Quick Start Manual

OptixPanel Printing

OPTIXPANEL-PRINTING-QS001-EN-AUG-2025

Sharadsoft

Contents

End-User License Agreement (EULA)

End-User License Agreement (EULA) for Software developed by Sharadsoft

Last Updated: 01-Mar-2025

1. Introduction

This End-User License Agreement (EULA) is a legal agreement between you (either an individual or a single entity) and Sharadsoft for the Software developed by Sharadsoft, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation ("Software").

2. License Grant

Sharadsoft grants you a revocable, non-exclusive, non-transferable, limited license to download, install, and use the Software solely for your personal, non-commercial, commercial purposes strictly in accordance with the terms of this Agreement. Any software offered for free by Sharadsoft must be provided free of charge to sub-customers, and license/activation does not apply to such free Software.

3. Restrictions

You agree not to, and you will not permit others to:

- Decompile, reverse engineer, disassemble, attempt to derive the source code of, or decrypt the Software.
- Make any modification, adaptation, improvement, enhancement, or translation work from the Software.
- Violate any applicable laws, rules, or regulations in connection with your access or use of the Software.
- Remove, alter, or obscure any proprietary notice (including any notice of copyright or trademark) of Sharadsoft or its affiliates, partners, suppliers, or the licensors of the Software.

4. Termination

This Agreement is effective from the date you first use the Software and shall continue until terminated. You may terminate this Agreement at any time by uninstalling and deleting the Software and all copies thereof. This Agreement will terminate immediately if you fail to comply with any of its Terms.

5. Intellectual Property

All titles, including but not limited to copyrights, in and to the Software and any copies thereof are owned by Sharadsoft or its suppliers. All rights not expressly granted are reserved by Sharadsoft.

6. Disclaimer of Warranties

The Software is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

7. Limitation of Liability

In no event shall Sharadsoft be liable for any special, incidental, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits, loss of data, or other information, for business interruption, for personal injury, for loss of privacy arising out of or in any way related to the use of or inability to use the Software, third-party Software and/or third-party hardware used with the Software, or otherwise in connection with any provision of this Agreement), even if Sharadsoft has been advised of the possibility of such damages and even if the remedy fails of its essential purpose.

8. Governing Law

This Agreement shall be governed by and construed in accordance with the laws of Mumbai-Maharashtra-India, without regard to its conflict of law principles.

9. Entire Agreement

This Agreement constitutes the entire Agreement between you and Sharadsoft with respect to the use of the Software and supersedes all prior or contemporaneous understandings regarding such subject matter.

Third-Party Trademarks Disclaimer

Disclaimer:

All product names, logos, and brands are the property of their respective owners. All company, product, and service names used in this manual are for identification purposes only. Use of these names, logos, and brands does not imply endorsement.

Sharadsoft acknowledges all third-party trademarks and logos referenced in this manual, including but not limited to:

- FactoryTalk Optix Studio, OptixPanel are the trademarks of Rockwell Automation.

Any third-party trademarks or service marks referenced in this manual are the property of their respective owners. The inclusion of any third-party trademarks does not imply a partnership or endorsement by Sharadsoft.

For any trademark-related concerns or inquiries, please contact us at info@sharadsoft.com

Introduction

We are happy to deploy the libraries for Optix Studio, developed with a focus on fulfilling your specific printing requirements. While these libraries offer the essential functionality you need, please note that they are designed to provide a simple and user-friendly experience to achieve the desired outcomes of printing through OptixPanel.

Key Highlights:

1. Simplicity and Efficiency: Our libraries are built to offer a straightforward and efficient way to get the prints without the complexity of the OptixPanel NetLogic development.
2. Tailored to Your Needs: We have prioritized the features that are most relevant to your printing requirements, ensuring that the tool meets your specific needs without overwhelming you with unnecessary options.
3. Ease of Use: Allows for quick and easy integration, making it accessible even for users with minimal technical expertise.
4. Focused Functionality: While it may not have all the advanced features of professional parameter configuration and feedback mechanisms, our tool is enough to handle your printing tasks effectively.

We understand that you may compare our libraries with more comprehensive professional alternatives. It's important to recognize that our libraries are designed to provide the essential functionality required for your printing needs, focusing on simplicity and ease of use. This approach allows us to deliver a solution that is both cost-effective and user-friendly.

Important User Information

- The points and notes mentioned in this manual are based on our sole understanding and the data available to us as of Aug 1, 2025. Please be aware that interpretations or meanings may vary. Kindly refer to the respective third-party manuals or documentation for further clarification or detailed information.
- The network printer must be compatible with direct PDF printing.
- Keep PDF size minimal (safe limits vary by printer). 10-20MB is a safer range until verified.
- Sometimes, the Network printer experiences a service error and automatically restarts after a print command. An immediate solution available for now is reinitiating the print command.
- Please do not connect the LAN printer to the OptixPanel WAN port; instead, connect it to the OptixPanel LAN port directly (for demo application testing only) or through an Ethernet switch (as shown in the system architecture included in this manual).
- Use of a serial port for serial printing will consume one token.
- Images used in the HMI demo application are for demo purposes only and licensed by Sharadsoft for their use in an application developed by Sharadsoft. To maintain optimal CPU performance, avoid using multiple images within the application. Excessive image usage can lead to increased processing demands, potentially impacting overall application efficiency.
- Please keep in mind the CPU performance, Resource constraints, Real-time processing, and Power consumption of the HMI while designing & deploying the libraries on the HMI.

Prerequisites

System used while testing - Hardware & Software

Description	Details
OptixPanel Standard	Firmware 6.0.1.165
Ethernet Switch	Unmanaged, 5 Port
HMI App Development Software	FT Optix Studio v1.6.2
Windows Operating System	Windows 11 Pro
Visual Studio	2022
Workstation RAM	8 GB or more
Sharadsoft Software	1.00.00
Network Printer	With Direct PDF Print capability

Note:

1. We have tested the FilePrintLibrary on HP make LaserJet Pro M403n
2. We have tested the SerialPrintLibrary on an EPSON Dot Matrix Printer
3. Check for 'PDF Printing' in the HP printer specification before purchasing the printer.

Sharadsoft software details

Description	Activation	Functionality
FilePrintLibrary	Requires *1, 3	Print PDF File from Optix Graphic Terminal on the Network Printer connected to the LAN port.
SerialPrintLibrary	Requires *1, 4	Print text from the Optix Graphic Terminal on the Serial Printer connected to the Serial port.

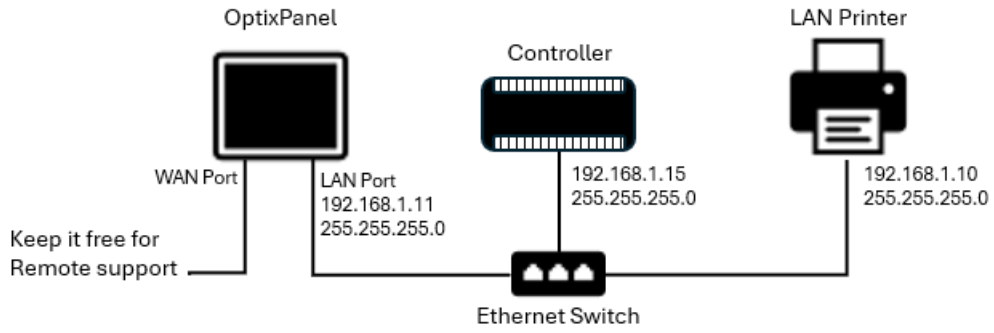
Note:

1. FilePrintLibrary is a separate purchase and requires an individual activation key.
2. SerialPrintLibrary is a separate purchase and requires an individual activation key.
3. FilePrintLibrary: Without activation, it will print only the sample PDF included with the demo application
4. SerialPrintLibrary: Without activation, it will add 'Demo Mode' as a suffix to each line

Section1: File Printing from OptixPanel

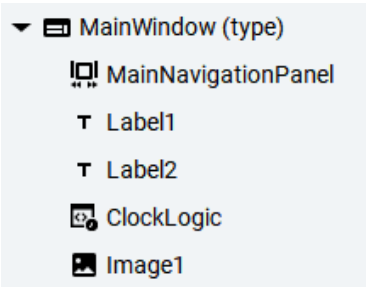
Your first step will be creating the HMI application using HMI application development software. We assumed you had already developed reporting functionality and saved the report PDF to HMI memory.

System architecture-File Printing

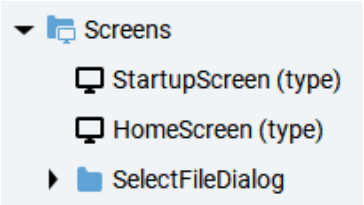


Configuration-MainWindow, Screens, Tags, RuntimeNetLogic

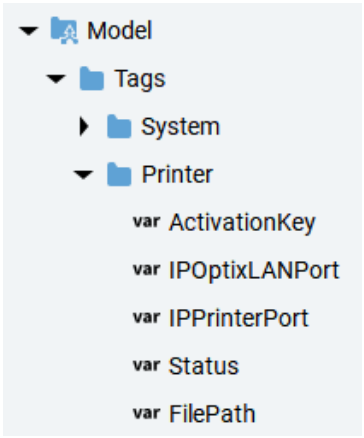
1. Configure the 'MainWindow' section with your required screen navigation options. For the demo application, we have used a navigation panel as follows:

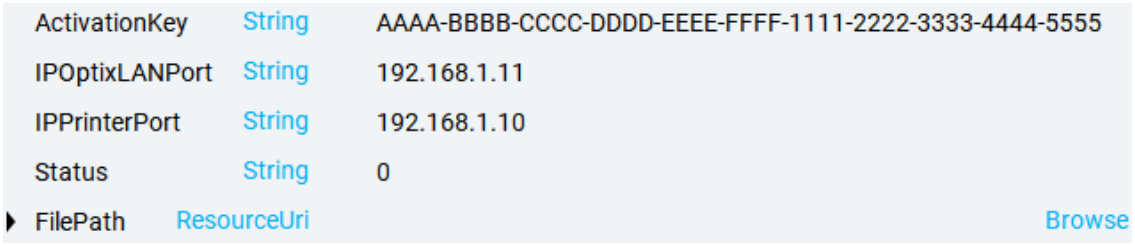


2. Create and configure the required screens for your application. For the demo application, we have created and configured 'StartupScreen' and 'HomeScreen' named screens and added them to the main window navigation panel.



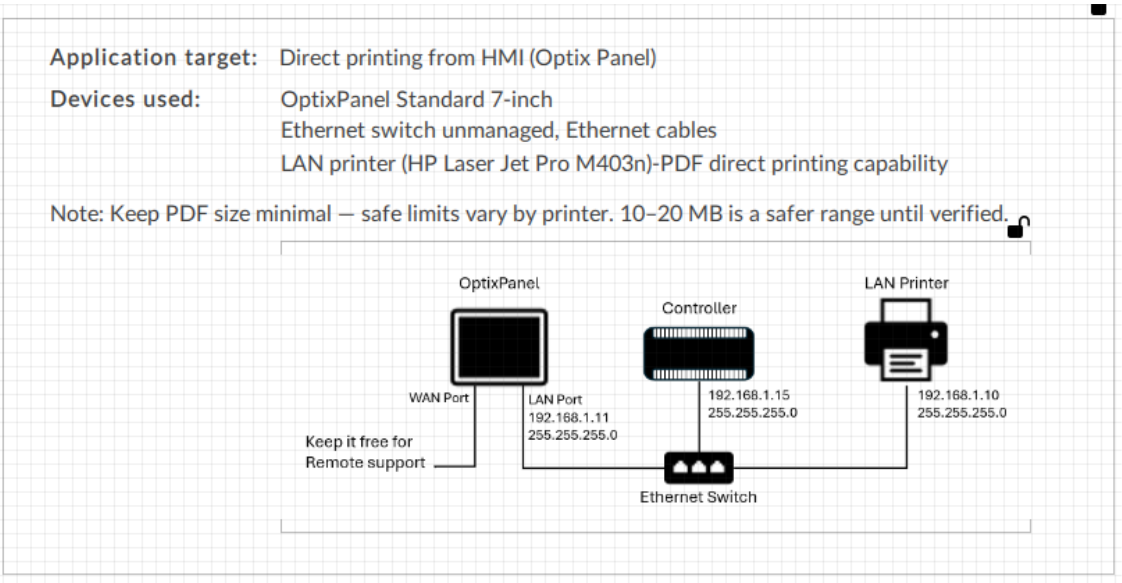
3. Create the tags under the ‘Model’ section. For the demo application, we have created the tags as follows:



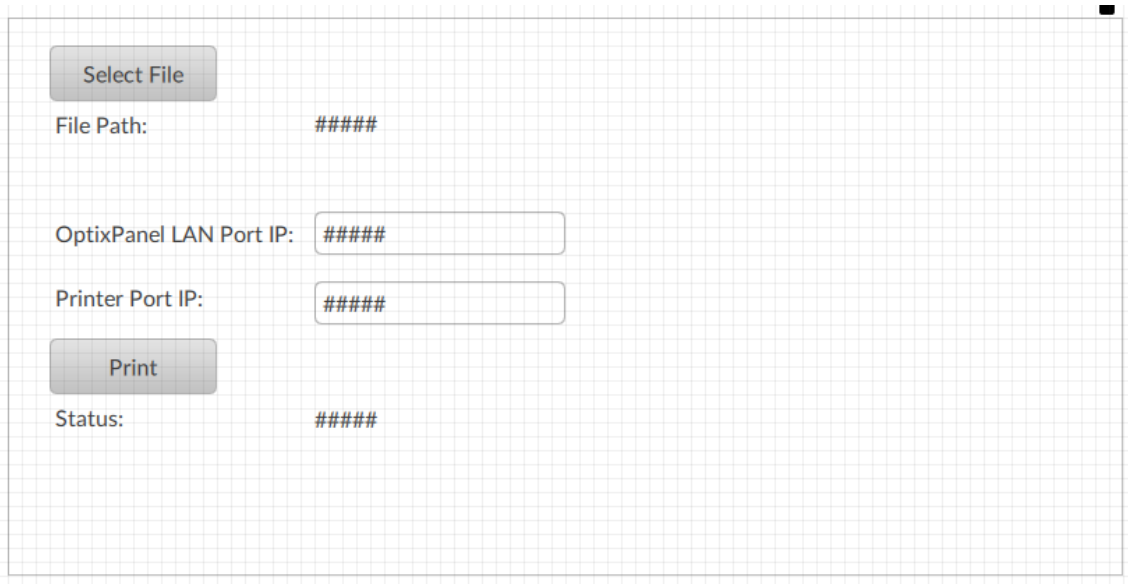


You can set ‘ActivationKey’, ‘IPOptixLANPort’ & ‘IPPrinterPort’ here only. ‘Status’ tag is the read-only tag that will give you the print command execution status. ‘FilePath’ is the tag where you need to pass the PDF file path during runtime.

4. The startup screen developed for the demo application is as follows:



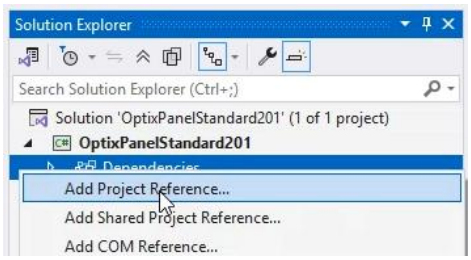
5. The home screen developed for the demo application is as follows:



'Select File' button will offer you a file selection dialog box, from where you can select the PDF file, and 'Print' button will provide you with the option to send that PDF file to the network-connected LAN printer directly.

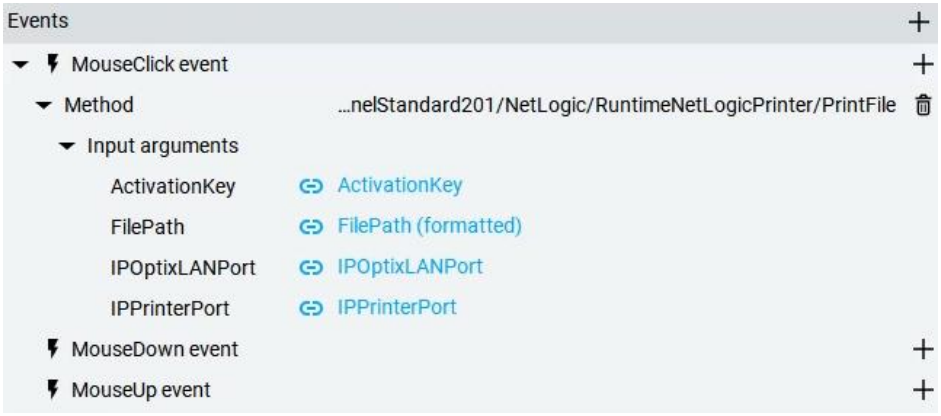
'Select File' button event configuration is as follows:

6. Add the ‘RuntimeNetLogicPrinter’ script under ‘NetLogic’ section. Open the created NetLogic using Visual Studio 2022 for editing purposes. Add the ‘FilePrintLibrary’ dll file as a reference in the dependencies section. Edit/copy the script as per the given demo application script.




```
#region Using directives
using System;
using UAManagedCore;
using OpcUa = UAManagedCore.OpcUa;
using FTOptix.UI;
using FTOptix.HMIProject;
using FTOptix.NativeUI;
using FTOptix.CoreBase;
using FTOptix.System;
using FTOptix.Retentivity;
using FTOptix.NetLogic;
using FTOptix.SerialPort;
using FTOptix.Core;
using FilePrintLibrary;
#endregion

public class RuntimeNetLogicPrinter : BaseNetLogic
{
    FilePrinter MyFilePrinter=new FilePrinter();

    public override void Start()
    {
        // Insert code to be executed when the user-defined logic is started
    }


    public override void Stop()
    {
        // Insert code to be executed when the user-defined logic is stopped
    }

    [ExportMethod]


    public void PrintFile(string ActivationKey, string FilePath, string IPOptixLANPort, string IPPrinterPort)
    {
        string Status = string.Empty;

        MyFilePrinter.PrintFile(ActivationKey, FilePath, IPOptixLANPort, IPPrinterPort, out Status);

        LogicObject.GetVariable("Status").Value = Status;
    }

}
```

7. ‘Print’ button event configuration is as follows:

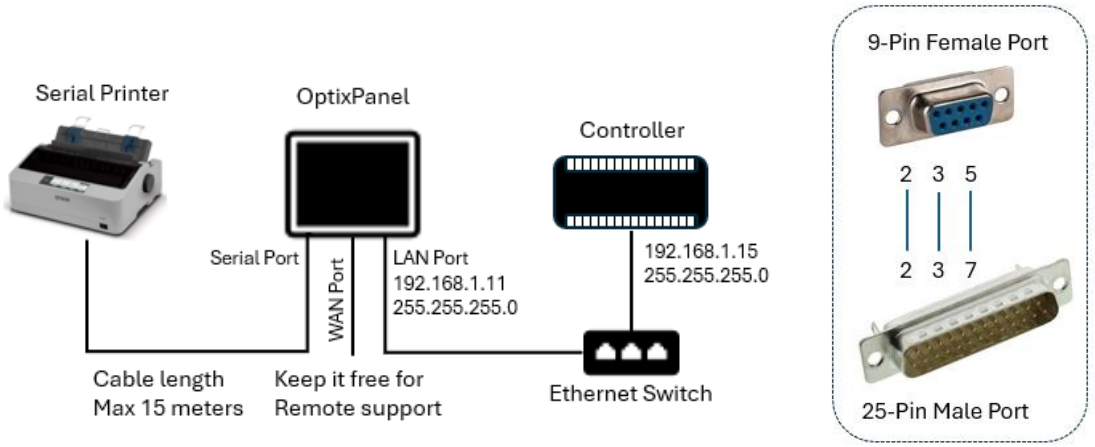


8. Now you are ready with the application for testing. Close Visual Studio if it is open. Save the application and download it to the OptixPanel graphic terminal for testing.

Section2: Serial Printing from OptixPanel

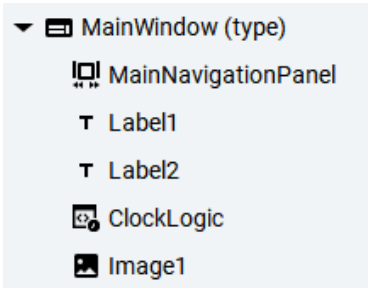
Your first step will be creating the HMI application using HMI application development software.

System architecture-Serial Printing

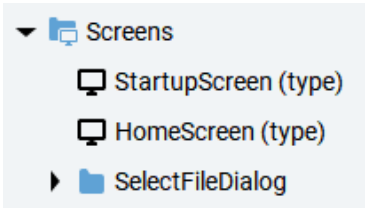


Configuration-MainWindow, Screens, Tags, RuntimeNetLogic

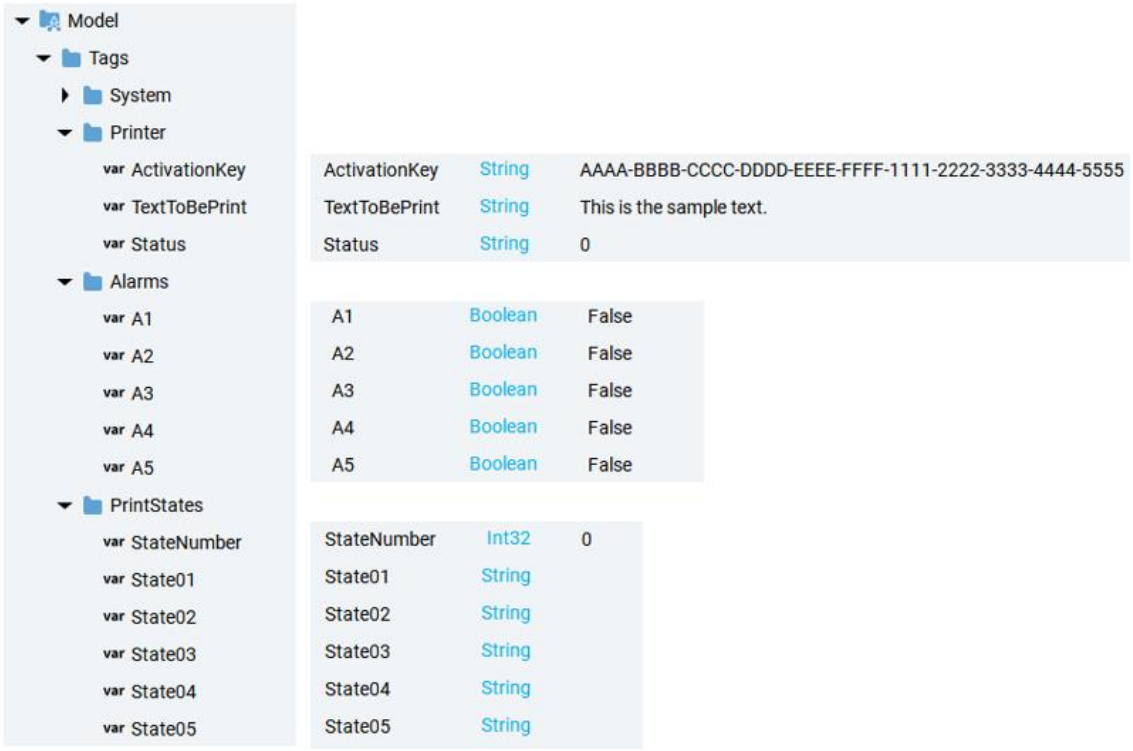
1. Configure the 'MainWindow' section with your required screen navigation options. For the demo application, we have used a navigation panel as follows:



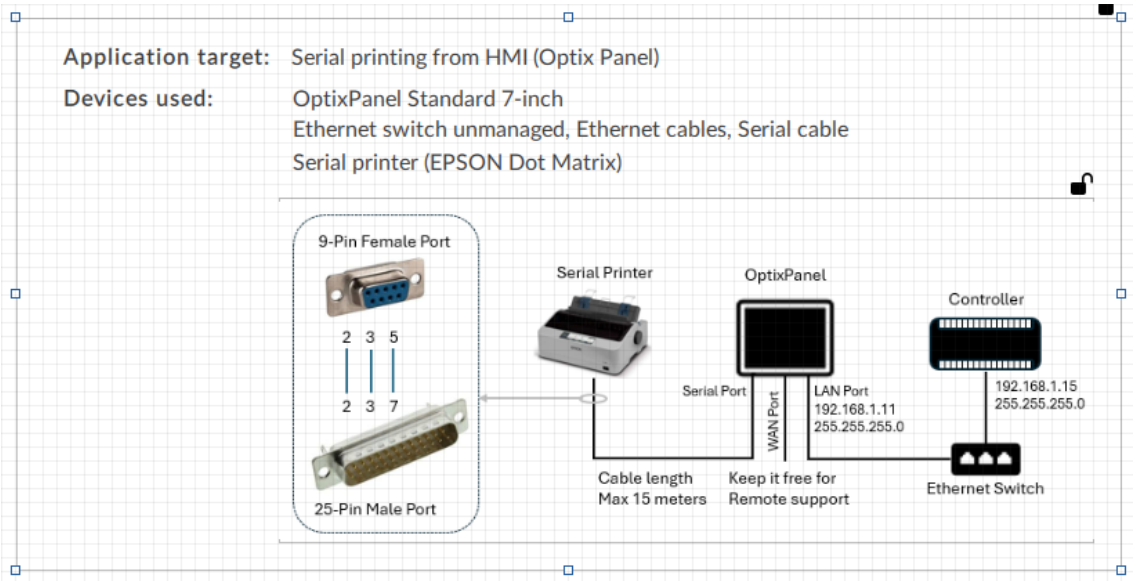
2. Create and configure the required screens for your application. For the demo application, we have created and configured 'StartupScreen' and 'HomeScreen' named screens and added them to the main window navigation panel.



3. Create the tags under the 'Model' section. For the demo application, we have created the tags as follows:

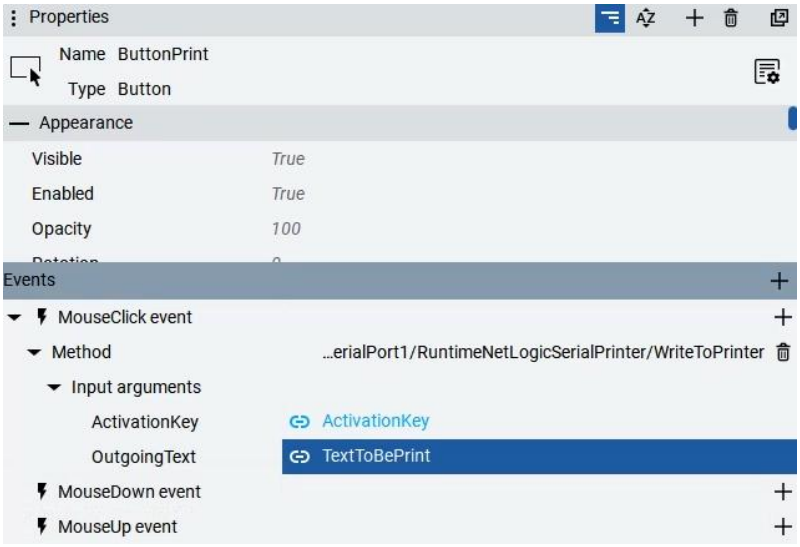


4. The startup screen developed for the demo application is as follows:

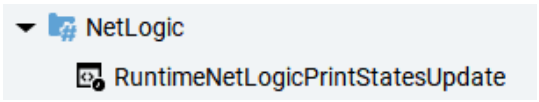


5. The home screen developed for the demo application is as follows:

'Print' button will send the text to the serial printer from the 'TextToBePrint' tag value. 'Print' button properties are set as follows:



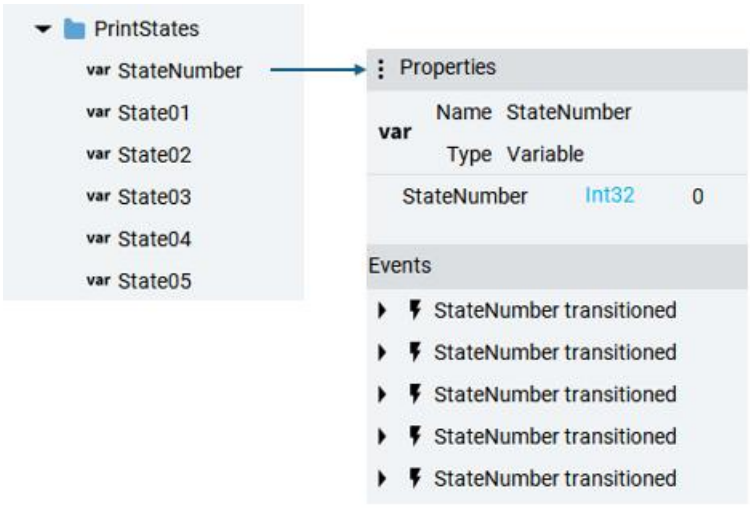
'StateNumber' tag value will send the text to the serial printer from the 'State01' to 'State05' tag value, depending on the value selection. You can design multiple states. These state tag values are updated periodically through NetLogic 'RuntimeNetLogicPrintStatesUpdate', which is scripted under 'NetLogic' section.



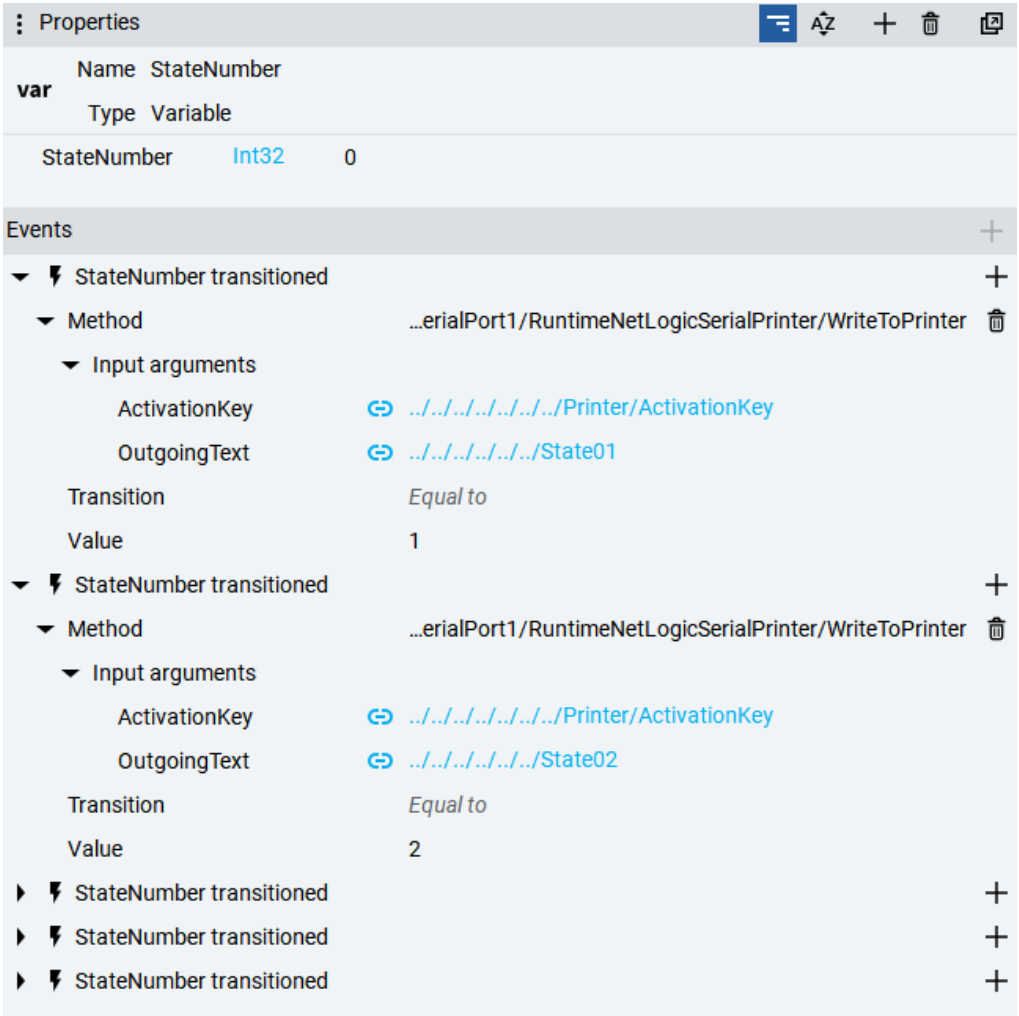

```
Using directives

public class RuntimeNetLogicPrintStatesUpdate : BaseNetLogic
{
    private PeriodicTask MyPeriodicTask;

    public override void Start()
    {
        // Insert code to be executed when the user-defined logic is started
        MyPeriodicTask = new PeriodicTask(UpdatePrintStates, 250, LogicObject);
        MyPeriodicTask.Start();

    }

    public override void Stop()
    {
        // Insert code to be executed when the user-defined logic is stopped
        MyPeriodicTask?.Dispose();

    }

    private void UpdatePrintStates()
    {
        Int32 Variable1, Variable2, Variable3;
        string StringDateTime, StringVariable1, StringVariable2, StringVariable3;

        Variable1 =Project.Current.GetVariable("Model/Tags/System/Hour").Value;
        Variable2 =Project.Current.GetVariable("Model/Tags/System/Minute").Value;
        Variable3 = Project.Current.GetVariable("Model/Tags/System/Second").Value;

        StringDateTime = DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss");
        StringVariable1 = Variable1.ToString("D2");
        StringVariable2 = Variable2.ToString("D2");
        StringVariable3 = Variable3.ToString("D2");

        Project.Current.GetVariable("Model/Tags/PrintStates/State01").Value = "-------------------------------------------------------------";
        Project.Current.GetVariable("Model/Tags/PrintStates/State02").Value = "Report Header";
        Project.Current.GetVariable("Model/Tags/PrintStates/State03").Value = "| Date & Time         |  Hour  | Minute | Second |";
        Project.Current.GetVariable("Model/Tags/PrintStates/State04").Value = "| " + StringDateTime + " |    " + StringVariable1 + "   |    " + StringVariable2 + "   |    " + StringVariable3 + "   |";
        Project.Current.GetVariable("Model/Tags/PrintStates/State05").Value = "Report Footer";
    }
}
```

Here, 'Hour', 'Minute' & 'Second' tag values are considered as process parameter values for demo purposes.

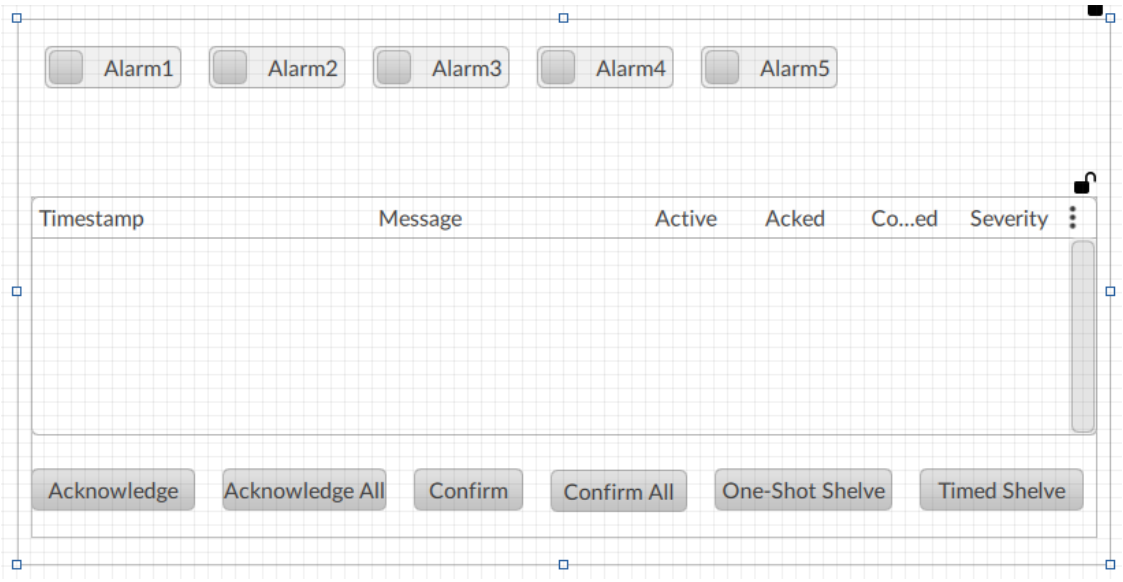
'StateNumber' tag properties are set as follows:



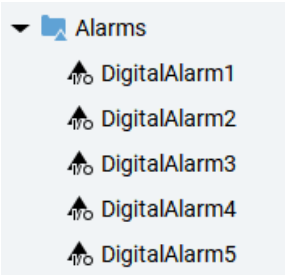
'StateNumber' transitioned event is configured as follows:



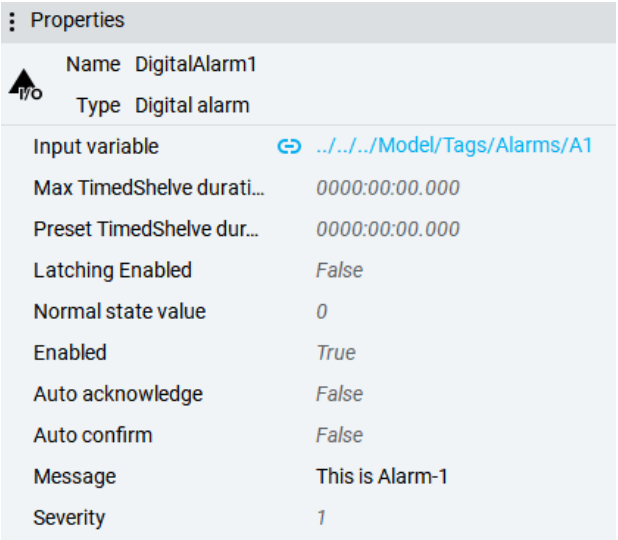
6. The alarm screen developed for the demo application is as follows:



Each alarm trigger will send the alarm message to the serial printer. Alarms are configured under the 'Alarm' section as follows:

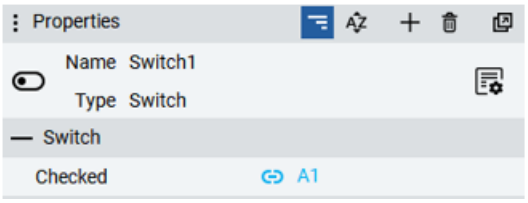


Set alarm properties as follows:

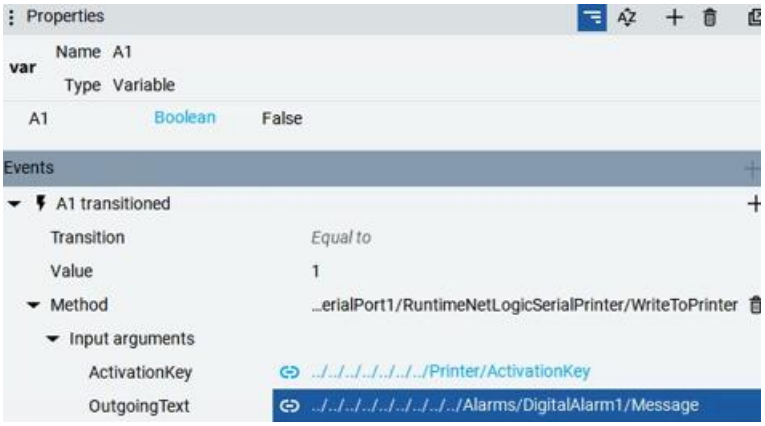


Tags from the 'Model/Tags/Alarms' section are bonded to the digital alarms as follows:

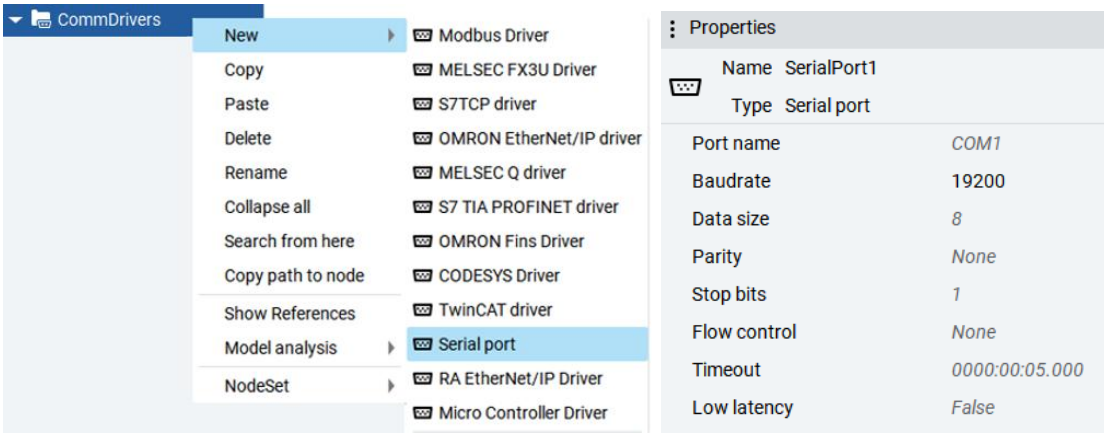
Tags are bonded to the switches on the 'HomeScreen'



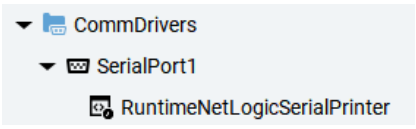
Tags properties are set as follows:

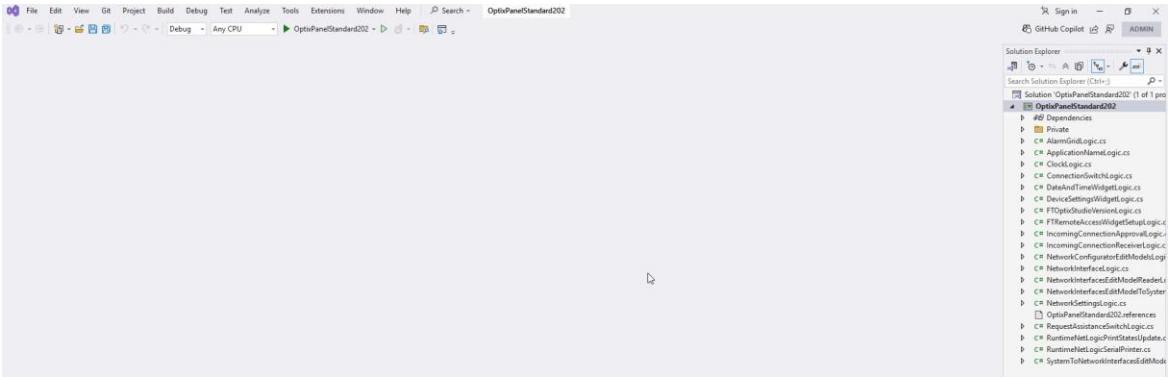


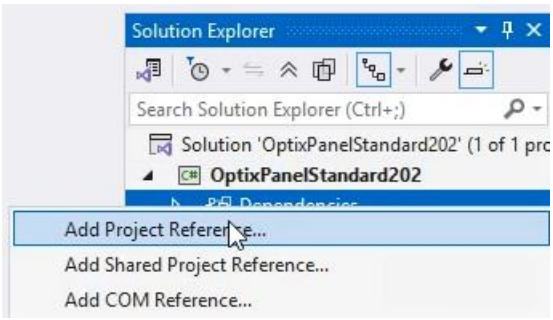
7. Add the serial port under 'CommDrivers' section and set its communication properties as follows:



8. Add the 'RuntimeNetLogicSerialPrinter' script under 'CommDrivers/SerialPort1' section. Open the created NetLogic using Visual Studio 2022 for editing purposes. Add the 'SerialPrintLibrary' dll file as a reference in the dependencies section. Edit/copy the script as per the given demo application script.







```
#region Using directives
using ...
#endregion

0 references
public class RuntimeNetLogicSerialPrinter : BaseNetLogic
{
    SerialPrinter MySerialPrinter = new SerialPrinter();
    private SerialPort OptixSerialPort;
    private SerialPrinter.ISerialPortWrapper SerialPortAdapter;
    0 references
    public override void Start()
    {
        // Insert code to be executed when the user-defined logic is started
        OptixSerialPort = (SerialPort)Owner;
        OptixSerialPort.Timeout = TimeSpan.FromMilliseconds(100);

        SerialPortAdapter = new OptixSerialPortAdapter(OptixSerialPort);
    }

    0 references
    public override void Stop()
    {
        // Insert code to be executed when the user-defined logic is stopped
    }

    2 references
    public class OptixSerialPortAdapter ...

    [ExportMethod]
    0 references
    public void WriteToPrinter(string ActivationKey, string OutgoingText)
    {
        string Status = string.Empty;

        MySerialPrinter.WriteToPrinter(ActivationKey, SerialPortAdapter, OutgoingText, out Status);

        LogicObject.GetVariable("Status").Value = Status;
    }

}
```

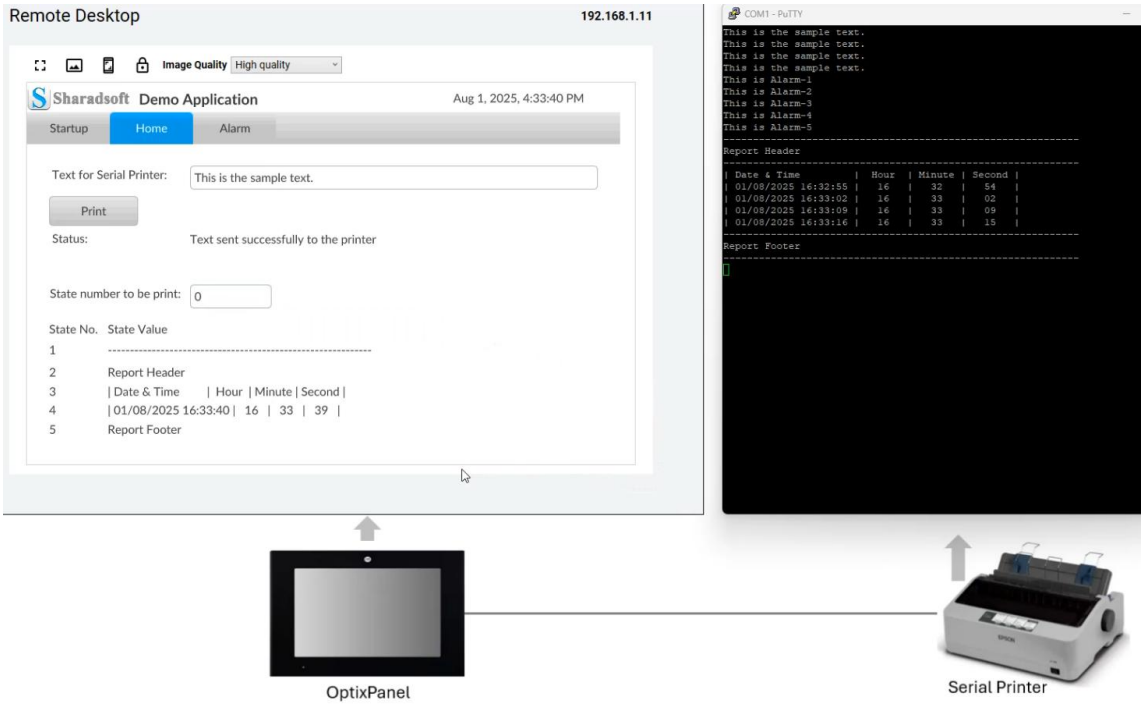
```
2 references
public class OptixSerialPortAdapter : SerialPrinter.ISerialPortWrapper
{
    private readonly dynamic DynamicPort;  // assuming 'SerialPort' type from optix

    1 reference
    public OptixSerialPortAdapter(dynamic DynamicSerialPort)
    {
        DynamicPort = DynamicSerialPort;
    }

    0 references
    public void Open()
    {
        DynamicPort.ExecuteMethod("Open");
    }

    0 references
    public void Write(byte[] buffer)
    {
        DynamicPort.ExecuteMethod("Write", new object[1] { buffer });
    }

    0 references
    public void Close()
    {
        DynamicPort.ExecuteMethod("Close");
    }

}
```

9. Now you are ready with the application for testing. Close Visual Studio if it is open. Save the application and download it to the OptixPanel graphic terminal for testing.

Section3: Demo application snapshot

Sanp-1: PDF file printing from the OptixPanel graphic terminal to a directly connected LAN printer.

Sanp-2: Serial printing from the OptixPanel graphic terminal to a directly connected Serial printer.



www.sharadsoft.com

Feedback/Support

Your comments will help us better serve your HMI printing needs. If you have any suggestions/ require support, then drop an email to ‘info@sharadsoft.com’.

Location: Mumbai, India.

Document updated on 02-Aug-2025