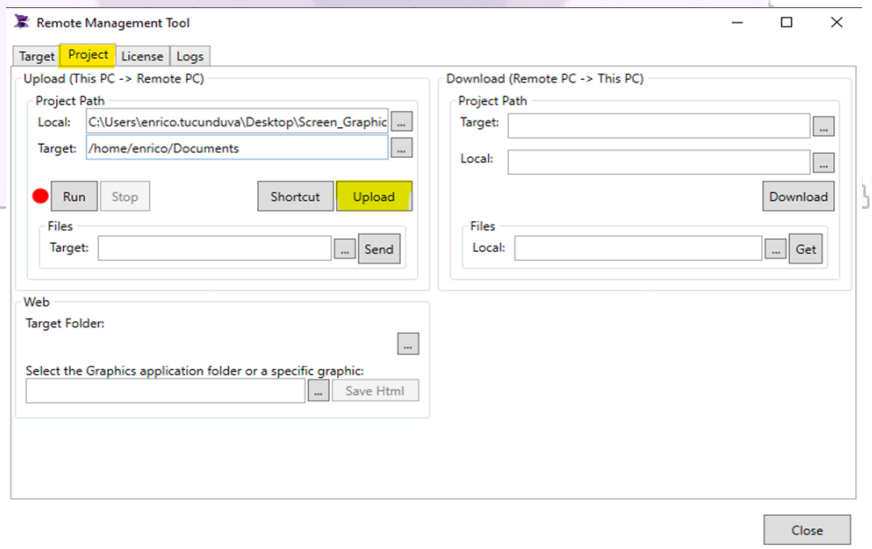
12.1.2. Uploading ADISRA SmartView projects to Linux using the Remote Management Tool

This is the best tool to transfer and manage the applications on a Linux environment.

To upload a project to Linux, you should connect using Remote Management Tool to the Linux server (the RemoteToolsServiceLnx must be running).

1. In the Remote Management Tool window, the 'Local' field should be filled in with the opened project. By default, it will have the current application folder loaded in the Engineering environment, so you probably won't need to change it.

2. In the 'Target' field go to the […] button and choose the path to where the project should be uploaded in the Linux server, then click the 'Upload' button for the action to be completed:



3. The project is uploaded to the Linux server and can be executed by clicking on the “Run” button.

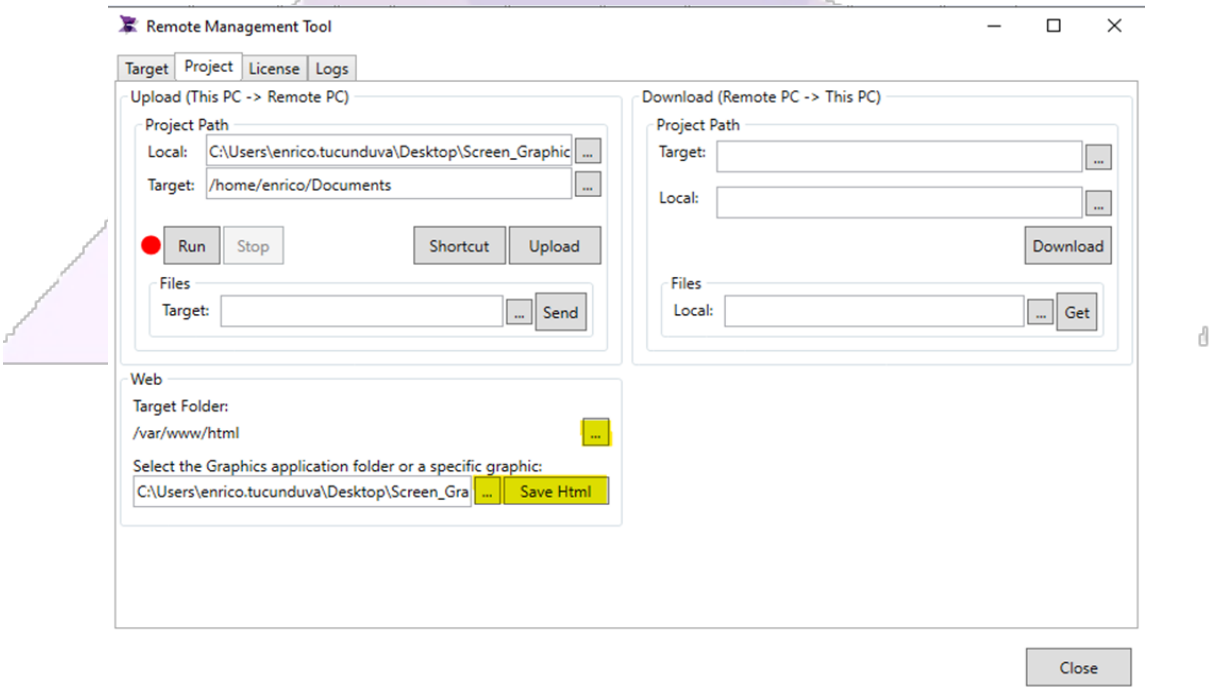
12.1.3. Uploading web pages to Linux (for Web Viewer access) using the Remote Management Tool

In the project tab the user also has the option to upload files manually or upload the html pages.

1. The image below highlights the target folder, the source, and the Save Html button. The source can be selected as the entire Graphics folder or a single graphic.

/var/www/html

2. Next, click the […] button next to the ‘Select the Graphics application folder or a specific graphic field to choose the desired graphic to be uploaded.



3. If everything is configured correctly, click the 'Save Html' button to upload the graphics.

12.2. Execute the Runtime Process

There are different ways to execute the Linux Runtime, in this section some of them will be explained.

12.2.1. Linux Runtime execution (command line)

To manually execute the Linux Runtime, copy the ADISRA Smart View project to the Linux workstation and type the following command (It will be necessary to type the root password):

Note: The *ProjectPath* argument must provide the path to the ADISRA SmartView project file;

```
cd /home/svlnxadm/ADISRA-RunTimeLnx-Release

sudo ./RunTimeLnx ProjectPath="/home/svlnxadm/projects/SVProjects/complete/complete.prj"
```

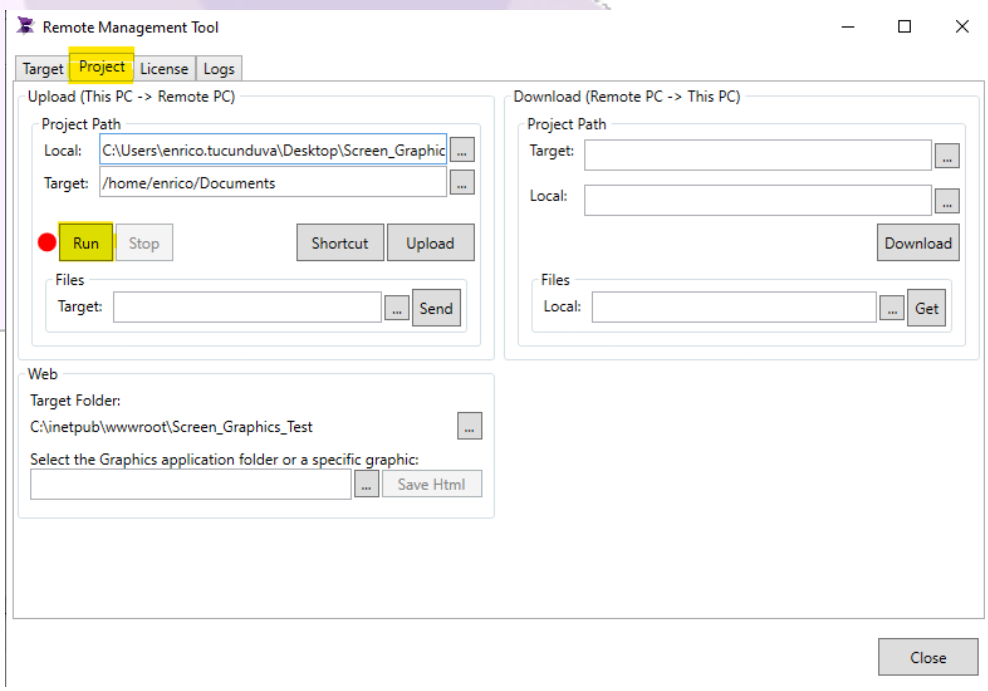
```
svlnxadm@svlnxvm:~/ADISRA-RunTimeLnx-Release$ sudo ./RunTimeLnx ProjectPath="/home/svlnxadm/projects/SVProjects/complete/complete.prj"
[sudo] password for svlnxadm:
Project path successfully read:[/home/svlnxadm/projects/SVProjects/complete/complete.prj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:9003
```

. . .

```
The engine Trigger has been started
The engine Alarms has been started
The engine Recipe has been started
The engine RuleBasedSystem has been started
All enginess have been started
info: RunTimeLnx.Worker[0]
      Worker running at: 03/31/2024 12:21:08 -03:00
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/svlnxadm/ADISRA-RunTimeLnx-Release
info: RunTimeLnx.Worker[0]
      Worker running at: 03/31/2024 12:21:18 -03:00
```

12.2.2. Linux Runtime execution (Remote Management Tool)

In the project tab the user also has the option to upload files to the Linux server. After the RMT connection is made, and the project uploaded, simply click the 'Run' button:



12.2.3. Linux Runtime execution (As a service)

To execute the ADISRA SmartView Runtime as a service so that it becomes available immediately after the boot (before login), the Linux Systemd service manager can be used. The steps below briefly describe how it can be configured.

IMPORTANT: *Not all Linux distributions natively support Systemd.*

12.2.3.1. Configure the appsettings.json

Edit the appsettings.json file, located in the ADISRA SmartView Linux Runtime installation, and set the ADISRA SmartView project that will be loaded into the Runtime:

/home/svlnxadm/ADISRA-RunTimeLnx-Release/appsettings.json

```
{
  "ProjectPath": "/home/svlnxadm/ADISRA-projects/complete/complete.prj",
  "grpcPort": "9000",
  "DebugPause": "FALSE",
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  }
}
```

Note: The paths illustrated will vary depending on the installation folder and project path.

12.2.3.2. Create the Systemd service file

The Systemd service file must be created as follows in the path "/etc/systemd/system". The following command will create the file and open the editor:

```
sudo vi /etc/systemd/system/runtimelnx.service
```

Press “i” to enter insert mode, paste the following content and save with “:x”:

```
[Unit]
Description=RunTime Linux Service Systemd

[Service]
Type=notify
ExecStart=/home/svlnxadm/ADISRA-RunTimeLnx-Release/RunTimeLnx
TimeoutStartSec=300
WorkingDirectory=/home/svlnxadm/ADISRA-RunTimeLnx-Release/

[Install]
WantedBy=multi-user.target
```

12.2.3.3. Configure Systemd

1. Reload the Systemd daemon so that the runtimelnx.service file gets loaded.

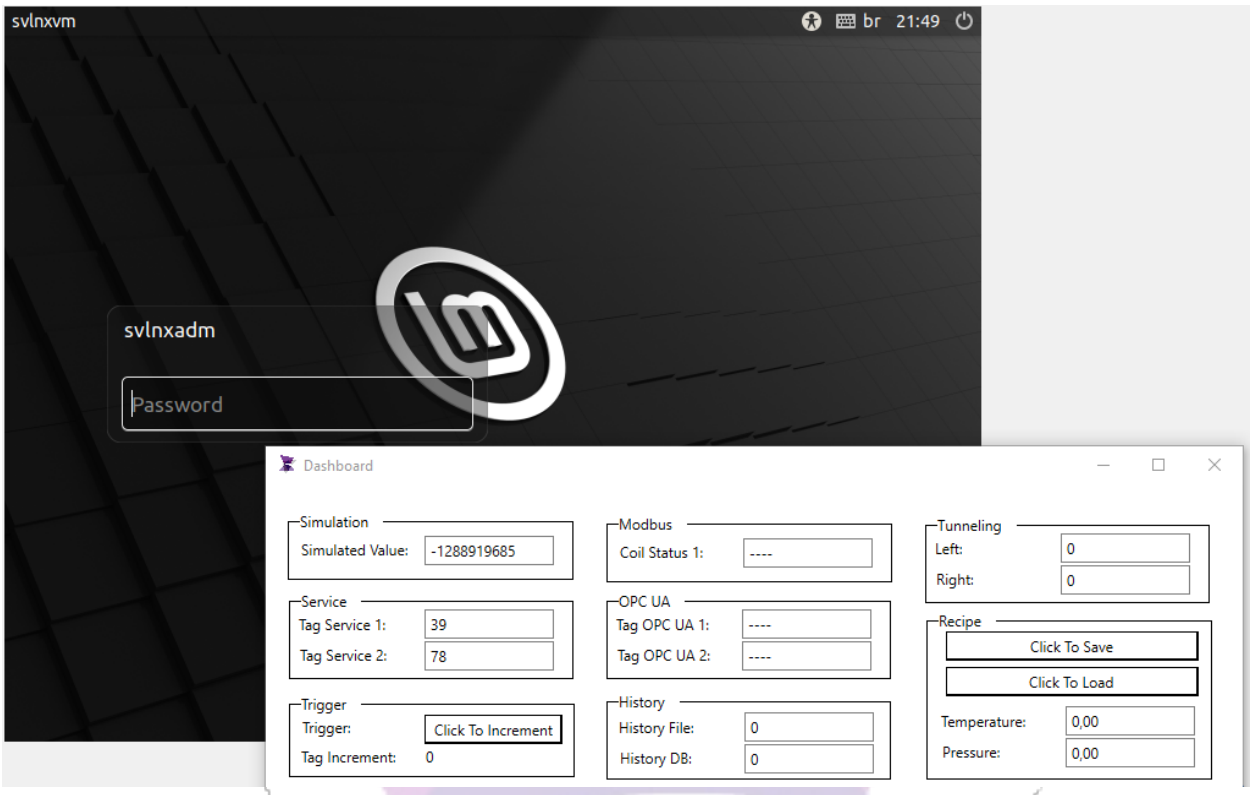
```
sudo systemctl daemon-reload
```

2. Configure it to be executed during the boot (before login)

```
sudo systemctl enable runtimelnx.service
```

12.2.3.4. Validate the Systemd Configuration

1. Restart the Linux Workstation.
2. When the machine gets ready (It’s not necessary to log in), open the Viewer in the Windows client workstation.
3. Make sure that the Linux Runtime is running and providing data:



12.2.3.5. Systemd useful commands

Command	Description
sudo systemctl status runtimelnx.service	Displays the service status (enabled, started, stopped, etc)
sudo systemctl start runtimelnx.service	Manually starts the service
sudo systemctl stop runtimelnx.service	Manually stops the service
sudo systemctl enable runtimelnx.service	Enables the service to be executed during the system startup (boot)
sudo systemctl daemon-reload	Must be executed whenever a change to the “runtimelnx.service” file is made.
sudo journalctl -u runtimelnx.service	Visualize the outputs generated from the ADISRA SmartView Linux Runtime while the service is running.

12.3. Ports

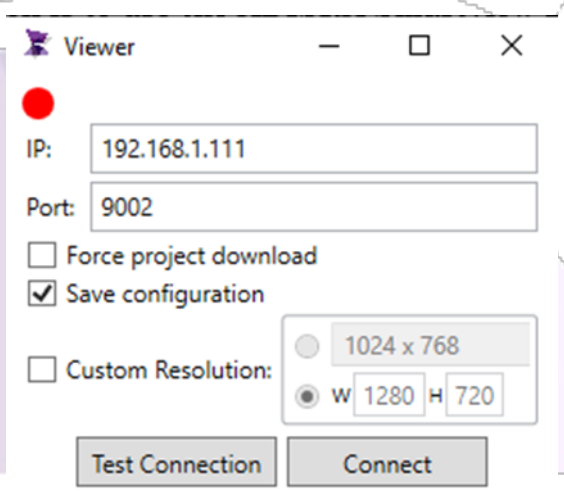
The application uses specific ports for some functionalities. Users can change these ports in project, the default configurations are as follows:

Viewer:
Default Port: 9002
Web Viewer:
Default Port: 9003
Remote Management Tools:
Default Port: 9004

12.4. Viewer Connection

This section shows the ADISRA SmartView Viewer being used for remote access.

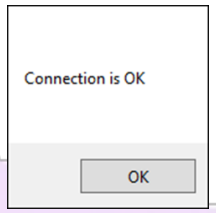
To access the Runtime remotely, open the Viewer application of the ADISRA SmartView using the Launcher or, go directly to the installation folder (by default, C:\Program Files (x86)\ADISRA\Adisra SmartView\Bin\Viewer.exe). The following window will appear:



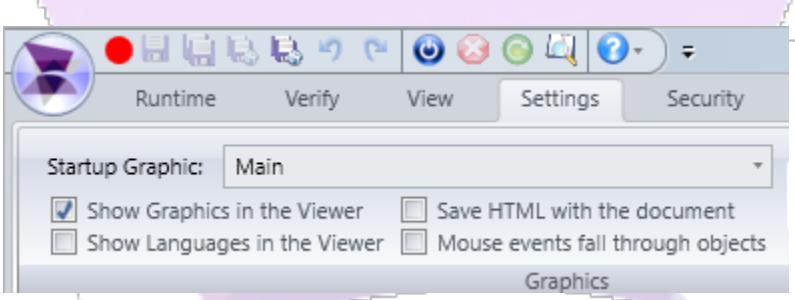
Go to the Linux server and in the Command Prompt use the following command to get its IP address:

```
hostname -I
# 192.168.2.215 2804:3404:101:8f00:3a84:af16:998e:45c7
2804:3404:101:8f00:ecc3:52bc:502c:d949
```

Insert the IP address in the 'IP' field and click the test connection button, if the Runtime is active and the connection is successful the following message should appear

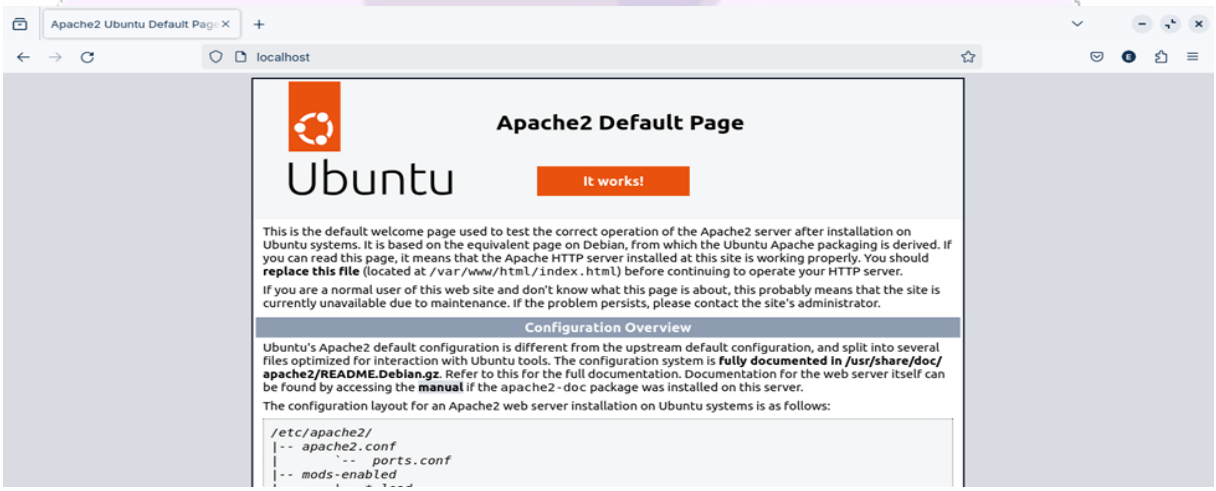


You should now be able to click the connect button and the project graphics will be loaded. Please make sure that the startup graphic is selected or else the screens won't be displayed:



12.5. Web Connection

After saving the graphics as Html and transferring the web pages to the target Linux machine, it is important that you have configured a Web Server. In this example we use Apache as the serves for the web pages and it uses the path "/var/www/html". By default, Apache's installation already places an index.html on that folder, so we can test it "localhost/index.html" using a local web browser for desktop distributions.



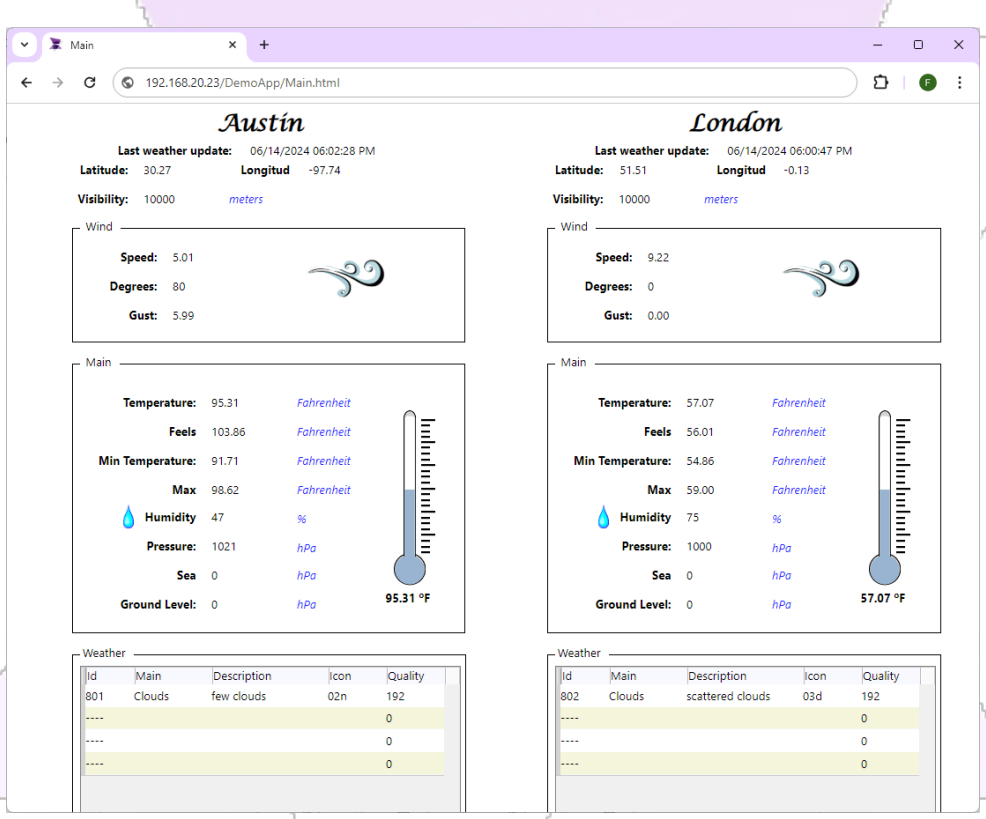
To open the web pages from your application, first make sure you transferred them to "/var/www/html". Let's say "/var/www/html/DemoApp".

We can access the web application using the following url:

"*http://localhost/DemoApp/Main.html*"

or remotely using the IP address

"*http://192.168.20.23/DemoApp/Main.html*".



12.6. Data Watcher Connection

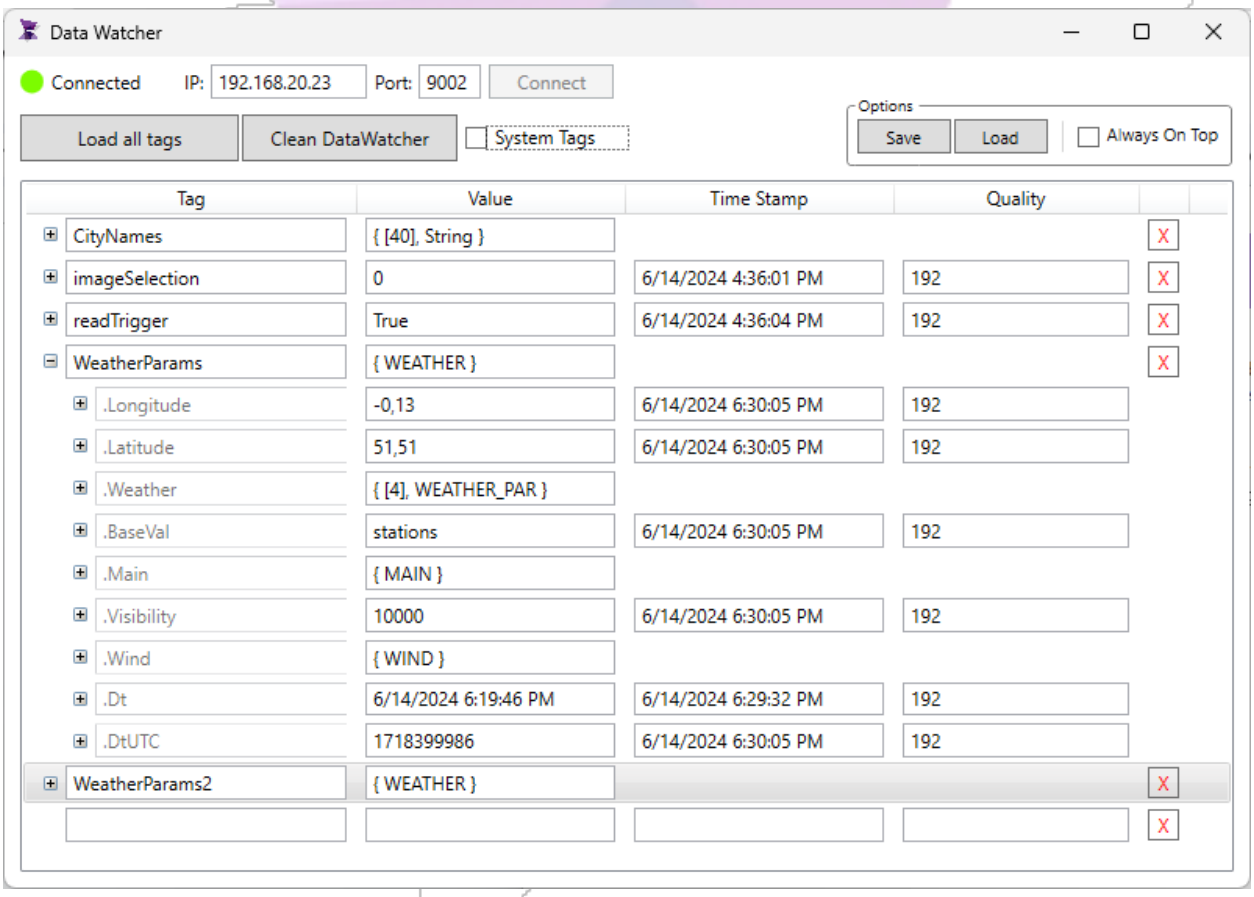
To remotely connect the Data Watcher to the ADISRA SmartView Runtime Linux please follow the steps below:

1. To get the Linux IP Address: Go to the Linux server and open a Command Prompt. Use the following command to get its IP address.

```
hostname -I
# 192.168.2.215 2804:3404:101:8f00:3a84:af16:998e:45c7
2804:3404:101:8f00:ecc3:52bc:502c:d949
```

2. In the Windows computer with the data watcher installed, please open the Data Watcher using the Launcher or directly from the installation folder.

3. As soon as the Data Watcher opens, you can set the IP address and the port number (9002 by default) and connect to the Linux Runtime.



Note: *The IP Address retrieved by the Linux command should be inserted in the windows command.*

4. Now you can start monitoring the application tags remotely.

13. Linux Runtime troubleshooting

This topic provides concise guidance for diagnosing and resolving issues encountered while running Runtime applications on Linux systems.

13.1. How to check if the Runtime executables have permission to be executed

To check if the Runtime executables have permission to be executed, open a Command Prompt and use the following command:

```
ls -l ./AdisraRunTime/RemoteToolsService ./AdisraRunTime/RunTimeLnx
# -rwxrwxr-x 1 svlnxadm svlnxadm 142840 Dec 27 10:28
./AdisraRunTime/RemoteToolsService
# -rwxrwxr-x 1 svlnxadm svlnxadm 142840 Dec 27 10:23
./AdisraRunTime/RunTimeLnx
```

13.2. Permission denied in Web Viewer

When accessing a project via web viewer it is possible that the folder permissions are blocked. To resolve the problem, open a Command Prompt and use the following command:

```
sudo chown -R svlnxadm:svlnxadm /var/www/html/<project_name>
sudo chmod -R 755 /var/www/html/<project_name>
```

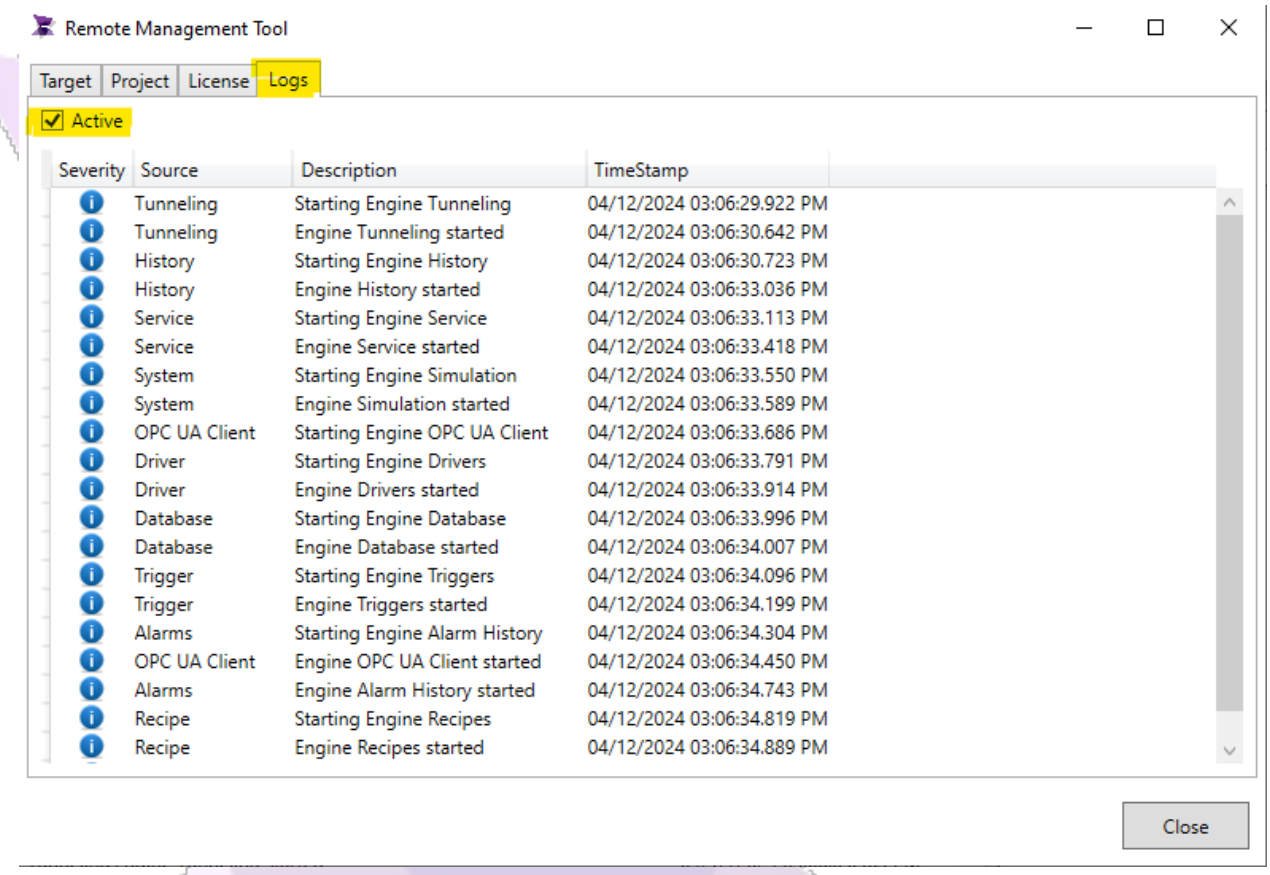
Note: *The user 'svlnxadm' should be adjusted according to the use case scenario, and the same should be done with '<project_name>' using the actual project name.*

13.3. Runtime Logs on Linux

If analyzing logs is required, there are different ways to access them (via Remote Management Tool or by configuring logs to be saved locally on Linux for future reference).

13.3.1. Analyzing Logs using the Remote Management Tool

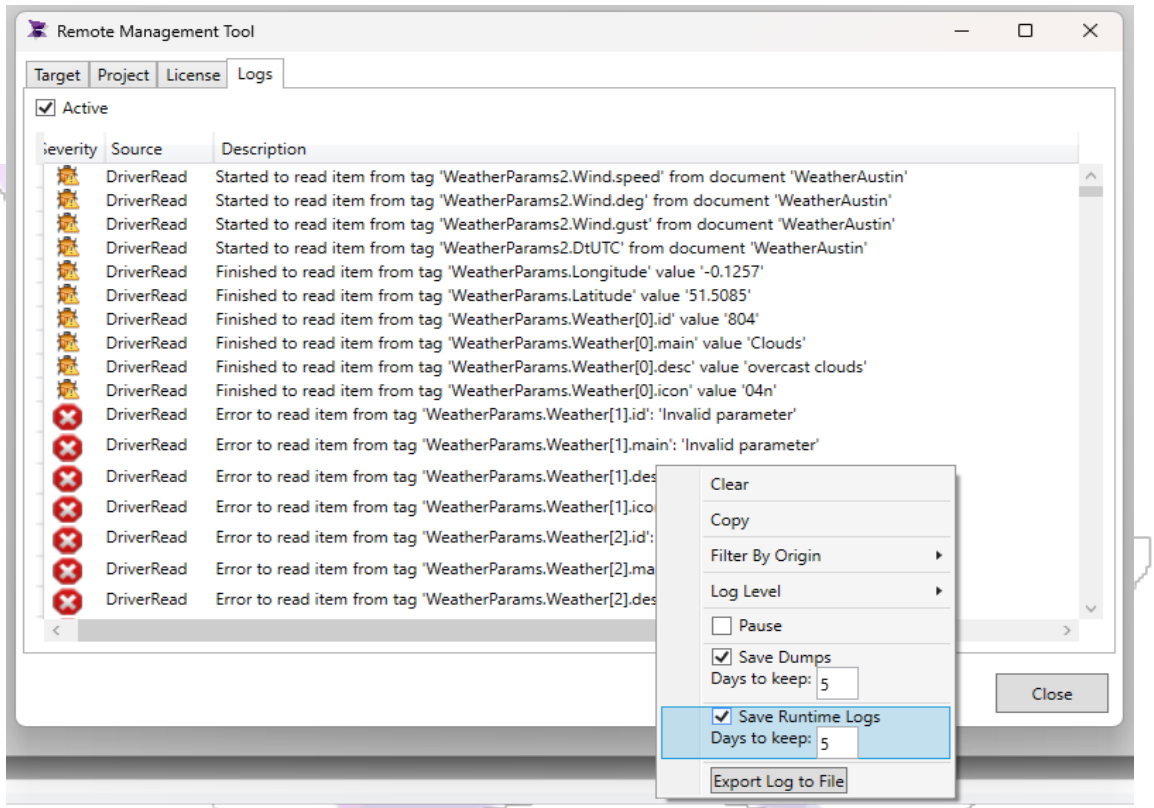
In the Remote Management Tool, go to the 'Logs' tab and select the 'Active' checkbox:



Every time you change the filters; you will need to stop and re-upload the application to the target system and restart it.

13.3.2. Analyzing Logs saved locally on the Linux machine.

You can set the runtime log to be saved locally on the Linux application folder by editing the application to save runtime logs (disabled by default). As soon as you enable it, please stop the application, re-upload it to the Linux machine and start the application. It will save the runtime log to the logs folder inside the application folder.


Remote Management Tool
Target
Project
License
Logs
Active
Severity
Source
Description
DriverRead Started to read item from tag 'WeatherParams2.Wind.speed' from document 'WeatherAustin'
DriverRead Started to read item from tag 'WeatherParams2.Wind.deg' from document 'WeatherAustin'
DriverRead Started to read item from tag 'WeatherParams2.Wind.gust' from document 'WeatherAustin'
DriverRead Started to read item from tag 'WeatherParams2.DtUTC' from document 'WeatherAustin'
DriverRead Finished to read item from tag 'WeatherParams.Longitude' value '-0.1257'
DriverRead Finished to read item from tag 'WeatherParams.Latitude' value '51.5085'
DriverRead Finished to read item from tag 'WeatherParams.Weather[0].id' value '804'
DriverRead Finished to read item from tag 'WeatherParams.Weather[0].main' value 'Clouds'
DriverRead Finished to read item from tag 'WeatherParams.Weather[0].desc' value 'overcast clouds'
DriverRead Finished to read item from tag 'WeatherParams.Weather[0].icon' value '04n'
DriverRead Error to read item from tag 'WeatherParams.Weather[1].id': 'Invalid parameter'
DriverRead Error to read item from tag 'WeatherParams.Weather[1].main': 'Invalid parameter'
Clear
Copy
Filter By Origin
Log Level
Pause
Save Dumps
Days to keep: 5
Save Runtime Logs
Days to keep: 5
Export Log to File
Close