



Analysis of Palis2250 printing processes

Sridharan Ramar, Sheik Mohammed,
Ramarson Technology Developers LLP, Tiruchirapalli. India

July 25, 2016



Contents

| | | |
|-------|---|----|
| 1 | Documentation Road-map | 6 |
| 1.1 | Document Management and Configuration Control Information | 6 |
| 2 | Introduction | 7 |
| 3 | Initialization | 7 |
| 3.1 | Initialization View for Developers | 7 |
| 3.2 | Initialization View for Operators | 14 |
| 3.3 | Initialization View for Service Technician | 14 |
| 3.3.1 | App.config | 15 |
| 3.3.2 | PrinterUnit.config | 15 |
| 3.3.3 | EasyOpc.config | 16 |
| 3.3.4 | Simulation.config | 16 |
| 3.3.5 | ConnectionString.config | 17 |
| 4 | Shutdown | 18 |
| 4.1 | Shutdown View for Developers | 18 |
| 4.2 | Shutdown View for Operators | 20 |
| 4.3 | Shutdown View for Service Technician | 21 |
| 4.3.1 | shutdownRoutine.config | 21 |
| 5 | Cleaning | 22 |
| 5.1 | Cleaning View for Developers | 24 |
| 5.2 | Cleaning view for Operators | 25 |
| 5.3 | Cleaning View for Service Technician | 26 |
| 5.3.1 | printer.config | 26 |
| 5.3.2 | InkjetUnit config | 27 |
| 5.3.3 | EasyOPC.config | 27 |
| 6 | Calibration | 28 |
| 7 | Printhead Calibration | 28 |
| 7.1 | Printhead Calibration View for Developers | 28 |
| 7.2 | Printhead Calibration View for Experts | 37 |
| 7.3 | Printhead Calibration View for Service Technician | 37 |
| 7.3.1 | Printer.Config for Printhead Calibration | 37 |
| 7.3.2 | PrintUnitConfig.Config for Printhead Calibration | 38 |
| 7.3.3 | App.Config for Printhead Calibration | 39 |
| 8 | Cartridge Calibration | 40 |
| 8.1 | Cartridge Calibration View for Developers | 40 |
| 8.2 | Cartridge Calibration View for Experts | 42 |
| 8.3 | Cartridge Calibration View for Service Technician | 42 |

| | | |
|--------|---|----|
| 8.3.1 | Printer.Config for Cartridge Calibration | 42 |
| 8.3.2 | App.Config for Cartridge Calibration | 43 |
| 9 | Density Calibration | 43 |
| 9.1 | Density Cartridge Calibration View for Developers | 44 |
| 9.2 | Density Cartridge Calibration View for Operators | 52 |
| 9.3 | Density Cartridge Calibration View for Service Technician | 53 |
| 9.3.1 | App.Config for Density Calibration | 54 |
| 9.3.2 | Printer.Config for Density Calibration | 54 |
| 9.3.3 | ConnectionString.Config for Density Calibration | 54 |
| 10 | Standby | 55 |
| 10.1 | Standby view for Developers | 56 |
| 10.2 | Standby view for Operators | 59 |
| 10.3 | Standby view for Service Technician | 59 |
| 11 | Printing | 60 |
| 11.1 | Printing View for Developers | 60 |
| 11.2 | Printing View for Operators | 65 |
| 11.3 | Printing View for Service Technician | 65 |
| 11.3.1 | Printer.Config for printing | 66 |
| 11.3.2 | PrintUnit.Config for printing | 66 |
| 11.3.3 | Production.Config | 67 |
| 11.3.4 | TestPattern.Config | 67 |
| 11.3.5 | ConnectionString.Config | 68 |
| 11.3.6 | DataHandling.Config | 68 |
| 11.3.7 | PrintheadSettings.Config | 69 |
| 11.3.8 | App.config from PrintAgent | 69 |
| 11.3.9 | App.config from Main UI | 70 |
| 12 | Conclusion | 71 |

List of Figures

| | | |
|----|--|----|
| 1 | InitializeObjectFlow. | 9 |
| 2 | shows dependent modules. | 10 |
| 3 | shows Initialization Overview. | 11 |
| 4 | shows Initilize PLC. | 12 |
| 5 | shows Boot process. | 13 |
| 6 | shows the Operator View. | 14 |
| 7 | shows the App.config settings | 15 |
| 8 | shows the PrintUnit.config settings | 16 |
| 9 | shows the EasyOpc.config settings | 16 |
| 10 | shows the Simulation.config settings | 16 |

| | | |
|----|---|----|
| 11 | shows the ConnectionString.config settings | 17 |
| 12 | shows the Service Technician View | 17 |
| 13 | shows the Shutdown Object Flow | 18 |
| 14 | shows the ShutDown flow diagram | 20 |
| 15 | shows the ShutDown View for Operator | 21 |
| 16 | shows the Shutdown Routine Config | 21 |
| 17 | shows the Shutdown Service Technician View | 22 |
| 18 | shows the Cleaning Object Mapping Diagram | 25 |
| 19 | shows the Cleaning view For Operator | 26 |
| 20 | shows the Cleaning Printer Config | 27 |
| 21 | shows the Cleaning PLC Faults | 27 |
| 22 | shows theEasy Opc Config | 27 |
| 23 | shows the Printhead Calibration Object Flow | 30 |
| 24 | Shows the enable and disable stepper power | 31 |
| 25 | Shows the SetXOffset function flow | 32 |
| 26 | Shows the Move YStepper To ReferencePosition function flow | 33 |
| 27 | Shows the Move Y Stepper funtion flow | 34 |
| 28 | Shows the unction MovePhiStepperToReferencePosition function flow | 35 |
| 29 | Shows the MovePhiStepper function flow | 36 |
| 30 | Shows the Printhead Calibration expert user workflow | 37 |
| 31 | shows the Printer Config | 38 |
| 32 | shows the Print.Unit Config | 39 |
| 33 | shows the App Config | 40 |
| 34 | shows the Cartridge Calibration Overview | 41 |
| 35 | shows the cartridge calibration view | 42 |
| 36 | shows the Cartridge Calibration PrinterConfig | 43 |
| 37 | shows the Cartridge Calibration App Config | 43 |
| 38 | shows the DensityDeviation | 43 |
| 39 | shows the unit arrangment in KJ4B-S Series 1200dpi printhead | 44 |
| 40 | shows the density calibration create new profile | 45 |
| 41 | shows the density calibration edit profile | 45 |
| 42 | shows the density calibration delete profile | 46 |
| 43 | shows the density calibration save profile | 46 |
| 44 | shows the density calibration create profile | 47 |
| 45 | shows the density calibration edit profile | 48 |
| 46 | shows the density calibration delete profile | 49 |
| 47 | shows the density calibration save profile | 50 |
| 48 | shows the density calibration cancel metadata profile editing | 51 |
| 49 | shows the density calibration print Test Pattern | 52 |
| 50 | shows the density calibration for expert user | 53 |
| 51 | shows the density calibration App.Config | 54 |
| 52 | shows the density calibration Printer.Config | 54 |
| 53 | shows the density calibration ConnectionStringConfig.Config | 55 |

| | | |
|----|--|----|
| 54 | shows the overall standby object mapping | 56 |
| 55 | shows the production standby funtion flow diagram | 57 |
| 56 | shows the standby function flow diagram | 58 |
| 57 | shows the operator flow diagram | 59 |
| 58 | shows the Printer.config for standby operator | 59 |
| 59 | shows the LoadJob object flow | 60 |
| 60 | shows the TestPattern object flow | 61 |
| 61 | shows the PrintJob object flow | 62 |
| 62 | shows the stop printing object flow | 63 |
| 63 | shows the Load Job function flow1 | 63 |
| 64 | shows the Load Job function flow2 | 64 |
| 65 | shows the operator sequence of actions for printing | 65 |
| 66 | shows the Printer.Config for printing | 66 |
| 67 | shows the PrintUnit.Config for printing | 67 |
| 68 | shows the Production.Config for printing | 67 |
| 69 | shows the TestPattern.Config for printing TestPatterns | 68 |
| 70 | shows the ConnectionString.Config | 68 |
| 71 | shows the DataHandling.Config for printing | 69 |
| 72 | shows the PrintheadSettings.Config for printing | 69 |
| 73 | shows the App.Config for PrintAgent | 70 |
| 74 | shows the App.Config for MainUI | 71 |

1 Documentation Road-map

This document gives an overview of the analysis report made on Printing process on Palis Test System software solution.

- Section 1.1 explains revision history.
- Section 2. Gives an Introduction to the analysis and its goals
- Section 3. Describes the Initialization and presents different views like Initialization View for Developers, Operators and Service Technicians
- Section 4. Describes the Shutdown process and presents different views like Shutdown View for Developers, Operators and Service Technicians
- Section 5. Describes the Cleaning view for Developers, Operators and Service Technicians
- Section 6. Describes the Calibration process
- Section 7. Describes the Printhead Calibration view for Developers, Operators and Service Technicians
- Section 8. Describes the Cartridge Calibration view for Developers, Operators and Service Technicians
- Section 9. Describes the Density Calibration view for Developers, Operators and Service Technicians
- Section 10. Describes the Standby view for Developers, Operators and Service Technicians
- Section 11. Describes the Printing process and describes views for Developers, Operators and Service Technicians
- Section 12. Gives a Conclusion of the above analysis

1.1 Document Management and Configuration Control Information

- Revision Number: 2016060001
- Revision Release Date: July 25, 2016
- Purpose of Revision: Initial Version
- Scope of Revision: none

2 Introduction

Goal of analysis are the following

- To create a map of Palis2250 software system.
- Find interdependencies and know module interaction in this process.
- Help to find duplicate object
- Improve the existing software design

The analysis is performed by keeping in mind different stakeholders

1. View for Developers
2. View for Operators
3. View for Service Technician

For each stockholder there is a dedicated view which will help him to better understand the relevant information to know from his perspective. The goal of this analysis is to understand the current model of the software system, the interdependencies and get glims on overall architecture. Analyse each software process and describe modules that are interacting in individual processes. The diagrams created will illustrate and help to understand each process better and help to find duplicate object.

3 Initialization

Initialization is the process by which all essential components are initialized for example the nodes, PLC and software agents (Print Agent) are started and corrective actions are triggered so that the machine is available for Printing.

3.1 Initialization View for Developers

The below diagram explains how the process initialize the machine is implemented in the software, what objects are interacting in the process.

- InitializeNode
 - PrintNodes [Boot and shutdown the print nodes]
 - IBootStrategy [Boot the node]
 - WakeOnLanBootStratgey [Wake up all the nodes via WakeOnLAN)
 - IWakeOnLanService [Wake up the node via wakeOnLan]
 - Magic Packet [create and send magic packet]

- Initialize PLC
 - PlcComplexFunction [run the function and return when its finished]
 - PlcFunction [Run the PLC function]
 - PLCImpulse [Send the impulse]
 - PlcAccess [Read and write the values to PLC]
- Wait for PrintAgent
 - IPrintEngine [Talking to controller for Controlling all the color engine]
 - IColorEngine [Manage the single color engine]
 - IPrintAgentClient [Proxy for forwarding the call to service]
 - BaseClient [Base class for providing the service call from WCF service]
 - Mexhelper [Provide the information about meta-data]
- Acknowledge Errors
 - IPrintEngine [Talking to controller for controlling all the color engines]
 - IColorEngine [Manage the single color engine]
 - IPrintAgentClient [Proxy for forwarding the call to service]
 - BaseClient [Base class for providing the service Call from WCF Service]

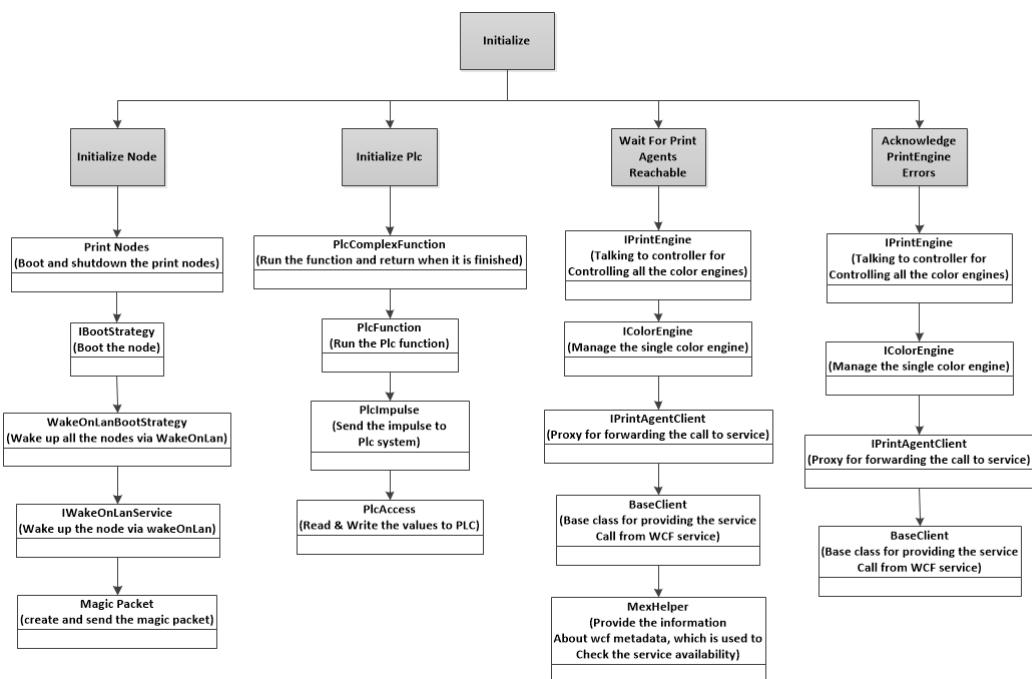


Figure 1: InitializeObjectFlow.

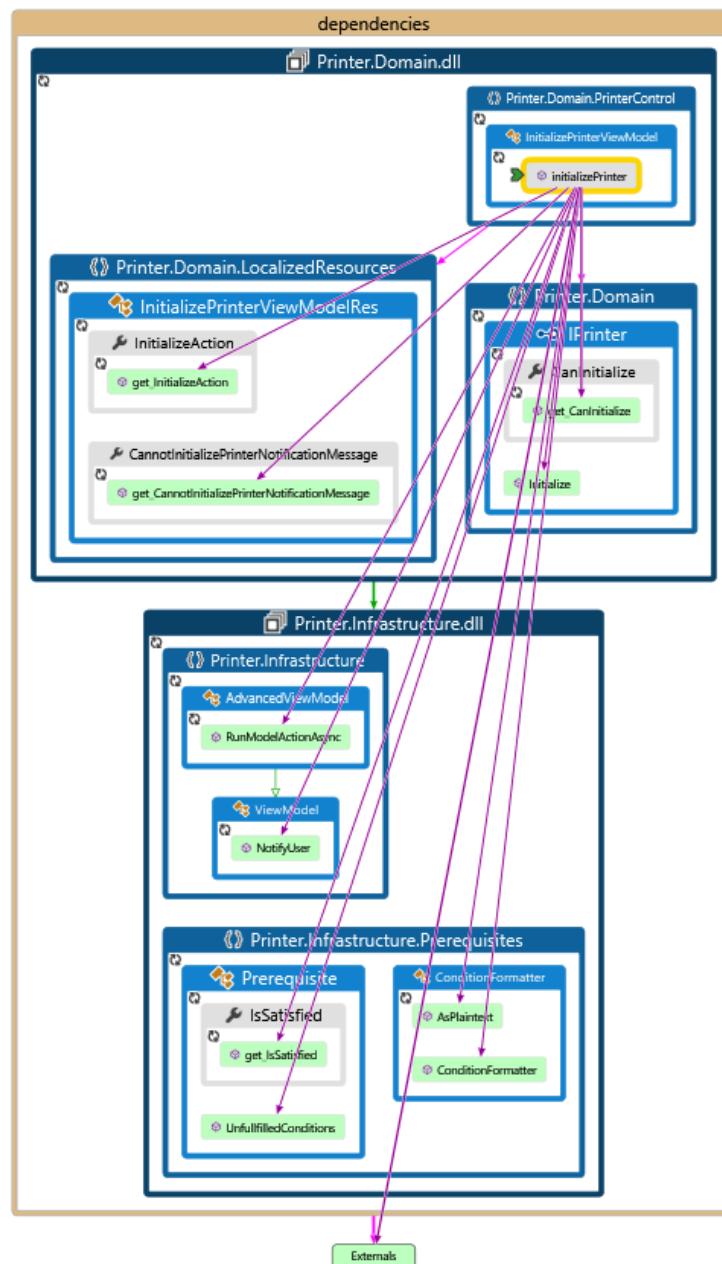


Figure 2: shows dependent modules.

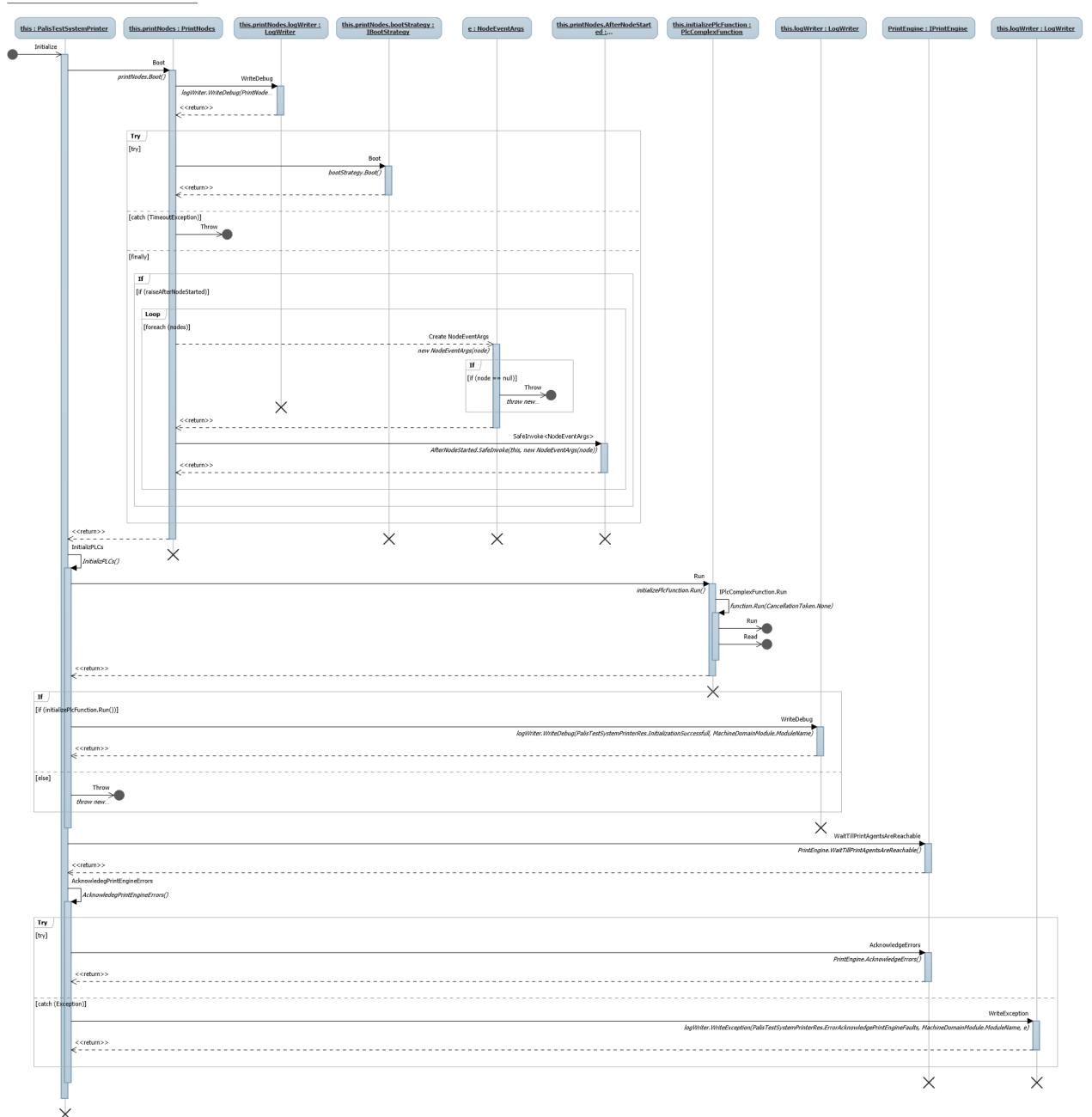


Figure 3: shows Initialization Overview.

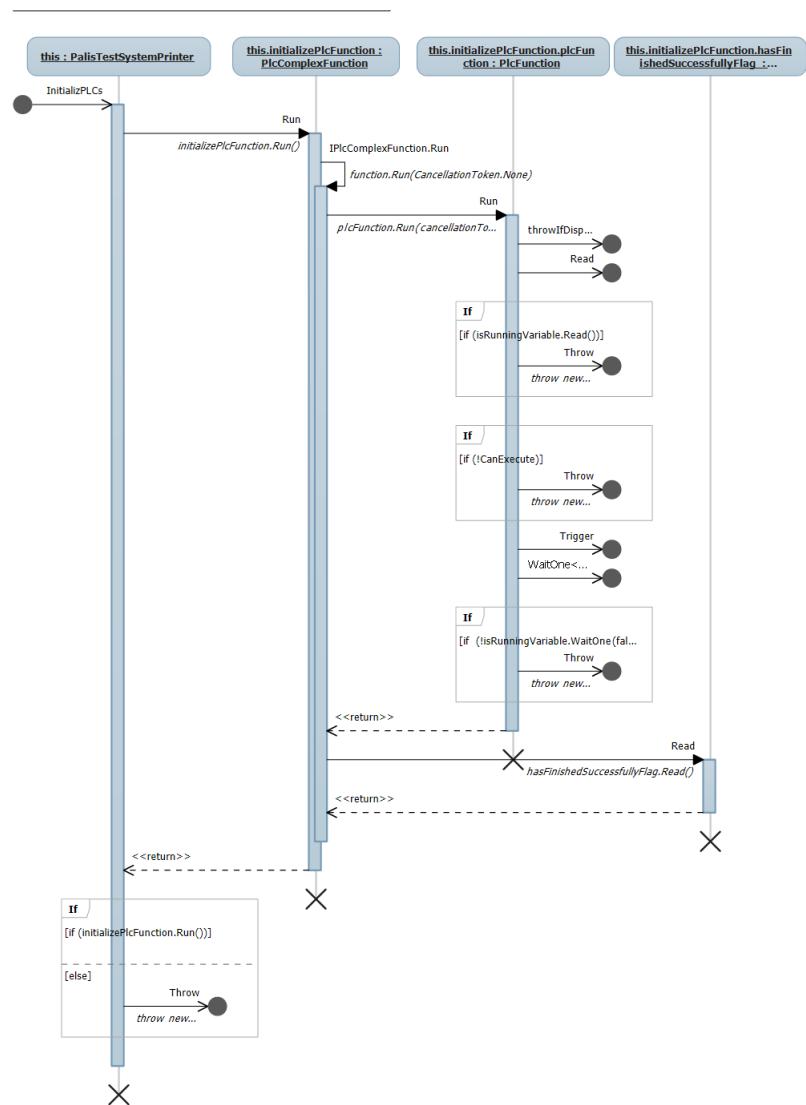


Figure 4: shows Initilize PLC.

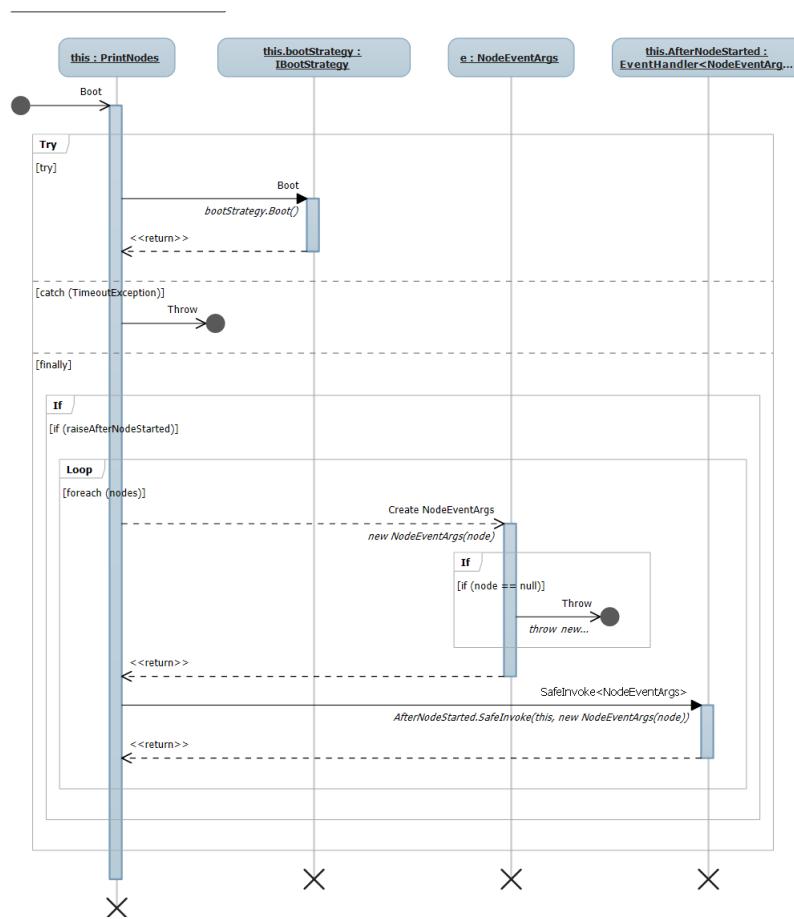


Figure 5: shows Boot process.

3.2 Initialization View for Operators

Looking from operator perspective; Once the application is started an operator can click the button Initialize and verify the following

1. Check If the machine is on, if machine is not on call Palis service team.

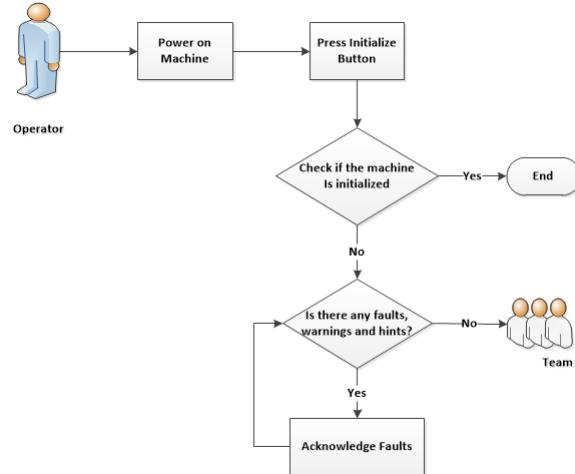


Figure 6: shows the Operator View.

3.3 Initialization View for Service Technician

Check the following config files during deployment

- App.config
- Printer.config
- EasyOpc.config
- Simulation.config
- Connectionstring.config

For a technician who is deploying the software; it would be interesting to know what are all the configs, that needs to be taken care. The above listed config files has to be configured appropriately in order to setup the system correctly, so that the process initialize is working as expected.

3.3.1 App.config

In App.config the endpoint address for the Print Agents has to be configured correctly, please refer the Node IPAddress from Initialization View for Service Technician for understanding which IP Address to configure correctly. Also check the PrinterStatusService address, if they are configured correctly.

```

<WcfConfiguration>
  <endPoints>
    <!--<EndPoint name="PrinterStatusClient" uri="net.tcp://localhost:8010/PrinterStatus" isDebugEnabled="true"/>
    <!--<EndPoint name="PrinterStatusService" uri="net.tcp://localhost:8010/PrinterStatus" isDebugEnabled="true"/>
    <!--<EndPoint name="PrinterConfigurationClient" uri="net.tcp://localhost:9001/PrinterConfigurationService" isDebugEnabled="true"/>
    <!--<EndPoint name="PrinterConfigurationService" uri="net.tcp://localhost:9001/PrinterConfigurationService" isDebugEnabled="true"/>
    <!--<EndPoint name="CleaningService" uri="net.tcp://localhost:9000/cleaningService" isDebugEnabled="true"/>
    <!--<EndPoint name="CleaningClient" uri="net.tcp://localhost:9000/CleaningService" isDebugEnabled="true"/>
    <!--<EndPoint name="UserManagementService" uri="net.tcp://localhost:9002/UserManagementService" isDebugEnabled="true"/>
    <!--<EndPoint name="UserManagementClient" uri="net.tcp://localhost:9002/UserManagementService" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintheadCalibrationService" uri="net.tcp://localhost:9003/PrintheadCalibrationService" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintheadCalibrationClient" uri="net.tcp://localhost:9003/PrintheadCalibrationService" isDebugEnabled="true"/>
    <!--<EndPoint name="TestPatternService" uri="net.tcp://localhost:9004/TestPatternService" isDebugEnabled="true"/>
    <!--<EndPoint name="TestPatternClient" uri="net.tcp://localhost:9004/TestPatternService" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintEnginePowerSupplyService" uri="net.tcp://localhost:9005/PrintEnginePowerSupplyService" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintEnginePowerSupplyClient" uri="net.tcp://localhost:9005/PrintEnginePowerSupplyService" isDebugEnabled="true"/>
    <!--<EndPoint name="DensityCalibrationService" uri="net.tcp://localhost:9006/DensityCalibrationService" isDebugEnabled="true"/>
    <!--<EndPoint name="DensityCalibrationClient" uri="net.tcp://localhost:9006/DensityCalibrationService" isDebugEnabled="true"/>
    <!--<EndPoint name="CartridgeCalibrationService" uri="net.tcp://localhost:9007/CartridgeCalibrationService" isDebugEnabled="true"/>
    <!--<EndPoint name="CartridgeCalibrationClient" uri="net.tcp://localhost:9007/CartridgeCalibrationService" isDebugEnabled="true"/>

    <!--<EndPoint name="PrintAgentNode1" uri="net.tcp://10.0.50.166:8083/PrintAgent" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintAgentNode2" uri="net.tcp://10.0.50.167:8083/PrintAgent" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintAgentNode3" uri="net.tcp://10.0.50.168:8083/PrintAgent" isDebugEnabled="true"/>
    <!--<EndPoint name="PrintAgentNode4" uri="net.tcp://10.0.50.169:8083/PrintAgent" isDebugEnabled="true"/>

    <!--<EndPoint name="PrinterPositioningService" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true"/>
    <!--<EndPoint name="PrinterPositioningClient" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true"/>
    <!--<EndPoint name="CurrentOrderService" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true"/>
    <!--<EndPoint name="CurrentOrderClient" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true"/>
  </endPoints>
</WcfConfiguration>

```

Figure 7: shows the App.config settings

3.3.2 PrinterUnit.config

In printunit config the following IP and MAC address has to be configured correctly in order to setup the system correctly. Please refer the Initialization View for Service Technician for understanding which IP Address to configure correctly.

```

<printUnit currentMode="DefaultMode" downloadedLocalRipFolder="C:\Spool">
    <isStepperReferenceMovementEnabled>"true"</isStepperReferenceMovementEnabled>
    <printdataFolder>".\RipMetadata"</printdataFolder>
    <showStartPrintNodes>"true"</showStartPrintNodes>
    <testpatternFolder>":\Spool\Testpattern"</testpatternFolder>
    <waveformFolder>""</waveformFolder>
    <maxFaultPrintheadTemperature>"40"</maxFaultPrintheadTemperature>
    <minimalTemperatureExcedanceDuration>"00:00:05"</minimalTemperatureExcedanceDuration>
    <printHeadWarmUptime>"00:00:40"</printHeadWarmUptime>
</printUnit>

<printNodes>
    <printNode colorId="Cyan" connectedPrintheads="20" daisyChainStart="1">
        <ipAddress>"10.0.50.166"</ipAddress>
        <macAddress>"74-D4-35-16-54-8E"</macAddress>
        <meteorConfiguration>"\\10.0.50.166\MeteorConfig\meteor.cfg"</meteorConfiguration>
        <printAgentConfiguration>"PrintAgentNode1"</printAgentConfiguration>
        <printdataFolder>"\\10.0.50.166\Transfer"</printdataFolder>
        <deltaForUpperWarning>"1.2"</deltaForUpperWarning>
        <deltaForLowerWarning>"2"</deltaForLowerWarning>
        <targetDensity>:"1.12"/>
    </printNode>
    <printNode colorId="Magenta" connectedPrintheads="20" daisyChainStart="1">
        <ipAddress>"10.0.50.167"</ipAddress>
        <macAddress>"74-D4-35-1E-C5-32"</macAddress>
        <meteorConfiguration>"\\10.0.50.167\MeteorConfig\meteor.cfg"</meteorConfiguration>
        <printAgentConfiguration>"PrintAgentNode2"</printAgentConfiguration>
        <printdataFolder>"\\10.0.50.167\Transfer"</printdataFolder>
        <deltaForUpperWarning>"1.2"</deltaForUpperWarning>
        <deltaForLowerWarning>"2"</deltaForLowerWarning>
        <targetDensity>:"0.95"/>
    </printNode>
    <printNode colorId="Yellow" connectedPrintheads="20" daisyChainStart="1">
        <ipAddress>"10.0.50.168"</ipAddress>
        <macAddress>"74-D4-35-1E-DC-31"</macAddress>
        <meteorConfiguration>"\\10.0.50.168\MeteorConfig\meteor.cfg"</meteorConfiguration>
        <printAgentConfiguration>"PrintAgentNode3"</printAgentConfiguration>
        <printdataFolder>"\\10.0.50.168\Transfer"</printdataFolder>
        <deltaForUpperWarning>"1.2"</deltaForUpperWarning>
        <deltaForLowerWarning>"2"</deltaForLowerWarning>
        <targetDensity>:"0.58"/>
    </printNode>
    <printNode colorId="Black" connectedPrintheads="20" daisyChainStart="1">
        <ipAddress>"10.0.50.169"</ipAddress>
        <macAddress>"74-D4-35-1F-5D-58"</macAddress>
        <meteorConfiguration>"\\10.0.50.169\MeteorConfig\meteor.cfg"</meteorConfiguration>
        <printAgentConfiguration>"PrintAgentNode4"</printAgentConfiguration>
        <printdataFolder>"\\10.0.50.169\Transfer"</printdataFolder>
        <deltaForUpperWarning>"1.2"</deltaForUpperWarning>
        <deltaForLowerWarning>"2"</deltaForLowerWarning>
    </printNode>
</printNodes>

```

Figure 8: shows the PrintUnit.config settings

3.3.3 EasyOpc.config

On EasyOpc client the IPAddress of the OPC has to be configured correctly, please refer the Initialization View for Service Technician for knowing the correct PLC address.

```

<Printer_EasyOpcClient>
    <endPointUrl>"opc.tcp://10.0.50.180:4848/"</endPointUrl>
    <sessionTimeout>"15"</sessionTimeout>
    <keepAliveInterval>"10"</keepAliveInterval>
    <sendKeepAliveSignal>"false"</sendKeepAliveSignal>
    <keepAliveSignalAddress>"nsu=http://www.boschrexroth.com/IndraLogic;s=Application.UI_SPS_GVL.b_Lebensbit_UI"/>
</Printer_EasyOpcClient>

```

Figure 9: shows the EasyOpc.config settings

3.3.4 Simulation.config

While running in the production environment make sure the simulation flag is set to low/false.

```

<?xml version="1.0" encoding="utf-8"?>
<Printer_Bootstrapping_Simulation simulateOpc="true"/>

```

Figure 10: shows the Simulation.config settings

3.3.5 Connectionstring.config

To make the initialize process work correctly, make sure the connection string which is used to connect to the local database is configured correctly.

```

<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <add
    name="MachineContext"
    providerName="System.Data.SqlClient"
    connectionString="<!--
      Server=.\SQLEXPRESS;
      Database=palitestBranch;
      integrated Security=true;
      MultipleActiveResultSets=true;-->"/>
</connectionStrings>

```

Figure 11: shows the ConnectionString.config settings

The service technician can also check the network configuration illustrated in this diagram; how the nodes and the PLCs are configured and make sure for a correct setup of the machine and to initialize the machine, the network settings has to be configured in the following manner.

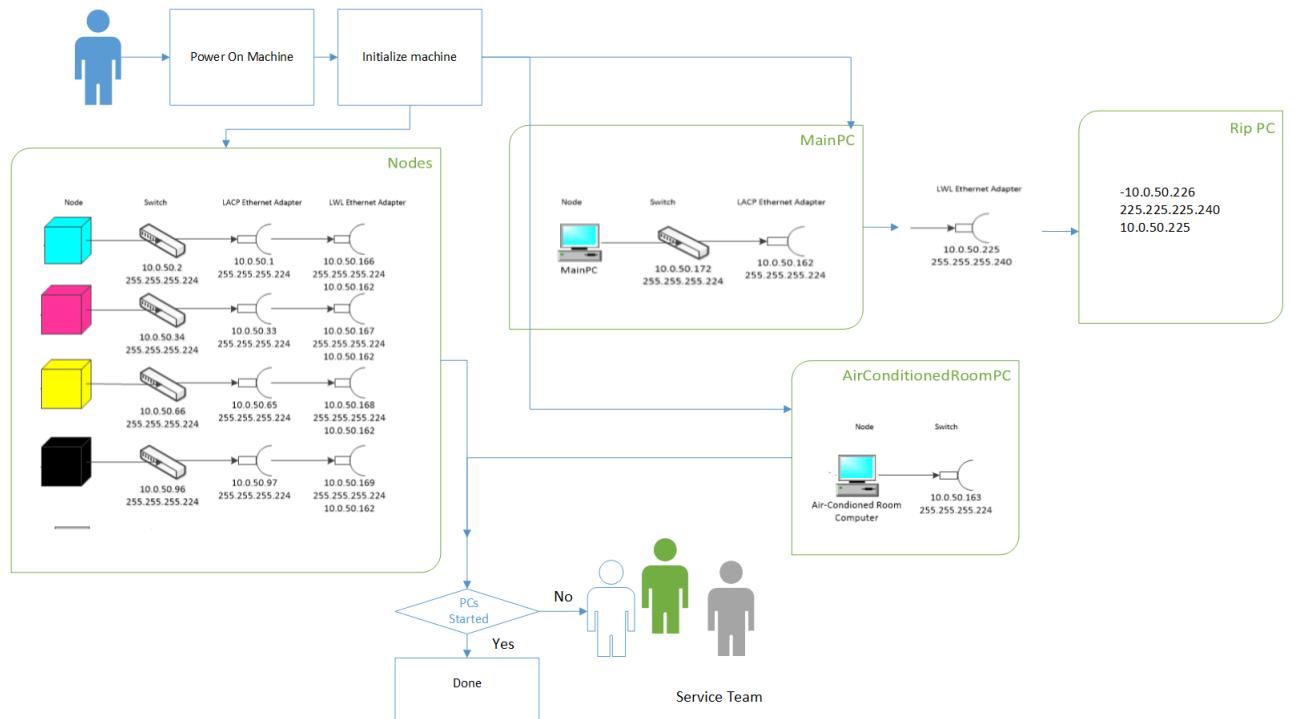


Figure 12: shows the Service Technician View.

4 Shutdown

Shutdown is the process to switch off everything (out of production) including the Main user interface and Air-Conditioned room computers. This is the successor of initialization, without switch on the machine we cannot switch off. IMPORTANT: The print heads power should be off always in this situation, otherwise there is no way to turn off the power.

4.1 Shutdown View for Developers

The below diagram give an illustration of shutdown process, how the objects are linked together, what is the role of objects and so on.

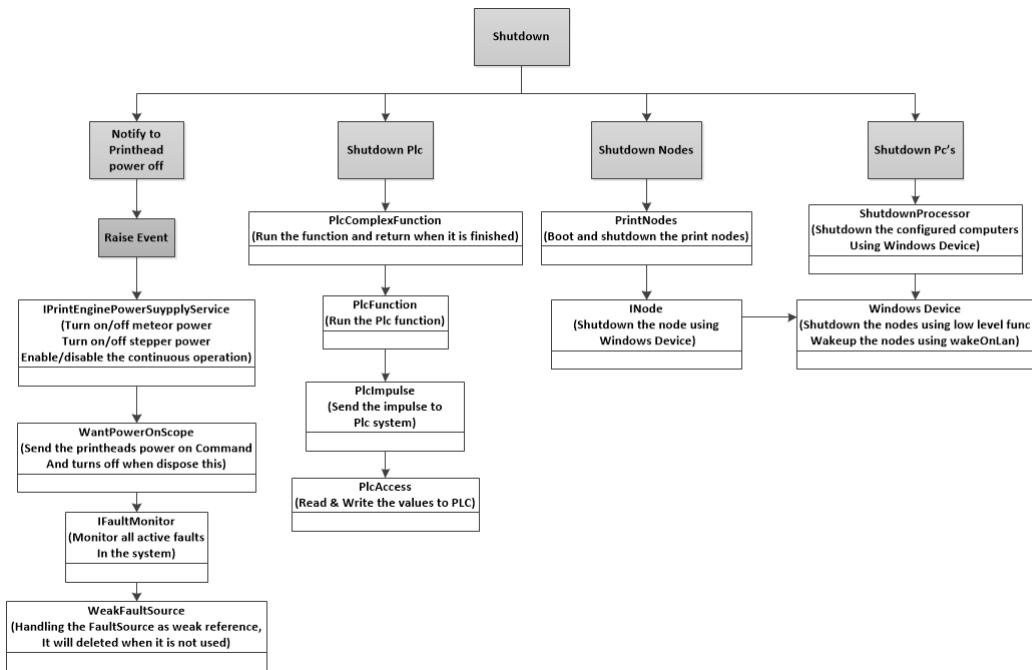


Figure 13: shows the Shutdown Object Flow

- Notify to Print heads power off
 - Raise Event [Raise the event when shutdown process trigger and notify to power supply service to turn off the print heads power].
 - IPrintEnginePowerSupplyService [Receives the event and turn off the power]
 - WantPowerOnScope [turn on power and Dispose this instance].
 - IFaultMonitor [Add/remove the active faults]
 - WeakFaultSource [Clean up the reference if it is unused].

- Shutdown PLC
 - PlcComplexFunction [run the function and return when it's finished].
 - PlcFunction [Run the PLC function].
 - PLCImpulse [Send the impulse].
 - PlcAccess [Read and write the values to PLC].
- Shutdown Nodes
 - Print Nodes [Boot and shutdown the print nodes].
 - INode [Shutdown the single node using Windows Device].
 - WindowsDevice [Initialize the nodes using wakeOnLan, shutdown the nodes using low level function].
- Shutdown PC's
 - Shutdown Processor [Shutdown the configured IP address using Windows Device].
 - Windows Device [Initialize the nodes using wakeOnLan, shutdown the nodes using low level function].

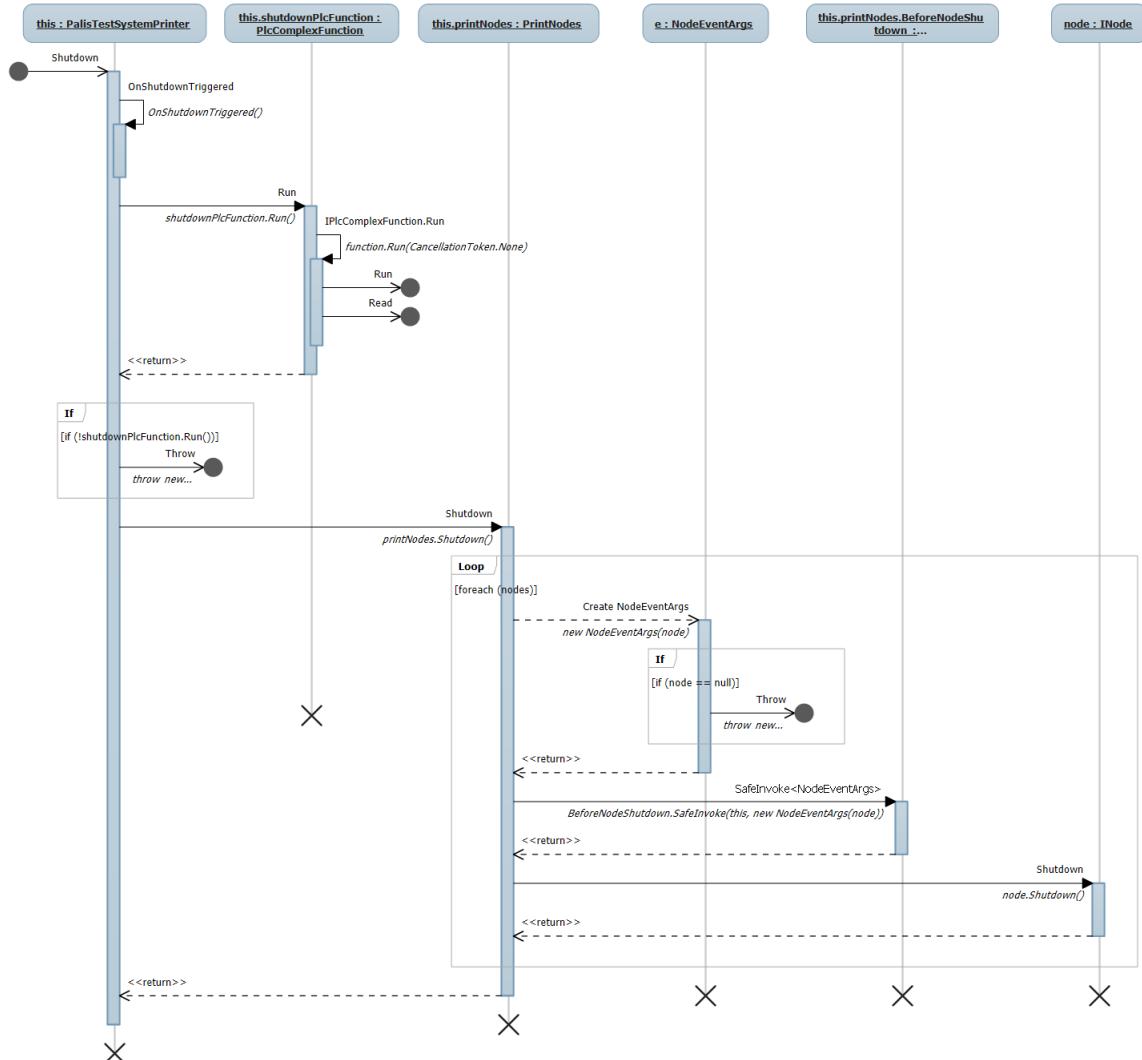


Figure 14: shows the ShutDown flow diagram

4.2 Shutdown View for Operators

Looking from the operator perspective, once the production over and if they decided to shutting down the machine, the operator has to check some important points after shutdown the machine from the user interface.

- Verify if the Air-Conditioned room computer shutdown properly.
- Verify if the Main PC shutdown properly.

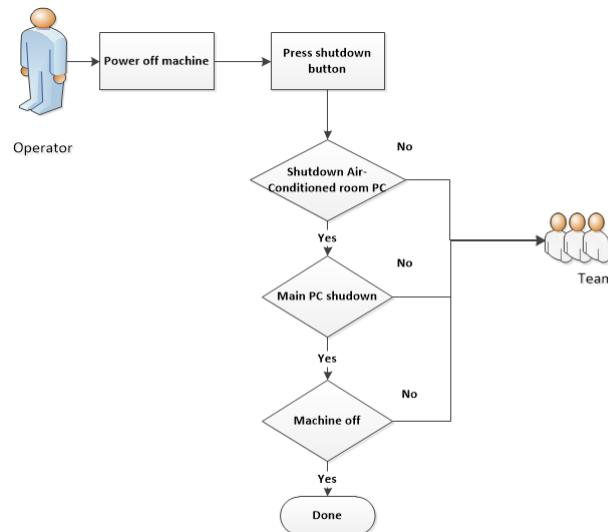


Figure 15: shows the ShutDown View for Operator

4.3 Shutdown View for Service Technician

Shutdown process is the successor of Initialization, so the Service Technician has to check the configuration file of initialization work-flow, but additionally he/she has to check the following configuration for shutting down the computers.

4.3.1 shutdownRoutine.config

```

<?xml version="1.0" encoding="utf-8" ?>
<shutdownRoutineConfiguration>
  <PCsToShutdown>
    <add identifier="MainPC" ipAddress="" />
    <add identifier="Air-Conditioned Room PC" ipAddress="" />
  </PCsToShutdown>
</shutdownRoutineConfiguration>
  
```

Figure 16: shows the Shutdown Routine Config

The service Technician also need to check the IPAddress of Main user interface computer and Air-Conditioned room computer.

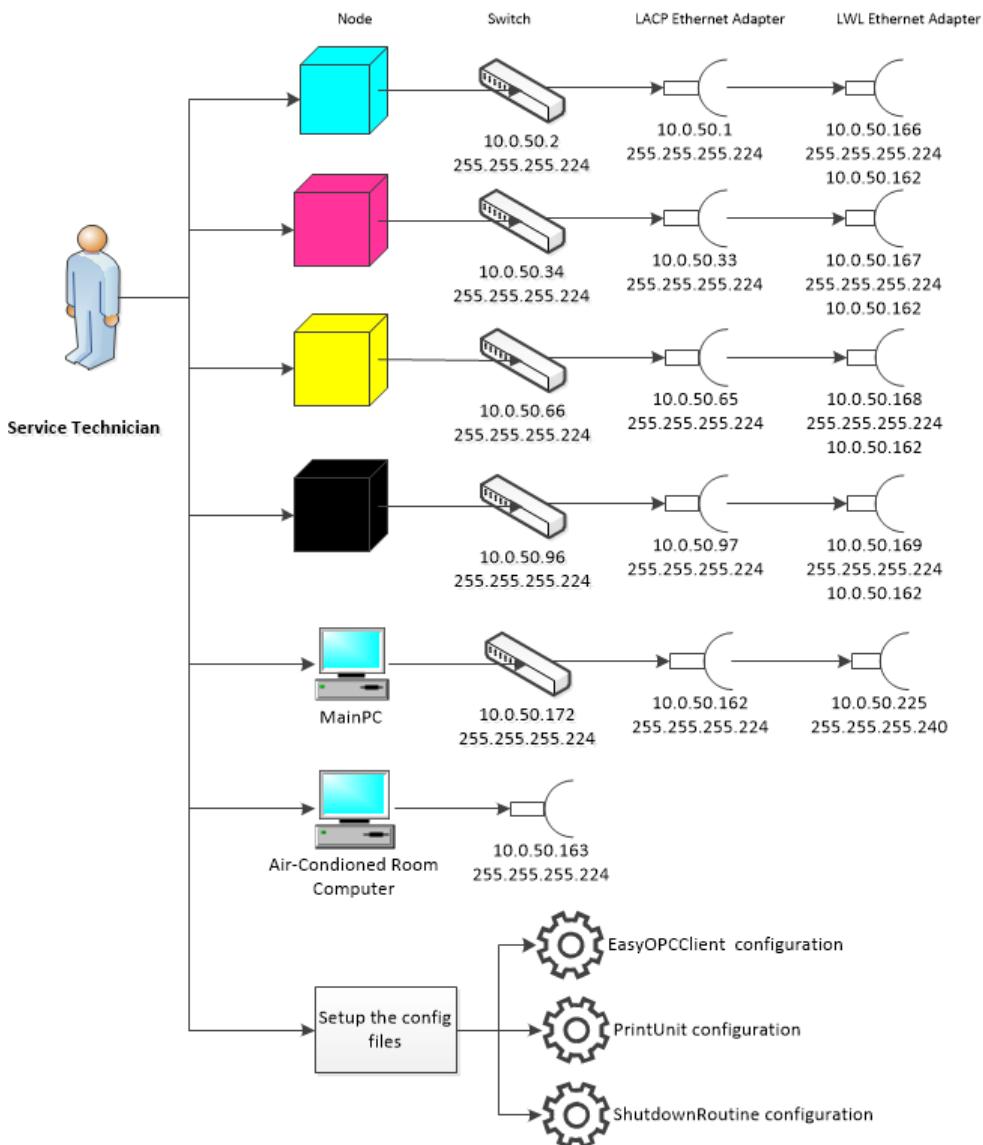


Figure 17: shows the Shutdown Service Technician View

5 Cleaning

Cleaning is an important process in our system, because it will help to maintain the printing quality in a consistent manner. While printing we have to ensure that maximum nozzle availability is there. By doing appropriate cleaning (Refreshing, Cleaning, Intensive Cleaning and Advanced Cleaning) the operator can make sure in all available printheads all the nozzles are firing. Note : We do not need to clean for each production job.

An use case when the nozzles can be dried up, For example: after a very long

production when we stop the machine or we leave it idle for some time, then the inks stays in the nozzles and it will get dried in that idle state, when we go on with the next print. Our quality may not be good; because the dried inks may be blocking some nozzles, so in order to avoid this problem always we have to do the cleaning at the beginning or before starting to do production when the machine is idle for longer time. It is advisable to perform a cleaning periodically to ensure high quality print results. The operator have the possibility to either select individual print heads or the whole cartridge for cleaning process.

- We can do the following Process

1. Refreshing.
2. Cleaning
3. Intensive Cleaning
4. Advanced Cleaning
5. Test Move Wipers
6. Move Wipers to Service Position
7. Move Wipers to Reference position

| Parameters | Refreshing | Cleaning | Intensive | Advanced |
|-----------------------------|------------|------------|------------|------------|
| Position after the cleaning | Inspection | Inspection | Inspection | User Input |
| BackflowPurgeVolume | 0 | 0 | 10 | User Input |
| CleanWithBackflow | No | No | Yes | User Input |
| PurgeVolume | 5 | 10 | 20 | User Input |
| Use wipers | Yes | Yes | Yes | User Input |

Table 1: Parameters for different Types of Cleaning.

If we use the wiper, the position can be capping or inspection, else If we are not using wiper, the position should be in capping.

- Test Move wipers
 - Moving the wiper from one side to another side.
 - Set the print head valves from the UI
 - Execute the PLC complex function.
- Move WipersToReferencePosition
 - Moving the wiper to reference position
 - Set the print head valves from one side to another side.

- Execute the PLC complex function.
- Move WipersToServicePosition
 - Moving the wiper to service position
 - Set the print head valves from one side to another side.
 - Execute the PLC complex function.

5.1 Cleaning View for Developers

The below diagram explains how the process cleaning is implemented in the software, what objects are interacting in the process.

- ICleaningUnitClient [interface to talk with wcf to exchange the information to View-Model]
- DuplexClient [Generic class which handle all the transaction between the client and wcf service adapter]
- ICleaningWcfAdapter [Wcf service for cleaning process]
- AsynchronousOperationExecutor [Responsible to perform the asynchronous operation]
- ICleaningService [Application level service to talk with the domain]
- ICleaningUnit [Domain logic to perform the cleaning operation]
- ColorSystem [Handling the single color system properties and methods]
- Cartridge [Base object which handle the cartridge related properties and methods]
- PrintheadValves [Dealing with valves for cleaning]
- PlcFlag [Set/reset the cleaning for specified valve]
- IPLCComplexFunction [Complex PLC function for execute the cleaning process in the PLC system]
- PLCFunction [Execute the function in the PLC]
- PLCImpulse [Send the impulse to PLC]
- IPlcAccess [Read/Write the values to the PLC]

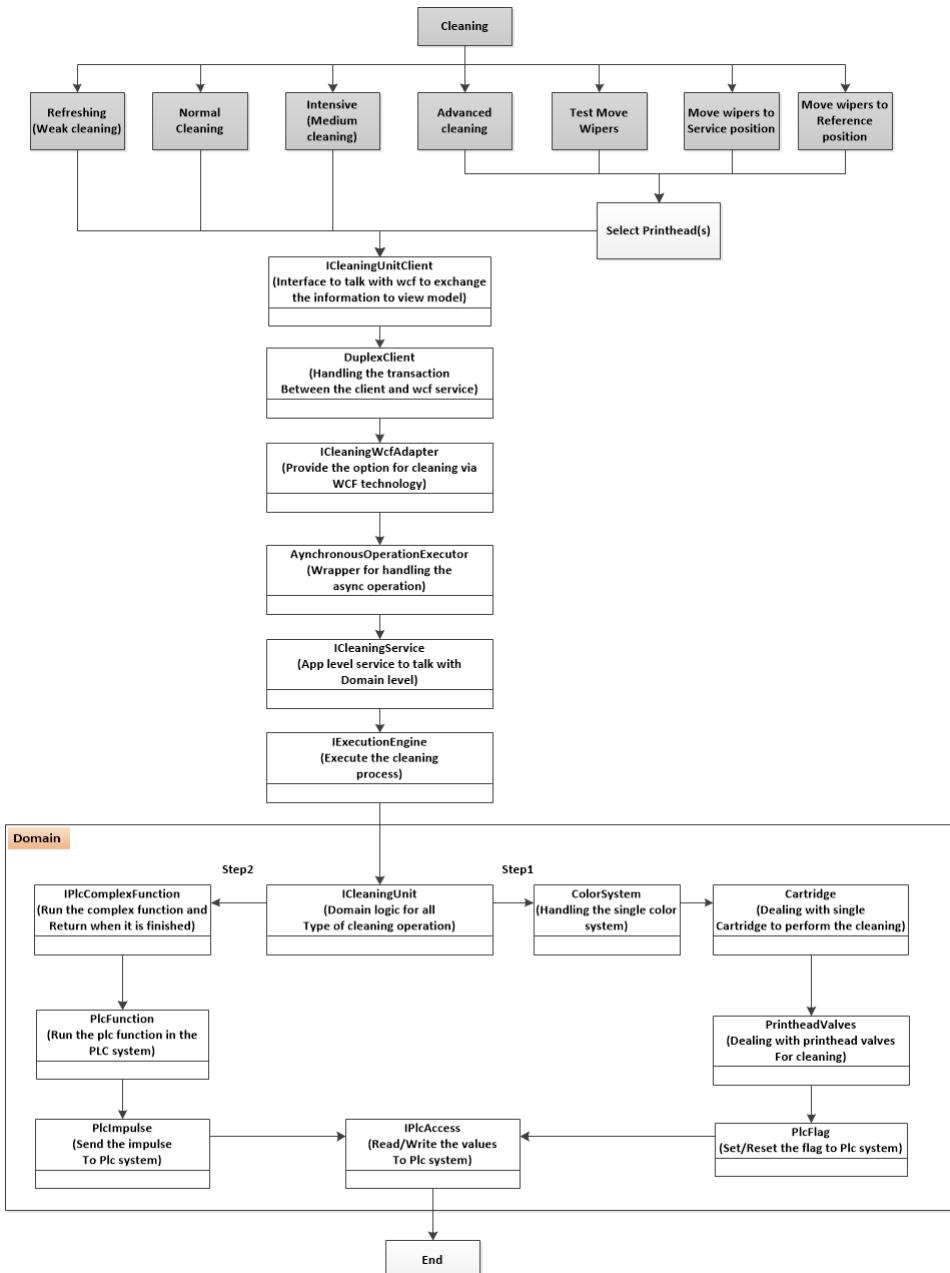


Figure 18: shows the Cleaning Object Mapping Diagram

5.2 Cleaning view for Operators

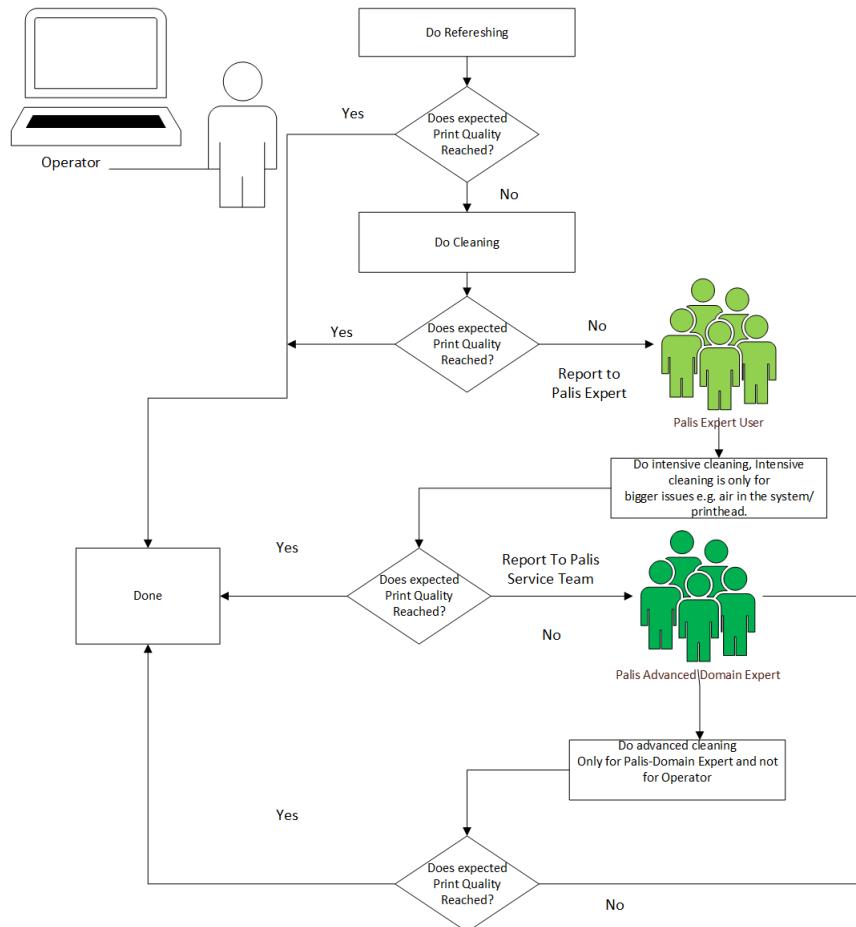


Figure 19: shows the Cleaning view For Operator

5.3 Cleaning View for Service Technician

The service technician needs to verify the following config to make sure cleaning is running correctly.

- Printer.config.
- InkjetUnit.config.
- EasyOPC.config.

5.3.1 printer.config

```

<?xml version="1.0" encoding="utf-8" ?>
<Printer printingDirection="Width"
    machineIdentifier="Palis750"
    isScanPrinter="false">

    <standby currentStrategy="overnight">
        <overnightStandbyStrategy spitLength="1" timeBetweenSpits="60"/>
        <productionStandbyStrategy spitLength="1" timeBetweenSpits="60"/>
    </standby>

    <colors>
        <color id="Cyan" argb="#00FFFF" ...</color>
        <color id="Red" argb="#FF0000" ...</color>
        <color id="Yellow" argb="#FFFF00" ...</color>
        <color id="Black" argb="#000000" ...</color>
    </colors>
</Printer>

```

Figure 20: shows the Cleaning Printer Config

5.3.2 InkjetUnit config

```

<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_MT130_C5_X_Status"
    name="Kassette 5 MT130-X Status" description="Fehlerhafte Statusmeldung MT130 X-Achse für Kassette 5" module="Inkjet Einheit" fix="-" priority="4"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_MT130_C1_Y_Status"
    name="Kassette 1 MT130-Y Status" description="Fehlerhafte Statusmeldung MT130 Y-Achse für Kassette 1" module="Inkjet Einheit" fix="-" priority="4"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_MT130_C2_Y_Status"
    name="Kassette 2 MT130-Y Status" description="Fehlerhafte Statusmeldung MT130 Y-Achse für Kassette 2" module="Inkjet Einheit" fix="-" priority="4"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_MT130_C3_Y_Status"
    name="Kassette 3 MT130-Y Status" description="Fehlerhafte Statusmeldung MT130 Y-Achse für Kassette 3" module="Inkjet Einheit" fix="-" priority="4"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_MT130_C4_Y_Status"
    name="Kassette 4 MT130-Y Status" description="Fehlerhafte Statusmeldung MT130 Y-Achse für Kassette 4" module="Inkjet Einheit" fix="-" priority="4"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_MT130_C5_Y_Status"
    name="Kassette 5 MT130-Y Status" description="Fehlerhafte Statusmeldung MT130 Y-Achse für Kassette 5" module="Inkjet Einheit" fix="-" priority="4"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_Slave_Reinigung_OK"
    name="Slave Reinigung" description="Es ist ein Fehler beim Slave der Reinigung aufgetreten" module="Inkjet Einheit" fix="-" priority="1"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_P_ZHT_Ueberdruck_OK"
    name="Zwischentank Überdruck" description="Der Überdruck in den Zwischentanks ist nicht in Ordnung." module="Inkjet Einheit" fix="-" priority="1"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_C1_Vorlauftank_Fuellstand_Analogeingang_Stoerung"
    name="Farbe1 Vorlauftank Analogeingang" description="Am Analageingang des Vorlauftanks der Farbe 1 liegt ein Fehler vor." module="Inkjet Einheit" fix="-" priority="1"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_C2_Vorlauftank_Fuellstand_Analogeingang_Stoerung"
    name="Farbe2 Vorlauftank Analogeingang" description="Am Analageingang des Vorlauftanks der Farbe 2 liegt ein Fehler vor." module="Inkjet Einheit" fix="-" priority="1"/>
<faultDescription id="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.SPS_UI_GVL_b_C3_Vorlauftank_Fuellstand_Analogeingang_Stoerung"
    name="Farbe3 Vorlauftank Analogeingang" description="Am Analageingang des Vorlauftanks der Farbe 3 liegt ein Fehler vor." module="Inkjet Einheit" fix="-" priority="1"/>

```

Figure 21: shows the Cleaning PLC Faults

5.3.3 EasyOPC.config

```

<Printer.EasyOpcClient
    ··endPointUrl="opc.tcp://10.0.50.180:4840/"
    ··sessionTimeouts="15"
    ··keepAliveInterval="10"
    ··sendKeepAliveSignal="false"
    ··keepAliveSignalAddress="nsu=http://www.boschrexroth.com/IndraLogic;s=Application.UI_SPS_GVL_b_Lebensbit_UI"/>

```

Figure 22: shows theEasy Opc Config

6 Calibration

Calibration is the process by which optimal parameters are set in the machine to achieve consistent high print quality. It is one of the important processes for preparing the machine for production print. The cartridges and printheads alignments are influencing the print quality, if the alignment is not set correctly the print quality will not be good. We will not get the high quality output, so in order to make a good print we should assure that all printheads and cartridges are aligned properly according to the chosen substrate, we can achieve good results by adjusting the position of the cartridges/Printheads and uniform density can be attained by adjusting the voltages of the printheads. There are three kinds of calibration like Printhead Calibration, Cartridge Calibration and Density Calibration. We will explore each calibration in individual sections.

7 Printhead Calibration

Printhead Calibration is the process of aligning the printheads in a straight line. Meaning each nozzle row in adjacent printheads are aligned to one another in a straight line. We need to position the printheads correctly by adjusting the stepper motors, this is a mechanical component for controlling printhead positions by adjusting in small steps. Each printhead is controlled by 2 stepper motors, one for controlling the Y position and another one to adjust the Phi position. X-axis Position of printhead will be controlled by ttp meteor command, before starting the print by giving the correct input.

Why do we need to calibrate Y?

We can adjust the Y direction by moving the printheads from left to right direction physically in clockwise or anti-clockwise direction, depending on the physical arrangement of printheads. If we don't calibrate Y, then the pixels are not aligned properly between successive printheads, so we may get misaligned print.

Why do we need to calibrate phi?

We can adjust the Phi direction by adjusting the angles of each printheads physically in clockwise or anti-clockwise direction depending on physical arrangement of printheads, by this way we make sure all printheads are aligned in the same angle, so that we are getting all pixels in the same row in a single straight line.

What happens if I'm totally wrong with my X offset?

If we are not aligning the x-direction correctly, we may get a stair print. Because each printhead has some value to start the print, if X-offset is not configured correctly then we will get an invalid print.

7.1 Printhead Calibration View for Developers

The below diagram explains how the process of printhead calibration is implemented in the software, what objects are interacting in the process.

- ViewModel[Execute the action trigger by respective ViewModel]
- IPrintheadCalibrationClient [Request/Response the information from the wcf service]
- DuplexClient [Generic class for handling the transaction between the client and wcf service]
- IPrintheadCalibrationWcfAdapter [WCF service for offering the printhead calibration via stepper motor]
- AsynchronousOperationExecutor[Execute the action asynchronously over the wcf]
- IPrintheadCalibrationService [Application service for talking with domain to calibrate the printheads]
- PrintheadDto [Data transfer object for single printhead in-order to pass the object over the wcf]
- ColorEngines [Manage all the color engine properties]
- Pcc [Controlling the single PCC information]
- PrintheadStepper [Holding the current Y and Phi position of stepper and do the movements]
- IExecutionEngine [Execute the action]
- IPrintAgentClient [Request/Response the information from PrintAgent wcf service]
- IPrintheadStepperService [Offer to manage the stepper movements]
- ITtpPrintEngine [Send/Receive the signal from ttp meteor].
- PrintEngineController [Controlling the color engines, print agents and so on]
- PccIdXOffset [Wrapper for holding the XOffset values of each Pcc].

The below diagram explains how the process Enable/Disable the stepper power is implemented in the software, what objects are interacting in the process.

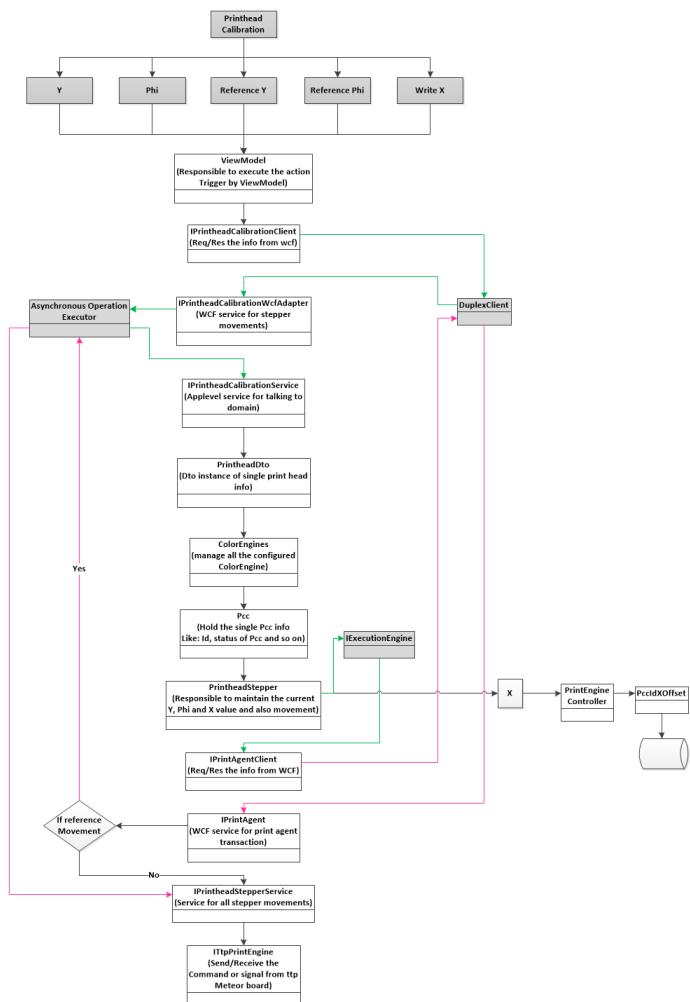


Figure 23: shows the Printhead Calibration Object Flow

- ViewModel[Execute the action trigger by respective ViewModel]
- IPrintEnginePowerSupplyClient [Interface to request or response from the WCF service]
- DuplexClient [Generic class for handling the transaction between the client and wcf service]
- IPrintEnginePowerSupplyWcfAdapter [WCF service for handling the power supply]
- AsynchronousOperationExecutor[Execute the action asynchronously over the wcf]
- IPrintEnginePowerSupplyService [Application service for handling the power supply]
- IExecutionEngine [Execute the action]

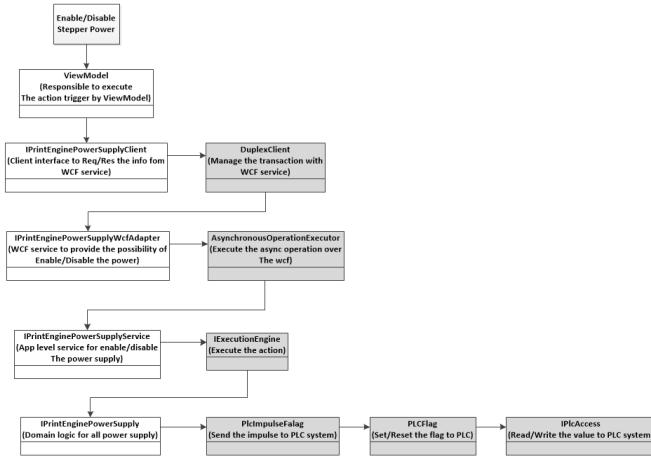


Figure 24: Shows the enable and disable stepper power

- IPrintEnginePowerSupply [Domain logic for power supply]
- PlcImpulseFlag [Send the Impulse to PLC system]
- PlcFlag [Set/reset the flag in PLC]
- IPlcAccess [Read/Write the value in PLC system].

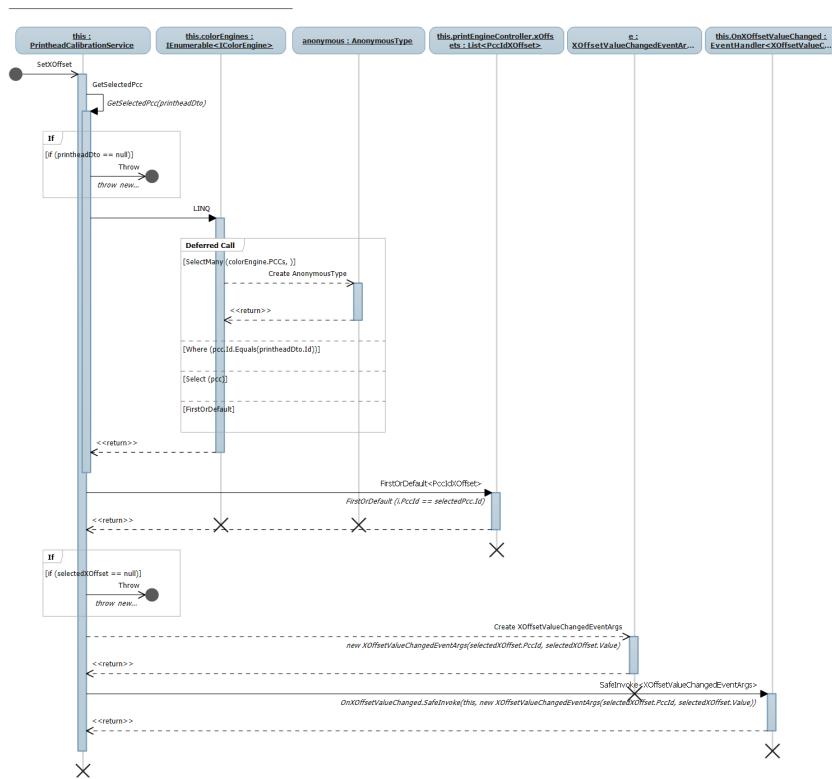


Figure 25: Shows the SetXOffset function flow

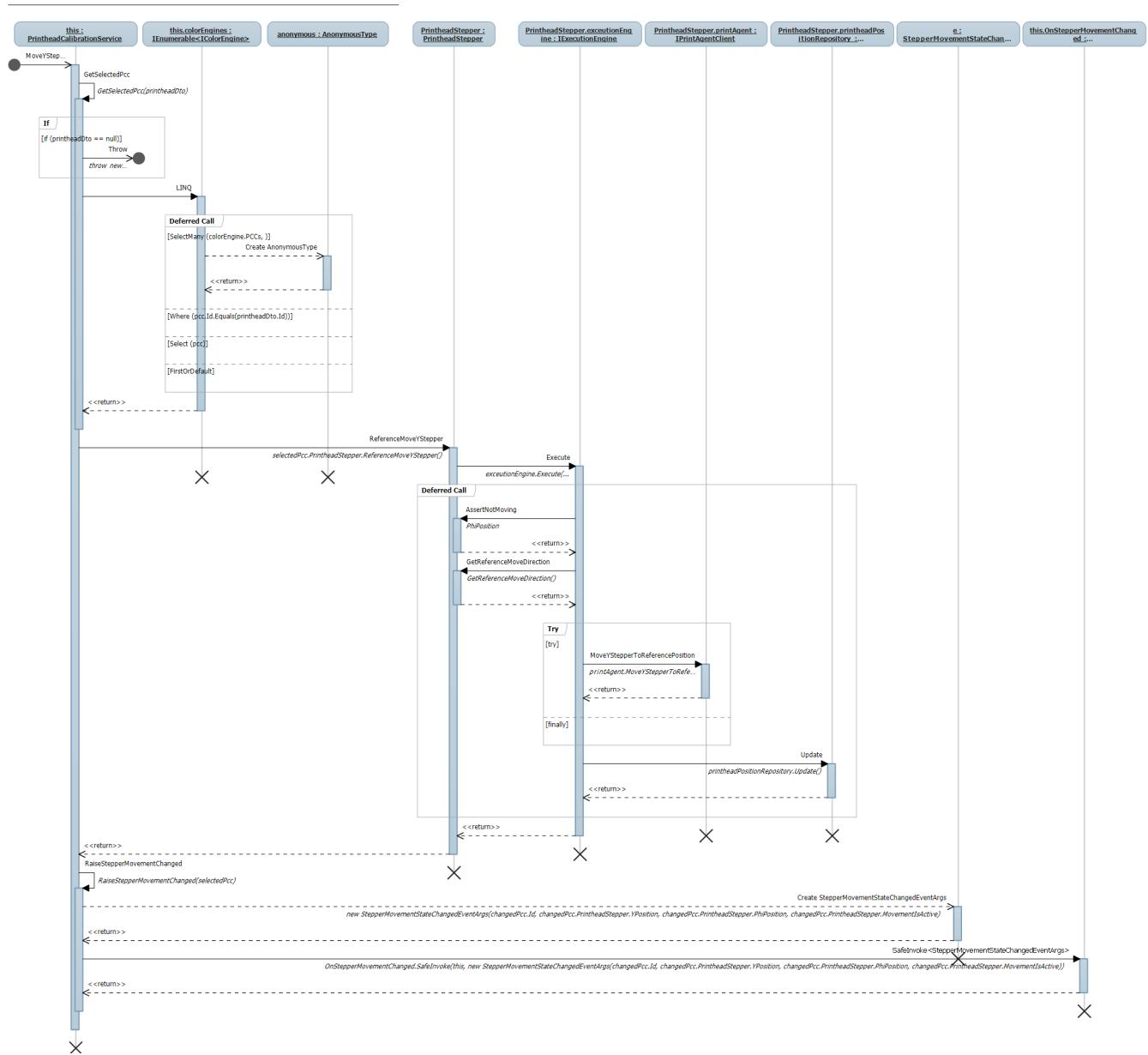


Figure 26: Shows the Move YStepper To ReferencePosition function flow

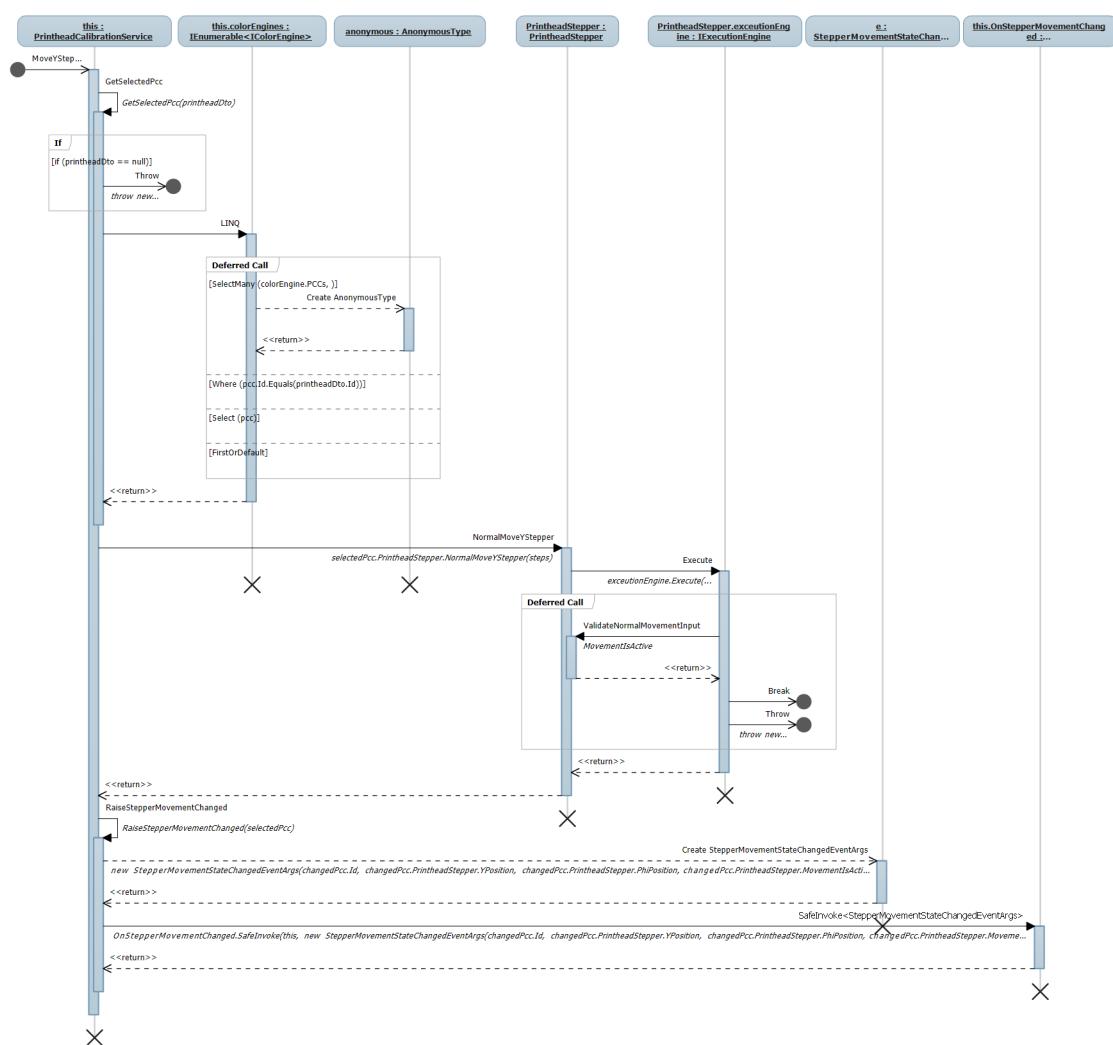


Figure 27: Shows the Move Y Stepper funtion flow

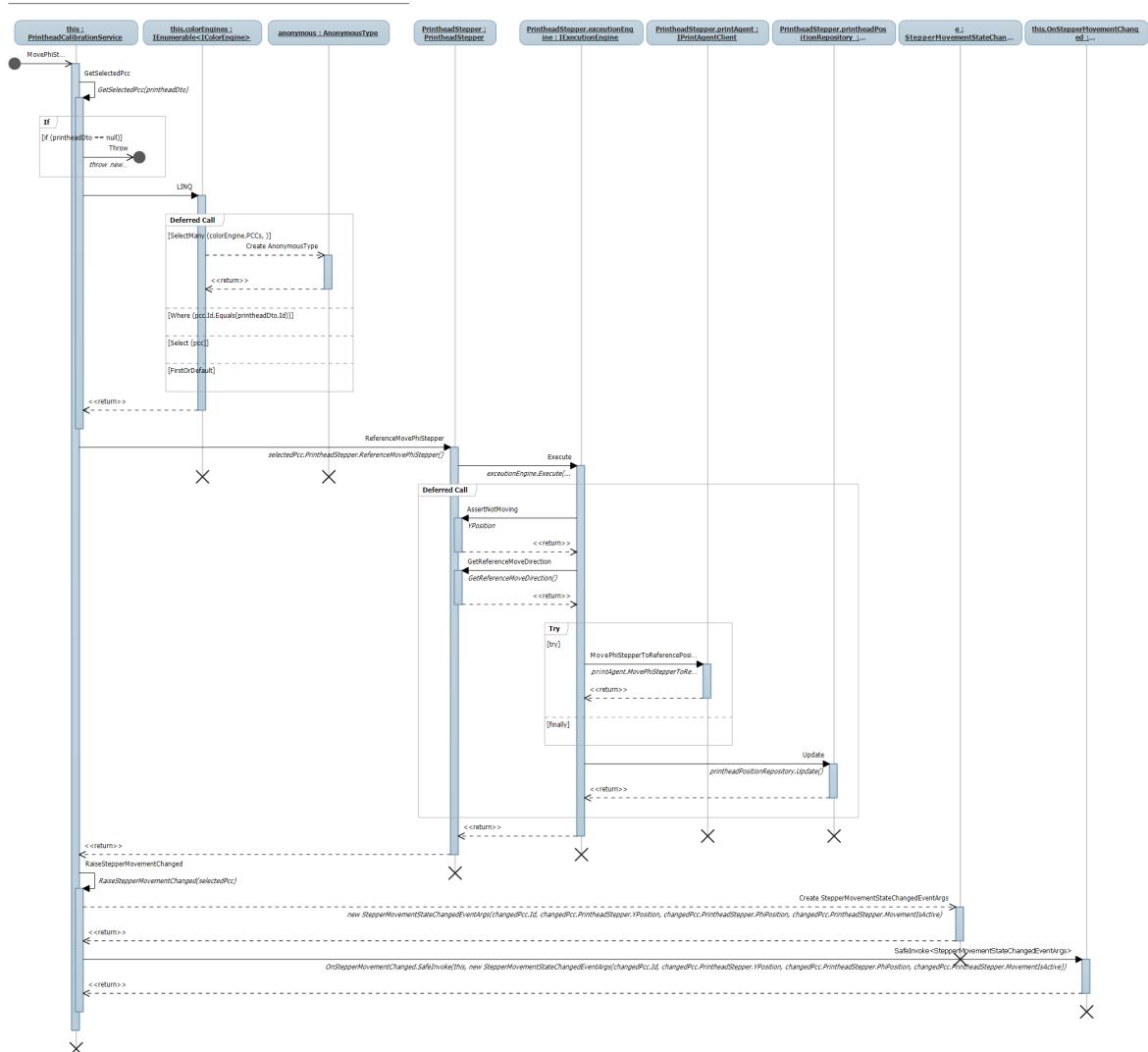


Figure 28: Shows theunction `MovePhiStepperToReferencePosition` function flow.

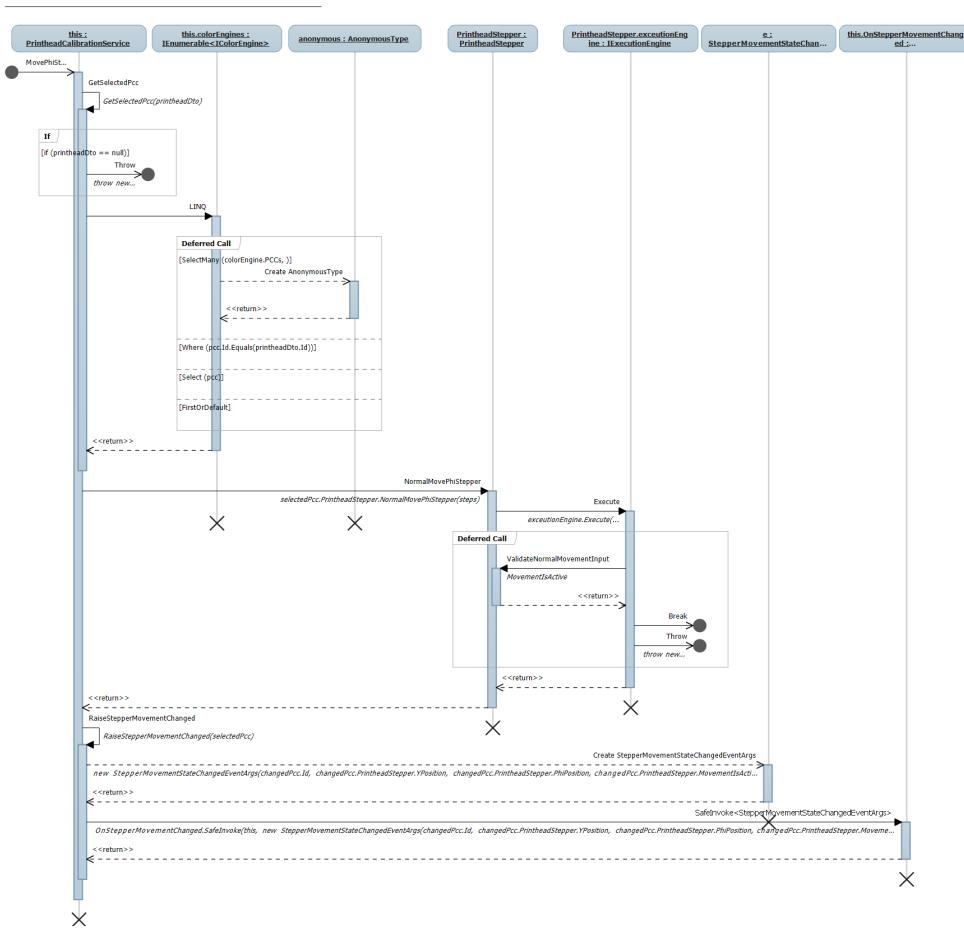


Figure 29: Shows the MovePhiStepper function flow

7.2 Printhead Calibration View for Experts

Printhead calibration is only allowed for a expert level user. An expert user first makes a test pattern print and verify if all the alignments are fine. If the alignments are fine. Then he leaves the settings as it is. If the settings are not okay. he can modify the X or Y or Phi positions accordingly. He can repeat the process until the expected level of alignment is achieved.

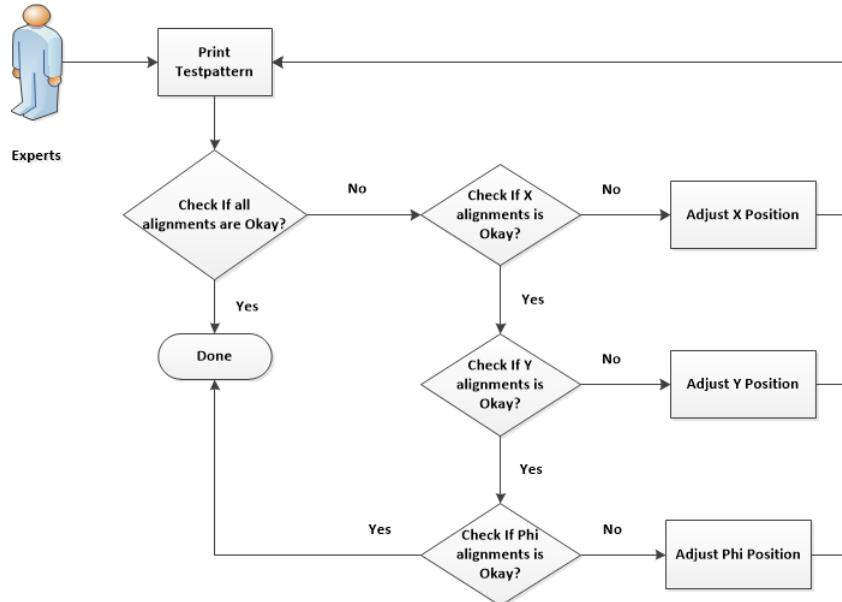


Figure 30: Shows the Printhead Calibration expert user workflow

7.3 Printhead Calibration View for Service Technician

The service technician needs to check the following configuration in order to perform the printhead calibration.

- Printer.Config
- PrintUnit.Config
- App.Config

7.3.1 Printer.Config for Printhead Calibration

The service technician needs to check if the printer config is configured correctly.



```
<colors>
<color id="Cyan" argb="#8800FFFF">
</color>

<color id="Red" argb="#FF0000">
</color>

<color id="Yellow" argb="#88FFFF00">
</color>

<color id="Black" argb="#88000000">
</color>
```

Figure 31: shows the Printer Config

7.3.2 PrintUnitConfig.Config for Printhead Calibration

The service technician needs to check if the PrintUnit config is configured correctly.

```

<printNodes>
  <printNode
    colorId="Cyan"
    connectedPrintheads="20"
    daisyChainStart="1"
    ipAddress="10.0.50.166"
    macAddress="74-D4-35-16-54-8E"
    maintenanceServiceConfiguration="MaintainanceService2"
    meteorConfiguration="\\"10.0.50.166\MeteorConfig\meteor.cfg"
    printAgentConfiguration="PrintAgentNode1"
    printdataFolder="\\"10.0.50.166\Transfer"
    deltaForUpperWarning="1.2"
    deltaForLowerWarning="2"/>

  <printNode
    colorId="Red"
    connectedPrintheads="20"
    daisyChainStart="1"
    ipAddress="10.0.50.167"
    macAddress="74-D4-35-16-C5-32"
    maintenanceServiceConfiguration="MaintainanceService2"
    meteorConfiguration="\\"10.0.50.167\MeteorConfig\meteor.cfg"
    printAgentConfiguration="PrintAgentNode2"
    printdataFolder="\\"10.0.50.167\Transfer"
    deltaForUpperWarning="1.2"
    deltaForLowerWarning="2"/>

  <printNode
    colorId="Yellow"
    connectedPrintheads="20"
    daisyChainStart="1"
    ipAddress="10.0.50.168"
    macAddress="74-D4-35-1E-DC-31"
    maintenanceServiceConfiguration="MaintainanceService2"
    meteorConfiguration="\\"10.0.50.168\MeteorConfig\meteor.cfg"
    printAgentConfiguration="PrintAgentNode3"
    printdataFolder="\\"10.0.50.168\Transfer"
    deltaForUpperWarning="1.2"
    deltaForLowerWarning="2"/>

  <printNode
    colorId="Black"
    connectedPrintheads="20"
    daisyChainStart="1"
    ipAddress="10.0.50.169"
    macAddress="74-D4-35-1F-5D-58"
    maintenanceServiceConfiguration="MaintainanceService2"
    meteorConfiguration="\\"10.0.50.169\MeteorConfig\meteor.cfg"
    printAgentConfiguration="PrintAgentNode4"
    printdataFolder="\\"10.0.50.169\Transfer"
    deltaForUpperWarning="1.2"
    deltaForLowerWarning="2"/>
</printNodes>

```

Figure 32: shows the Print.Unit Config

7.3.3 App.Config for Printhead Calibration

The service technician needs to check the App config is configured correctly.

```

<WcfConfiguration>
  <endPoints>
    < endPoint name="PrinterStatusClient" uri="net.tcp://localhost:8010/PrinterStatus" isDebugEnabled="true" />
    < endPoint name="PrinterStatusService" uri="net.tcp://localhost:8010/PrinterStatus" isDebugEnabled="true" />
    < endPoint name="PrinterConfigurationClient" uri="net.tcp://localhost:9001/PrinterConfigurationService" isDebugEnabled="true" />
    < endPoint name="PrinterConfigurationService" uri="net.tcp://localhost:9001/PrinterConfigurationService" isDebugEnabled="true" />
    < endPoint name="CleaningService" uri="net.tcp://localhost:9000/CleaningService" isDebugEnabled="true" />
    < endPoint name="CleaningClient" uri="net.tcp://localhost:9000/CleaningService" isDebugEnabled="true" />
    < endPoint name="UserManagementService" uri="net.tcp://localhost:9002/UserManagementService" isDebugEnabled="true" />
    < endPoint name="UserManagementClient" uri="net.tcp://localhost:9002/UserManagementService" isDebugEnabled="true" />
    < endPoint name="PrintheadCalibrationService" uri="net.tcp://localhost:9003/PrintheadCalibrationService" isDebugEnabled="true" />
    <EndPoint name="PrintheadCalibrationClient" uri="net.tcp://localhost:9003/PrintheadCalibrationService" isDebugEnabled="true" />
    < endPoint name="TestPatternService" uri="net.tcp://localhost:9004/TestPatternService" isDebugEnabled="true" />
    < endPoint name="TestPatternClient" uri="net.tcp://localhost:9004/TestPatternService" isDebugEnabled="true" />
    < endPoint name="PrintEnginePowerSupplyService" uri="net.tcp://localhost:9005/PrintEnginePowerSupplyService" isDebugEnabled="true" />
    < endPoint name="PrintEnginePowerSupplyClient" uri="net.tcp://localhost:9005/PrintEnginePowerSupplyService" isDebugEnabled="true" />
    < endPoint name="DensityCalibrationService" uri="net.tcp://localhost:9006/DensityCalibrationService" isDebugEnabled="true" />
    < endPoint name="DensityCalibrationClient" uri="net.tcp://localhost:9006/DensityCalibrationService" isDebugEnabled="true" />
    < endPoint name="CartridgeCalibrationService" uri="net.tcp://localhost:9007/CartridgeCalibrationService" isDebugEnabled="true" />
    < endPoint name="CartridgeCalibrationClient" uri="net.tcp://localhost:9007/CartridgeCalibrationService" isDebugEnabled="true" />

    < endPoint name="PrintAgentNode1" uri="net.tcp://10.0.50.166:8083/PrintAgent" isDebugEnabled="true" />
    < endPoint name="PrintAgentNode2" uri="net.tcp://10.0.50.167:8083/PrintAgent" isDebugEnabled="true" />
    < endPoint name="PrintAgentNode3" uri="net.tcp://10.0.50.168:8083/PrintAgent" isDebugEnabled="true" />
    < endPoint name="PrintAgentNode4" uri="net.tcp://10.0.50.169:8083/PrintAgent" isDebugEnabled="true" />

    < endPoint name="PrinterPositioningService" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true" />
    < endPoint name="PrinterPositioningClient" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true" />
    < endPoint name="CurrentOrderService" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true" />
    < endPoint name="CurrentOrderClient" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true" />

    < endPoint name="ComputedDensityCalibrationService" uri="net.tcp://localhost:9010/ComputedDensityCalibrationService" isDebugEnabled="true" />
    < endPoint name="ComputedDensityCalibrationClient" uri="net.tcp://localhost:9010/ComputedDensityCalibrationService" isDebugEnabled="true" />

    < endPoint name="DensityProfileService" uri="net.tcp://localhost:9011/DensityProfileService" isDebugEnabled="true" />
    < endPoint name="DensityProfileClient" uri="net.tcp://localhost:9011/DensityProfileService" isDebugEnabled="true" />

    < endPoint name="UserTransactionService" uri="net.tcp://localhost:9012/UserTransactionService" isDebugEnabled="true" />
    < endPoint name="UserTransactionClient" uri="net.tcp://localhost:9012/UserTransactionService" isDebugEnabled="true" />
  
```

Figure 33: shows the App Config

8 Cartridge Calibration

Cartridge calibration is the process of controlling X and Y position by adjusting the MT130 table of each cartridge, so that we make sure all the available cartridges are aligned properly. The service technician could check that by printing the test pattern before going to the production print, If the X and Y position are not configured correctly, we may not get a valid print. One who calibrates can adjust X and Y until he get a proper alignment with a valid print.

8.1 Cartridge Calibration View for Developers

The below diagram explains how the process Cartridge Calibration is implemented in the software, what objects are interacting in the process. The following objects are interacting in this process.

- ICartridgeCalibrationClient [The client interface for Request/Response the service from WCF]
- ICartrdigeCalibrationWcfAdapter [WCF service offers to calibrate the cartridges]
- AsynchronousOperationExecutor[Execute the action asynchronously over the wcf]
- DuplexClient [Generic class for handling the transaction between the client and wcf service]

- ICartridgeCalibrationService [The application level service for cartridge calibration]
- InkSystem [Manage all the color system properties]
- ColorSystem [Manage the single color system properties]
- MT130Axis [Responsible to positioning the cartridge towards X or Y direction]
- IExecutionEngine [Execute the action]
- PlcParameter [Used to write the parameter to PLC system]
- PlcFunction [Execute the PLC function]
- PlcFlag [Set/Reset the flag to PLC system]
- PlcImpulse [Send the impulse flag to PLC]
- IPlcAccess [Read/Write the values to the PLC system]

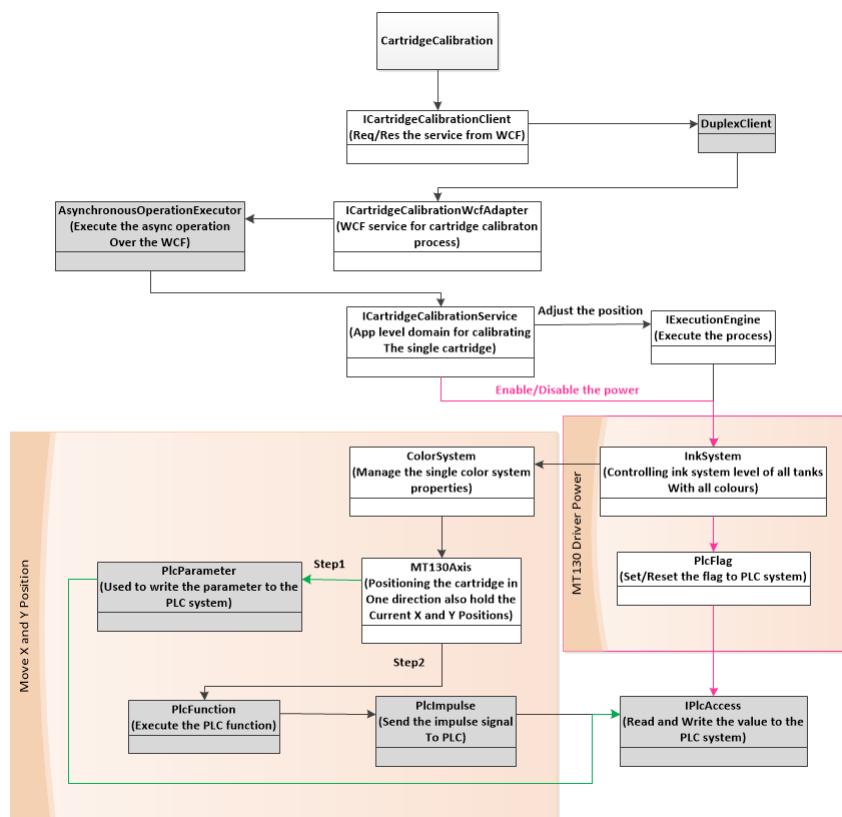


Figure 34: shows the Cartridge Calibration Overview

8.2 Cartridge Calibration View for Experts

Cartridge calibration is only allowed for a expert/service level user. The technician who performs the cartridge calibration can first makes a test pattern print and verify if X and Y alignments of cartridges are fine. If the alignments are fine, then leave the setting as it is. If the settings are not okay. he can modify the X or Y positions accordingly. The Technician can repeat the process until the expected level of Cartridge alignment is achieved.

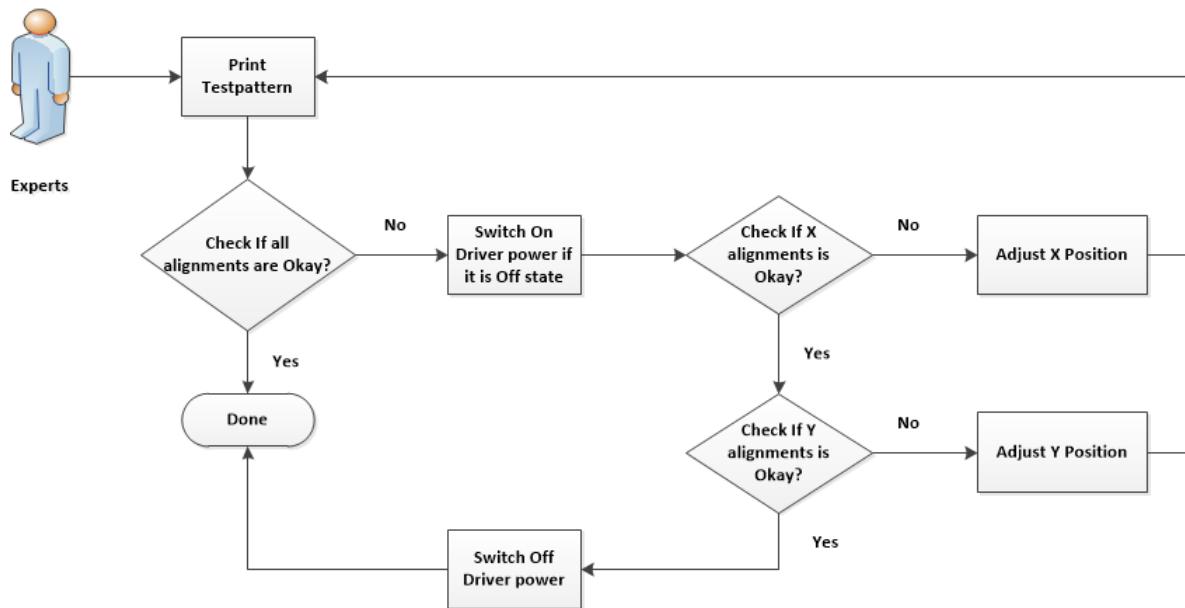


Figure 35: shows the cartridge calibration view

8.3 Cartridge Calibration View for Service Technician

The service technician needs to check the following configuration in order to do the Cartridge calibration.

- Printer.Config
- App.Config

8.3.1 Printer.Config for Cartridge Calibration

The service technician needs to check if the Printer config is configured correctly.

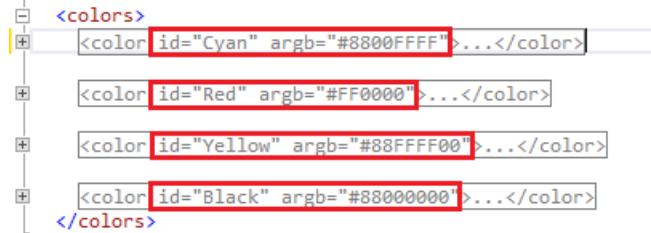


Figure 36: shows the Cartridge Calibration PrinterConfig

8.3.2 App.Config for Cartridge Calibration

The service technician needs to check if the App config is configured correctly.



Figure 37: shows the Cartridge Calibration App Config

9 Density Calibration

Density calibration is the process by which a technician who performs the calibration make sure the output print has uniform density distribution. The technician first checks if the color density measured from Test pattern print is uniform. If the density is nonuniform, he can use the density calibration procedure to perform the calibration. The below figure shows Density deviation in the output print.

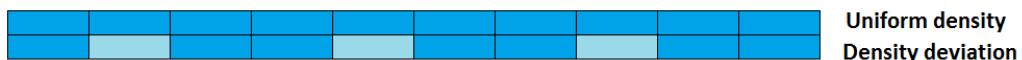


Figure 38: shows the DensityDeviation

Density can be adjusted by fine-tuning printhead voltages. By default, printheads delivered from the manufacturers are pre-calibrated with predefined voltages. In 2250

machine Kyocera KJ4B-S Series 1200dpi printhead are used, Each printhead has 2 regions and each one control 5 units by respective Ic's, so we can control each ic's by using TTP meteor API. The below diagram explains the arrangement of printhead.



Figure 39: shows the unit arrangement in KJ4B-S Series 1200dpi printhead

9.1 Density Cartridge Calibration View for Developers

Looking from the developers perspective, the below diagram explains density calibration create new profile overview

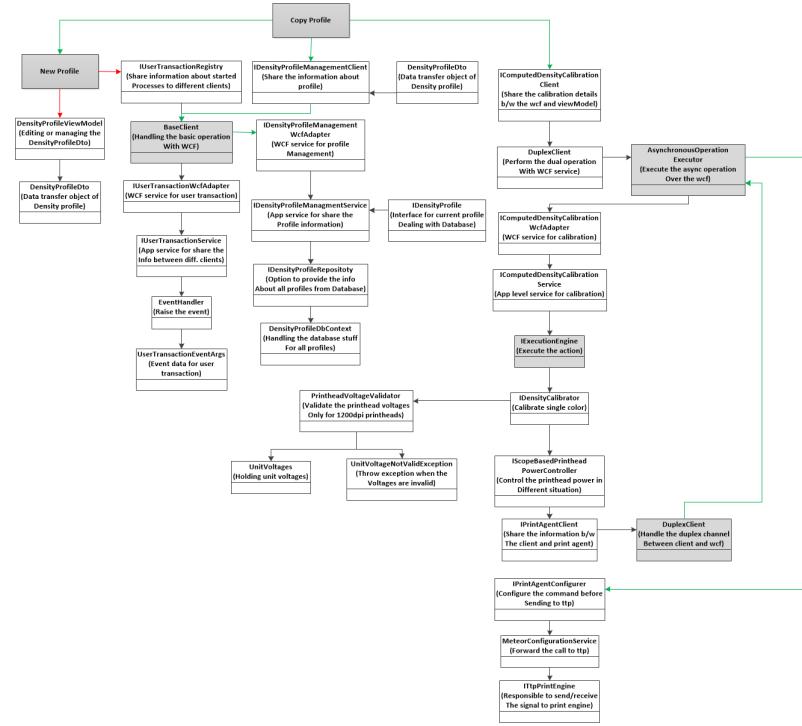


Figure 40: shows the density calibration create new profile

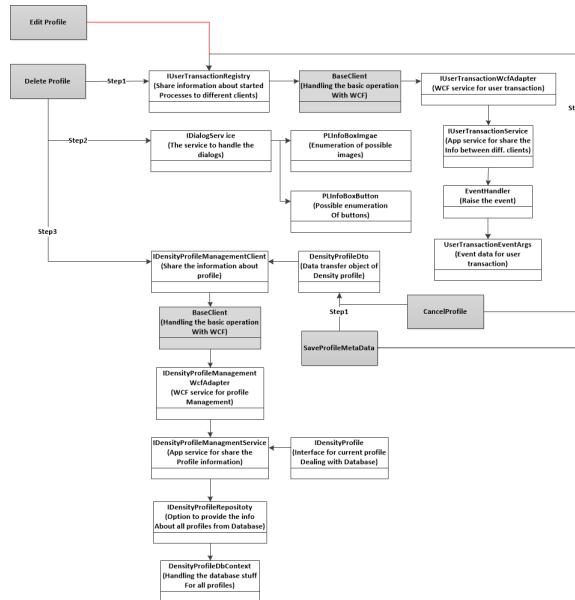


Figure 41: shows the density calibration edit profile

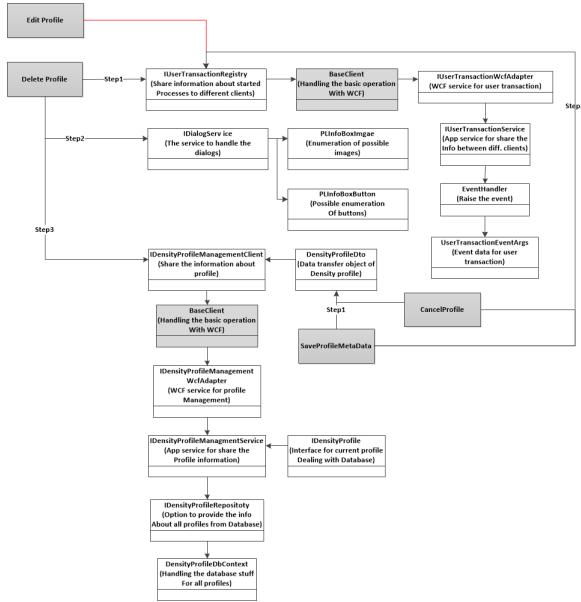


Figure 42: shows the density calibration delete profile

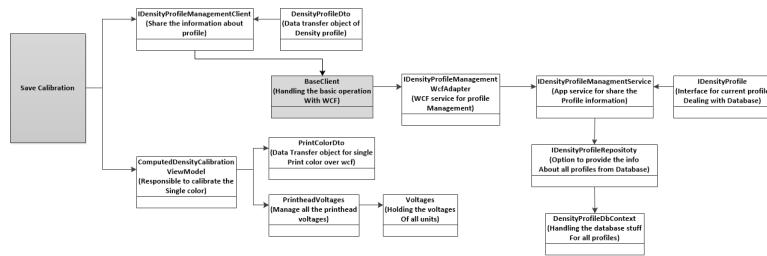


Figure 43: shows the density calibration save profile

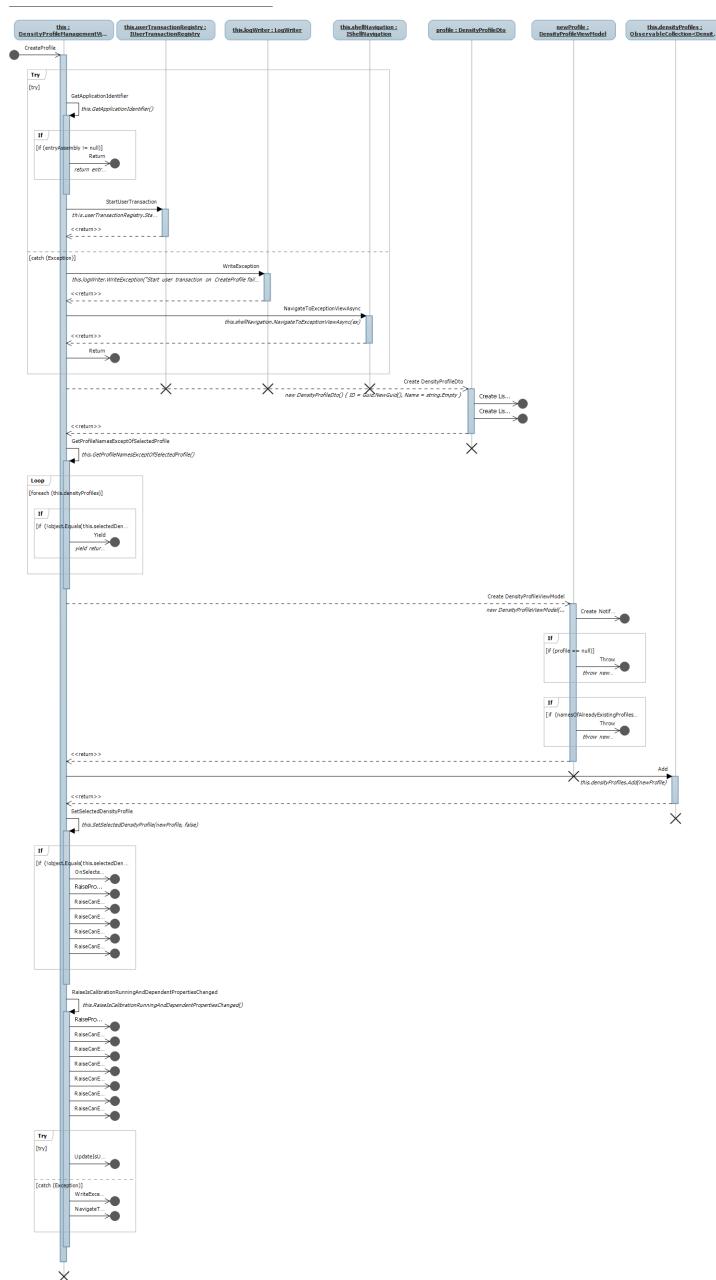


Figure 44: shows the density calibration create profile

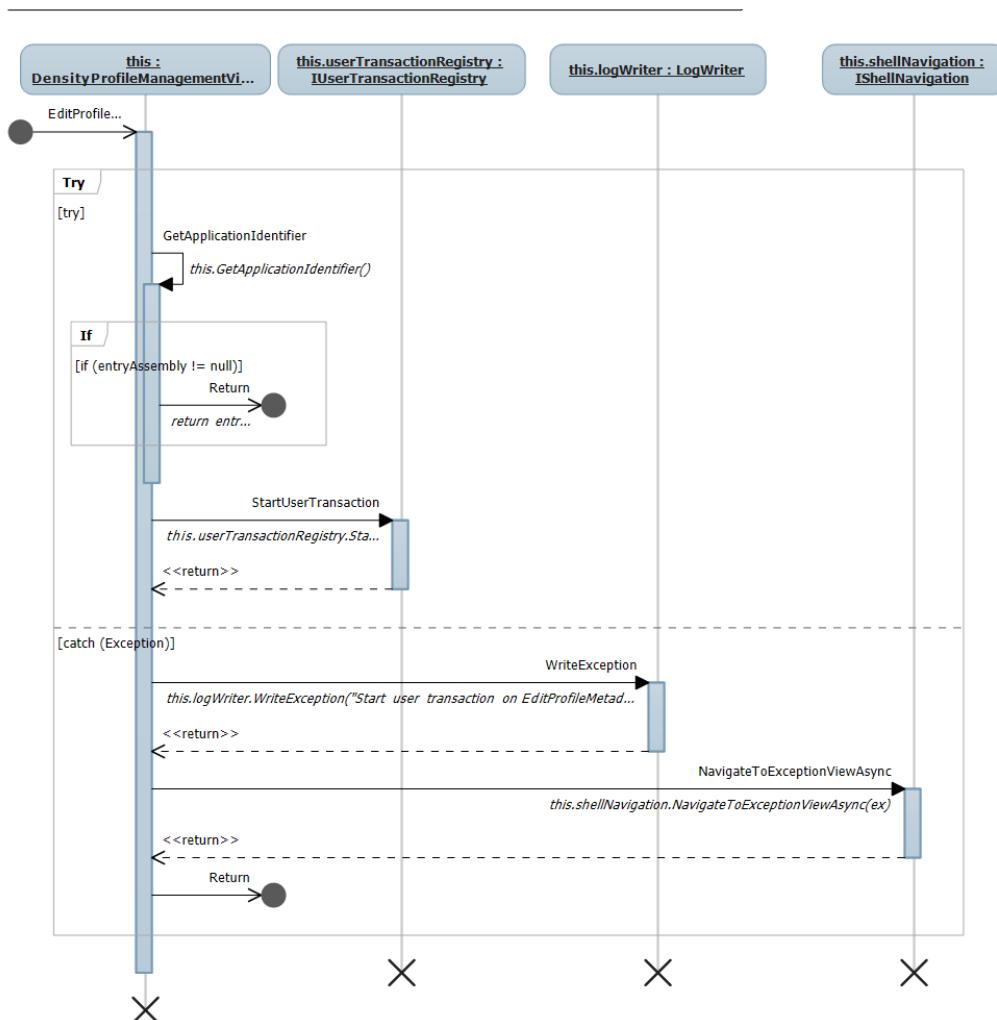


Figure 45: shows the density calibration edit profile

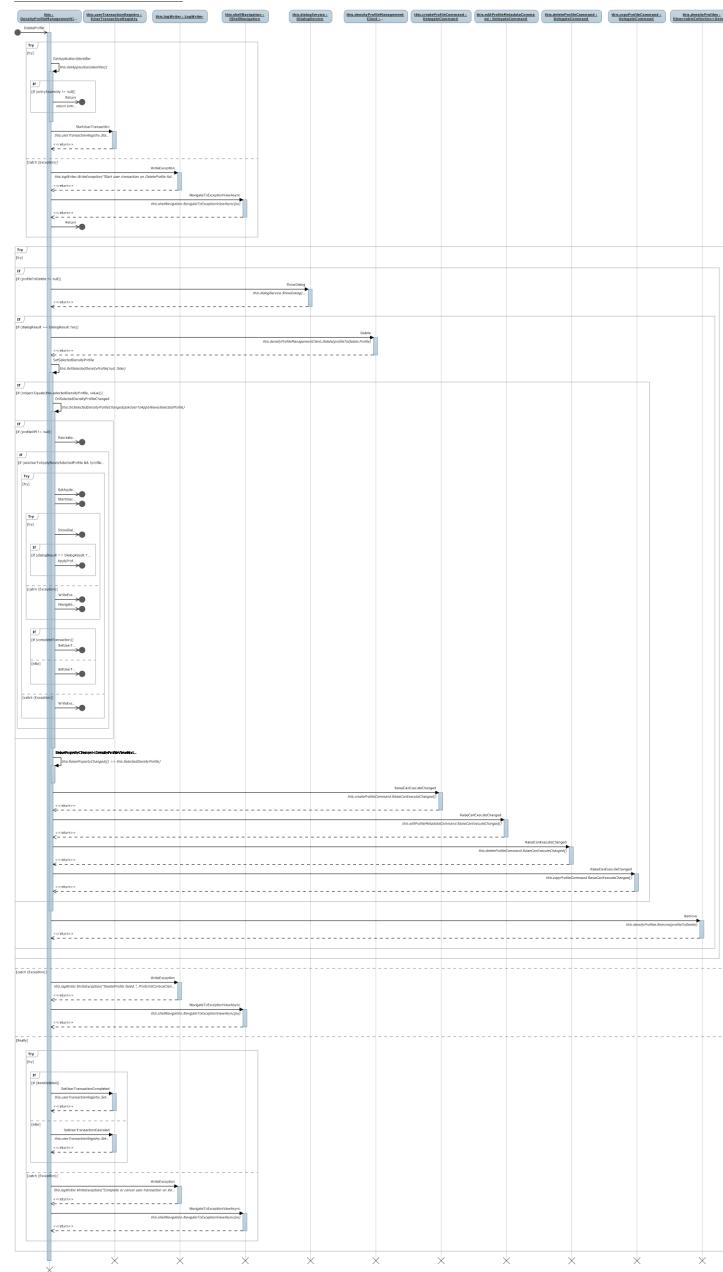


Figure 46: shows the density calibration delete profile

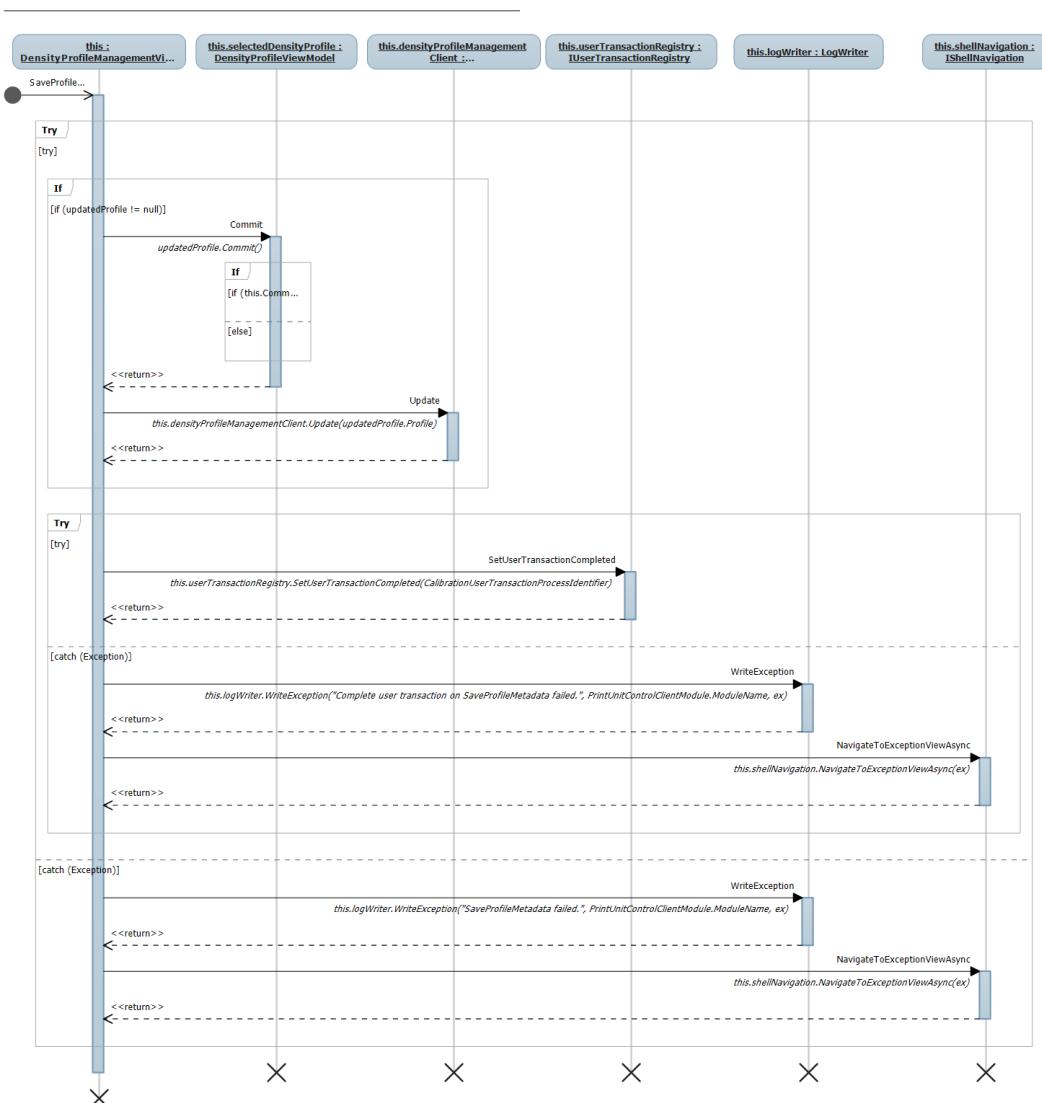


Figure 47: shows the density calibration save profile

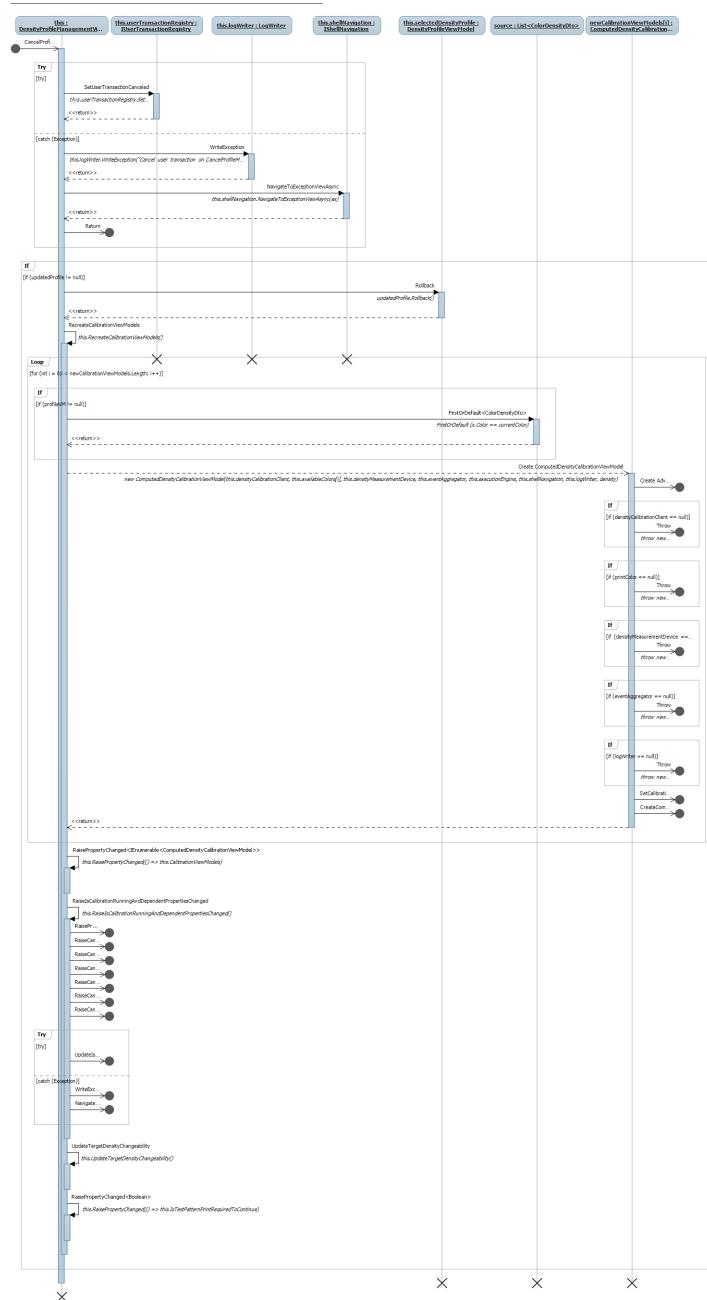


Figure 48: shows the density calibration cancel metadata profile editing

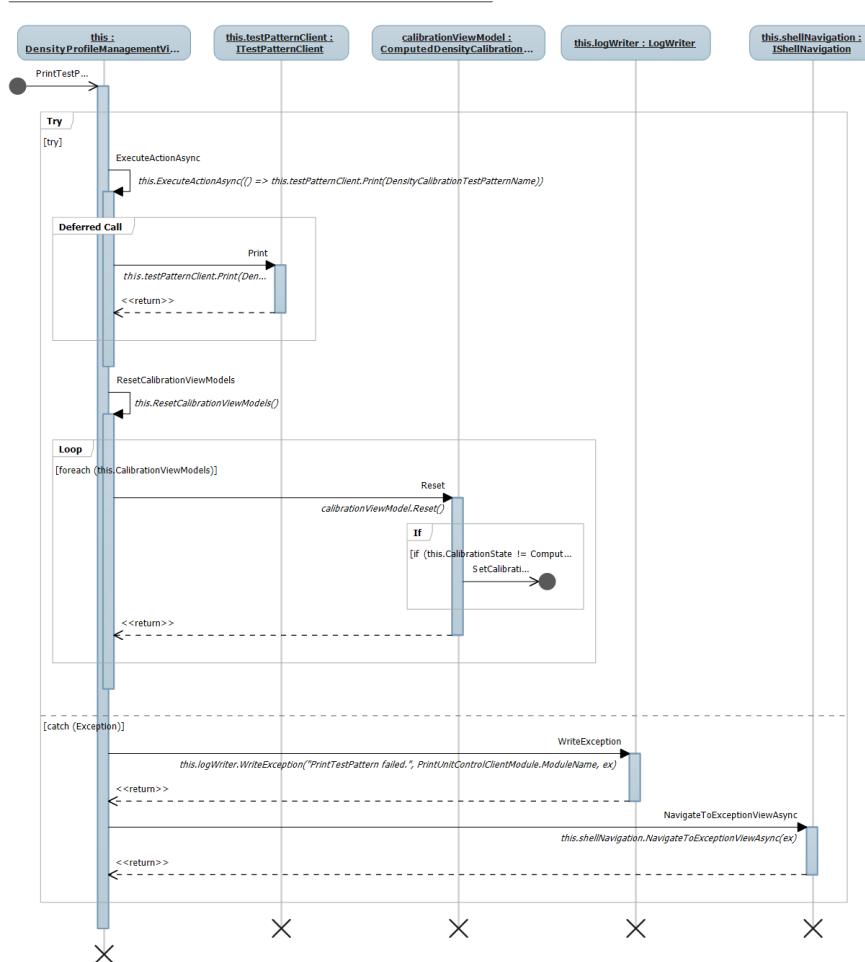


Figure 49: shows the density calibration print Test Pattern

9.2 Density Cartridge Calibration View for Operators

Density calibration is only allowed for a expert/service level user. The technician who performs the Density calibration can first makes a test pattern print and verify if Density distribution are fine and even. If the Density distribution are fine, then use the currently loaded profile. If the Density are not okay. he can start density calibration. The Technician can repeat the process until the expected level of density is reached.

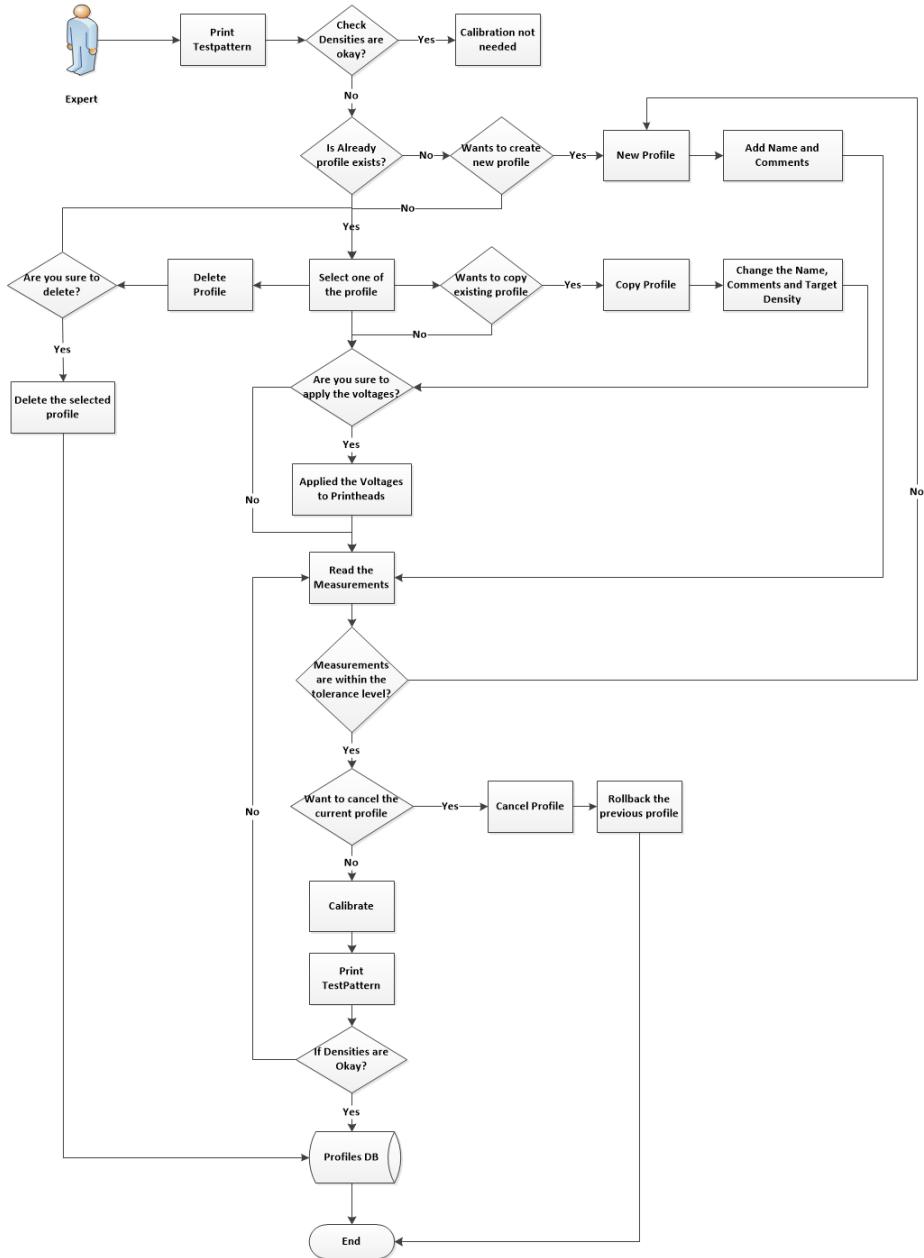


Figure 50: shows the density calibration for expert user

9.3 Density Cartridge Calibration View for Service Technician

The service technician needs to check the following configuration in order to do the cartridge calibration.

- App.Config

- Printer.Config
- ConnectionString.Config

9.3.1 App.Config for Density Calibration

The service technician needs to check if the App config is configured correctly.

```

<endPoint name="PrintAgentNode1" uri="net.tcp://10.0.50.166:8083/PrintAgent" isDebugEnabled="true" />
<endPoint name="PrintAgentNode2" uri="net.tcp://10.0.50.167:8083/PrintAgent" isDebugEnabled="true" />
<endPoint name="PrintAgentNode3" uri="net.tcp://10.0.50.168:8083/PrintAgent" isDebugEnabled="true" />
<endPoint name="PrintAgentNode4" uri="net.tcp://10.0.50.169:8083/PrintAgent" isDebugEnabled="true" />

<endPoint name="PrinterPositioningService" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true" />
<endPoint name="PrinterPositioningClient" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true" />
<endPoint name="CurrentOrderService" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true" />
<endPoint name="CurrentOrderClient" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true" />

<endPoint name="ComputedDensityCalibrationService" uri="net.tcp://localhost:9010/ComputedDensityCalibrationService" isDebugEnabled="true" />
<endPoint name="ComputedDensityCalibrationClient" uri="net.tcp://localhost:9010/ComputedDensityCalibrationService" isDebugEnabled="true" />

<endPoint name="DensityProfileService" uri="net.tcp://localhost:9011/DensityProfileService" isDebugEnabled="true" />
<endPoint name="DensityProfileClient" uri="net.tcp://localhost:9011/DensityProfileService" isDebugEnabled="true" />

<endPoint name="UserTransactionService" uri="net.tcp://localhost:9012/UserTransactionService" isDebugEnabled="true" />
<endPoint name="UserTransactionClient" uri="net.tcp://localhost:9012/UserTransactionService" isDebugEnabled="true" />
```

Figure 51: shows the density calibration App.Config

9.3.2 Printer.Config for Density Calibration

The service technician needs to check if the Printer config is configured correctly.



```

<colors>
  <color id="Cyan" argb="#8800FFFF">...</color>
  <color id="Red" argb="#FF0000">...</color>
  <color id="Yellow" argb="#88FFFFFF">...</color>
  <color id="Black" argb="#88000000">...</color>
</colors>
```

Figure 52: shows the density calibration Printer.Config

9.3.3 ConnectionString.Config for Density Calibration

The service technician needs to check if the ConnectionString config is configured correctly.

```

<?xml version="1.0" encoding="utf-8" ?>
<connectionStrings>
  <add
    name="MachineContext"
    providerName="System.Data.SqlClient"
    connectionString="
      Server=.\SQLEXPRESS;
      Database=palistestNewDEV;
      integrated Security=true;
      MultipleActiveResultSets=true;"/>
</connectionStrings>

```

Figure 53: shows the density calibration ConnectionStringConfig.Config

10 Standby

Standby is a process which helps to maintain the print quality, while the machine is not in printing mode after a long print, the machine can be in idle state for short or long duration. In such instances we need to make sure all nozzles are maintained as before, without this standby mechanism we may waste some inks and papers. One may need to perform cleaning and print test pattern in order to get the correct print quality. When the machine is waking up from idle to printing state, it will take some time to get a valid print. Due to the fact inks have the possibility to dry up. At times it may block some nozzles, so we are in a situation to clean the printheads to bring the nozzles back. To avoid such problems, we hold the machine in Production Standby or Standby depending on the situation using the following strategies.

1. Spit : It is sub process which activate nozzles at specific time interval and eject ink from nozzles.
2. Meniscus Activation : It is sub process which vibrate the ink inside printheads at a certain frequency but does not eject inks from the printheads.

We can put the machine in two different modes like Production Standby, Standby.

- Production Standby : Production Standby is one of the standby mode, Spit or Meniscus activation could be used in that mode. Use this option when the machine needs to be in an idle state for a short time after the production print to make the maximum nozzle availability.
- Standby : Standby is a mode where, we can put the machine in idle state for longer period. Either spit or meniscus activation strategy could be configured to make the maximum nozzle availability.
- None : Deactivate the production or standby option, if we go with this option we may needs to clean the printheads when we have a break between the jobs.

Either one of the option is selected based on the operator choice. Normally in production standby the spit duration and spit interval will be shorter then the Standby mode.

10.1 Standby view for Developers

Looking from the developers perspective, the below diagram explains overall standby process object flow.

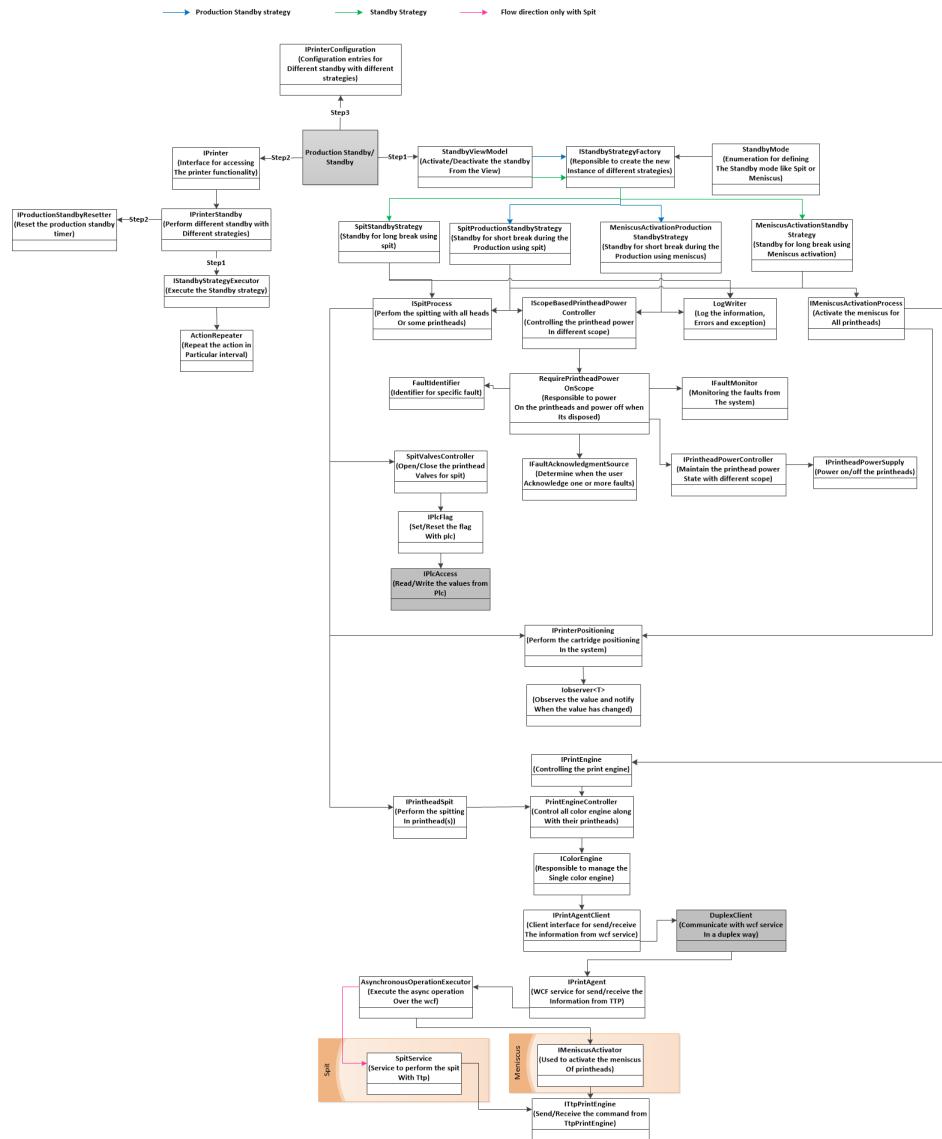


Figure 54: shows the overall standby object mapping

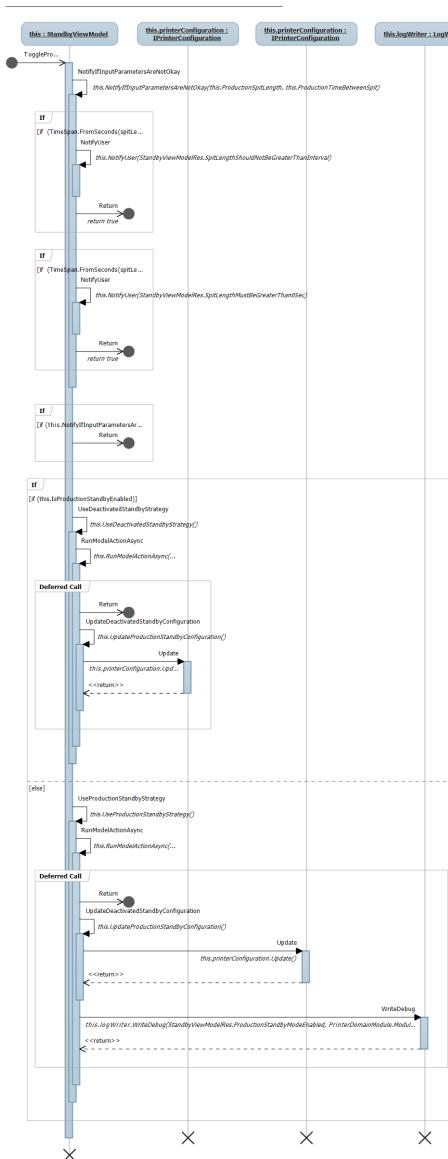


Figure 55: shows the production standby funtion flow diagram

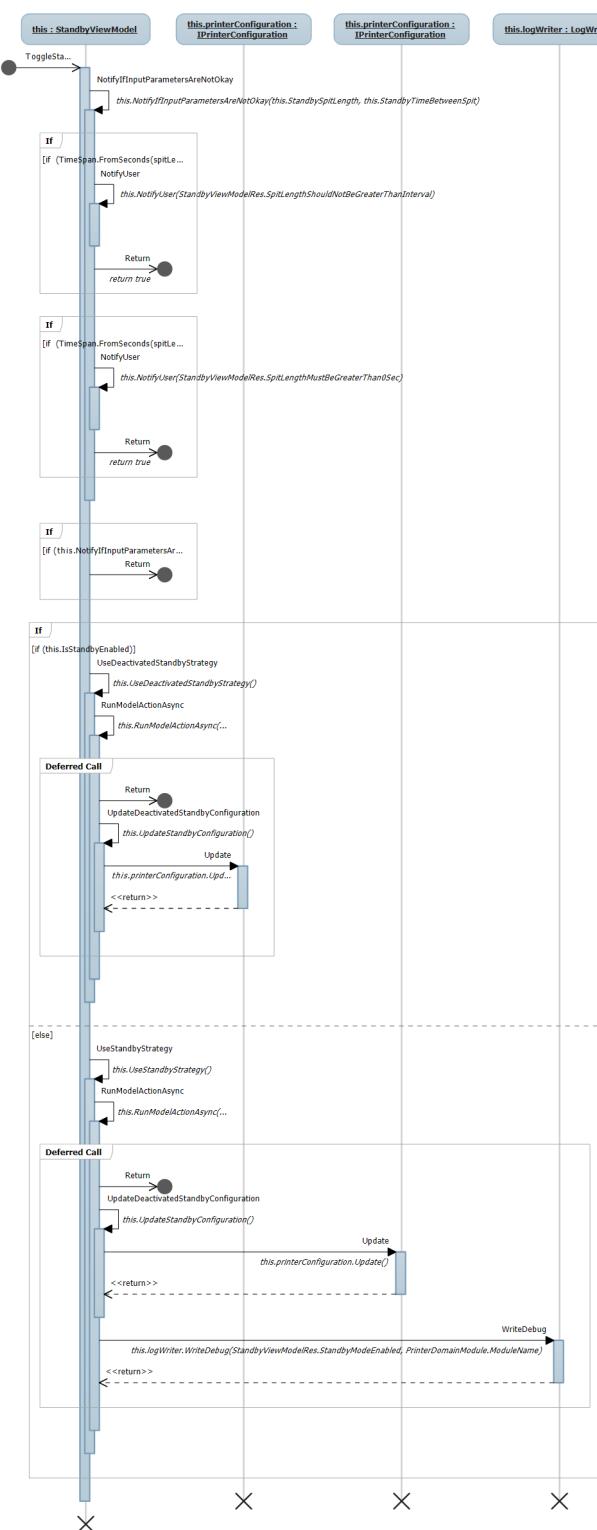


Figure 56: shows the standby function flow diagram

10.2 Standby view for Operators

The operator while switching between printing production jobs, he may need to leave the machine idle for a short or long time, depending on the time interval between printing successive jobs. He can choose the standby mode accordingly and leave the machine in Production Standby or Standby mode. The operator can choose standby for the whole night 'Standby mode'. The below diagram explains the process.

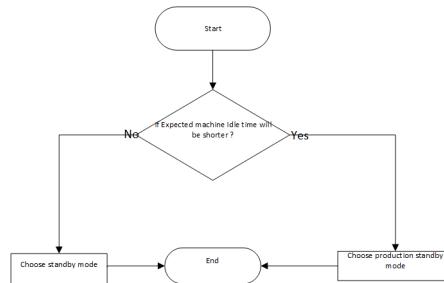


Figure 57: shows the operator flow diagram

10.3 Standby view for Service Technician

The service technician needs to check the following configuration in order to put the machine in standby mode.

- Printer.Config

The following section in Printer.config has to be checked

```

<xrxml version="1.0" encoding="utf-8" />
<Printer>
  <printingDirection>Width</printingDirection>
  <machineIdentifier>Palis750</machineIdentifier>
  <isScanPrinter>false</isScanPrinter>
  <!--currentStandbyMode possible values => Standby, Production or None-->
  <!--currentStandbyTimeout: 0 disables the timer-->
  <standby>
    <currentStandbyMode>Standby</currentStandbyMode>
    <!--Strategy types possible values => Spilt, MeniscusActivation-->
    <!--standbyStrategy type="Spilt" executionDuration="00:00:01" maintenanceInterval="01:00:00"-->
    <!--productionStandbyStrategy type="Spilt" executionDuration="00:00:01" maintenanceInterval="01:00:00"-->
  </standby>
  <!--<colors>
    <!--<color id="Cyan" argb="#8800FFFF">
      <!--<printheads>
        <!--<printhead id="1" displayX="0" displayY="45" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="2" displayX="50" displayY="0" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="3" displayX="105" displayY="45" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="4" displayX="160" displayY="0" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="5" displayX="215" displayY="45" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="6" displayX="270" displayY="0" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="7" displayX="325" displayY="45" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="8" displayX="380" displayY="0" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="9" displayX="435" displayY="45" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="10" displayX="490" displayY="0" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="11" displayX="545" displayY="45" displayWidth="65" displayHeight="40"/>
        <!--<printhead id="12" displayX="600" displayY="0" displayWidth="65" displayHeight="40"/>
      </printheads>
    </color>
  </colors>
  <!--<printheads>
    <!--<printhead id="1" displayX="0" displayY="45" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="2" displayX="50" displayY="0" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="3" displayX="105" displayY="45" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="4" displayX="160" displayY="0" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="5" displayX="215" displayY="45" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="6" displayX="270" displayY="0" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="7" displayX="325" displayY="45" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="8" displayX="380" displayY="0" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="9" displayX="435" displayY="45" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="10" displayX="490" displayY="0" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="11" displayX="545" displayY="45" displayWidth="65" displayHeight="40"/>
    <!--<printhead id="12" displayX="600" displayY="0" displayWidth="65" displayHeight="40"/>
  </printheads>
  </Printer>

```

Figure 58: shows the Printer.config for standby operator

11 Printing

Printing is the main process in a digital printer. An operator can print available job with chosen parameters according to the customer requirements. Before going to the mass production print, the operator has to check the print quality, nozzle availability by printing the test pattern. The test pattern is usually printed with preconfigured job parameters. Once we get the valid result from the test pattern, one can start the production. There is also an option to stop the print in between by our software or emergency stop from the machine if something goes wrong in the process. Note: Test pattern is not necessary for every production print, it's required only during calibration or proofing/sampling.

11.1 Printing View for Developers

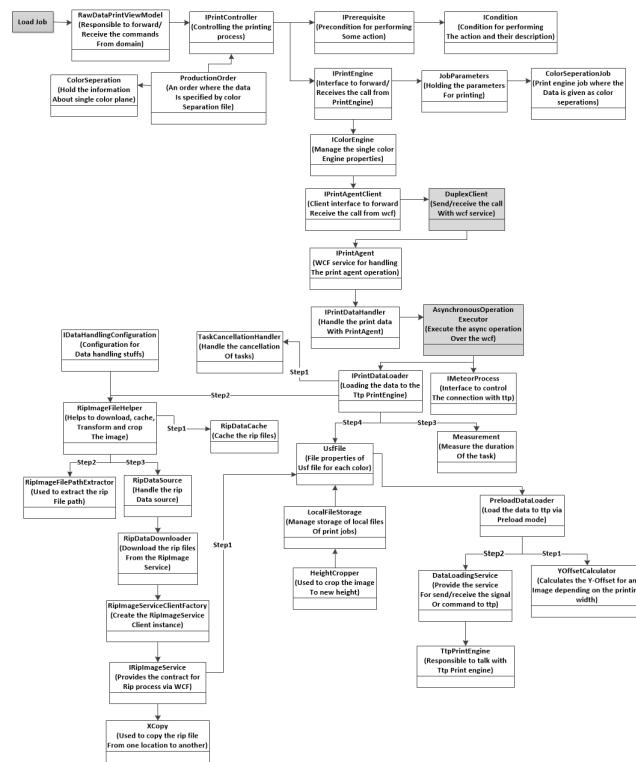


Figure 59: shows the LoadJob object flow

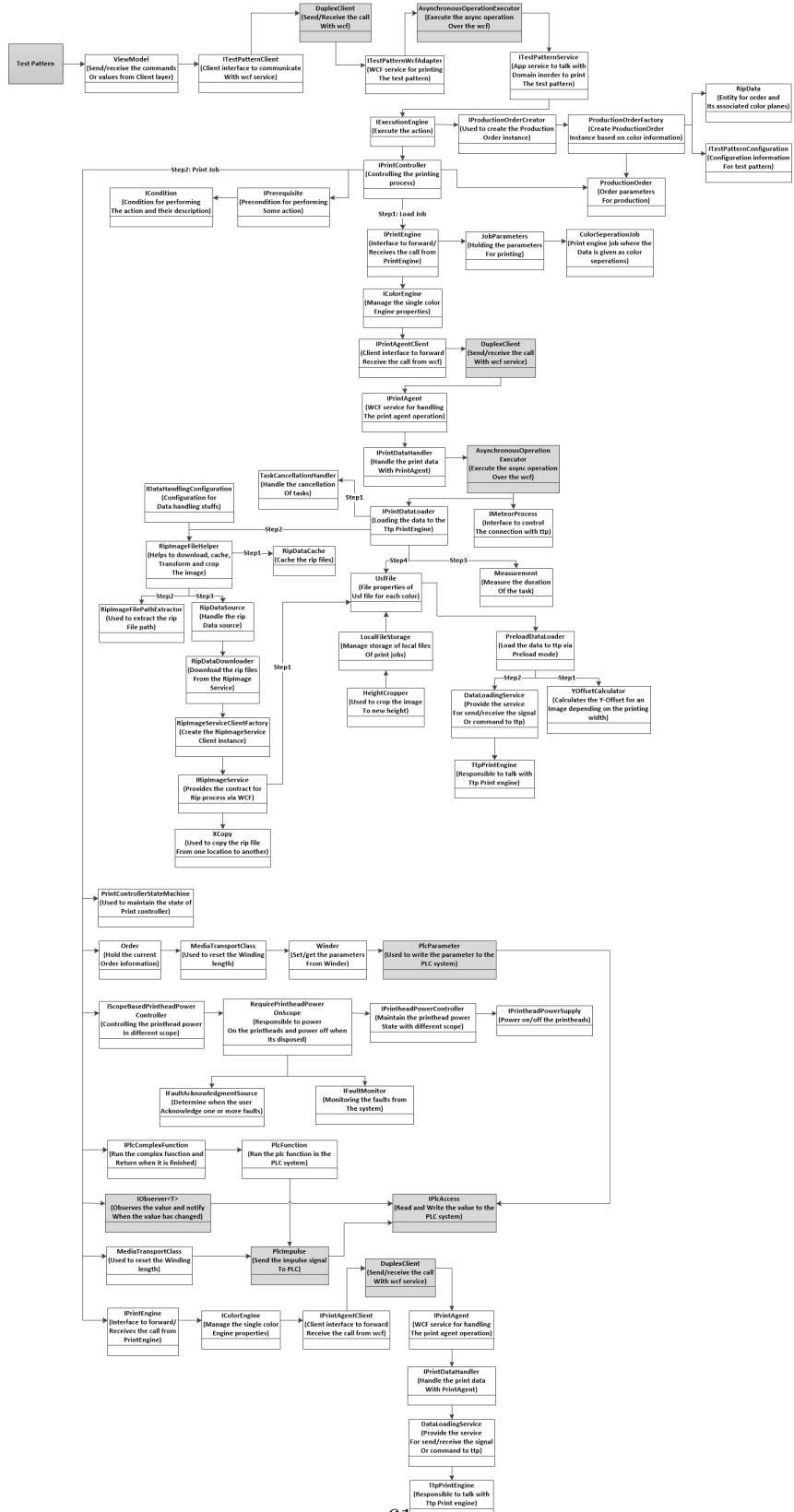


Figure 60: shows the TestPattern object flow

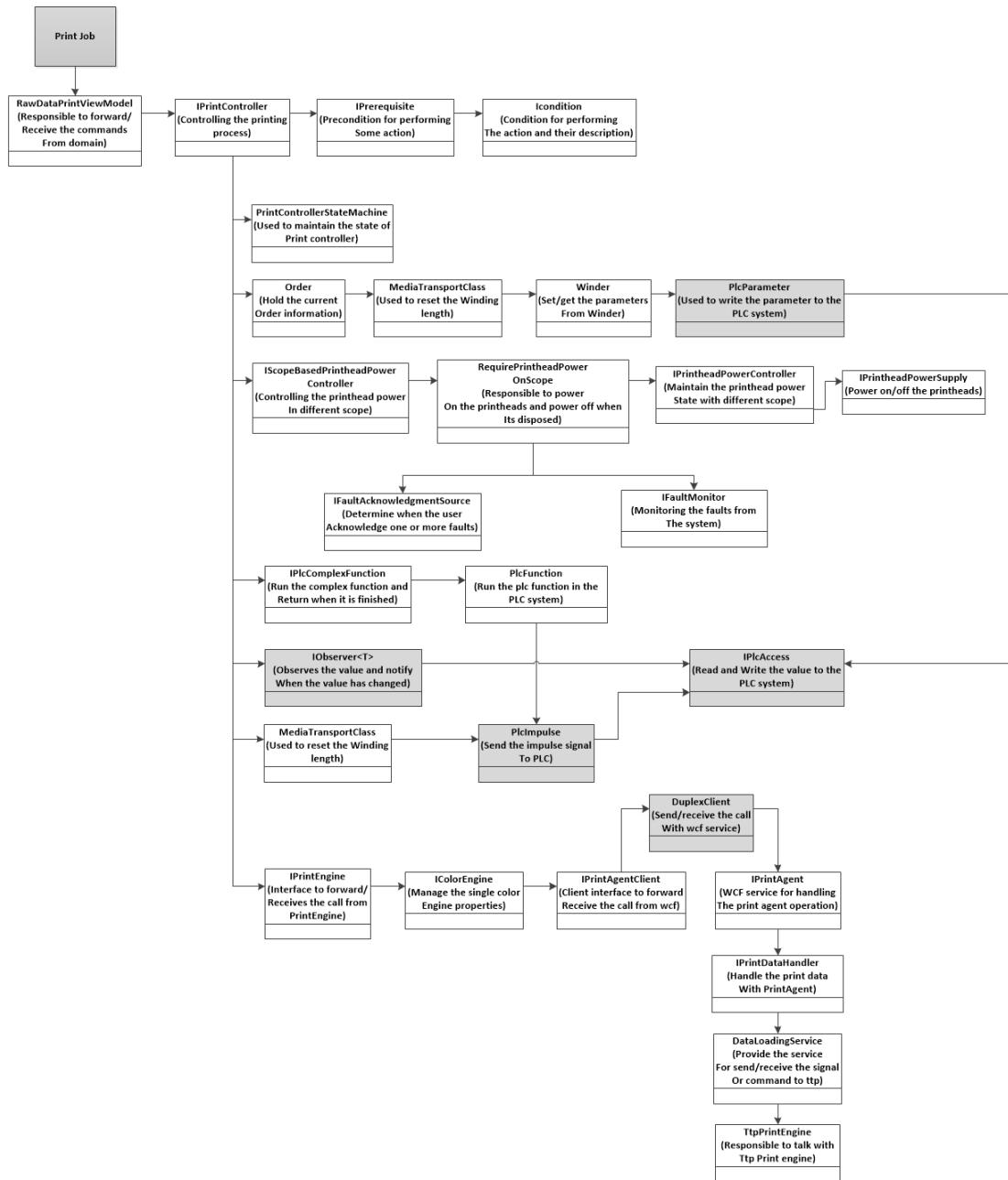


Figure 61: shows the PrintJob object flow

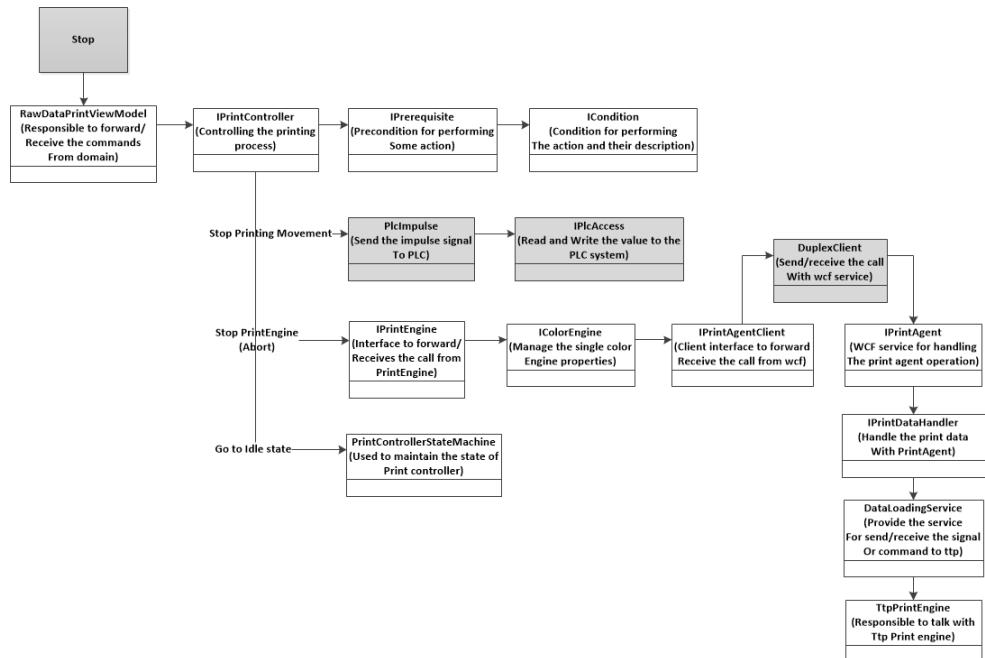


Figure 62: shows the stop printing object flow

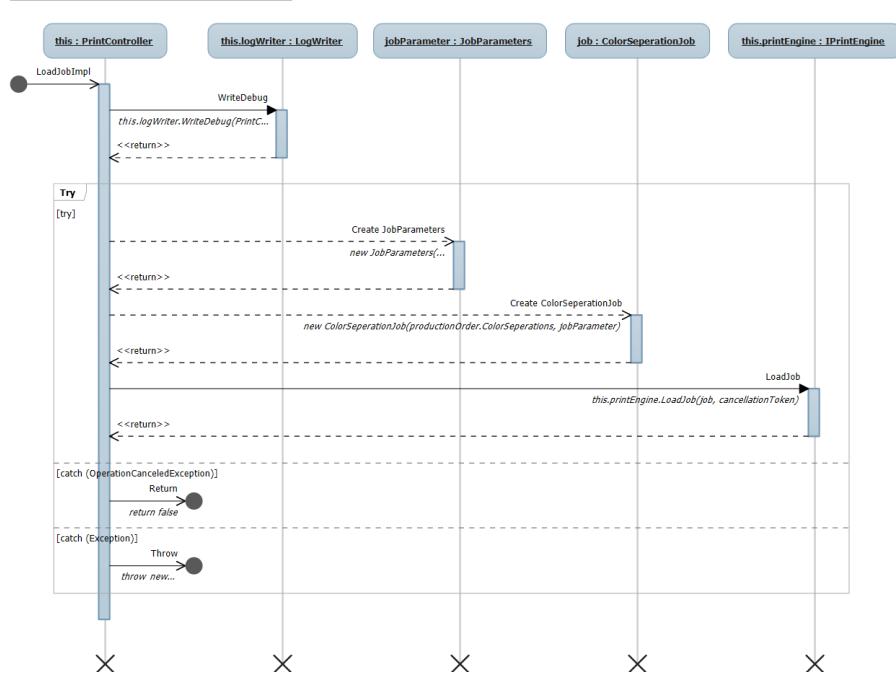


Figure 63: shows the Load Job function flow1

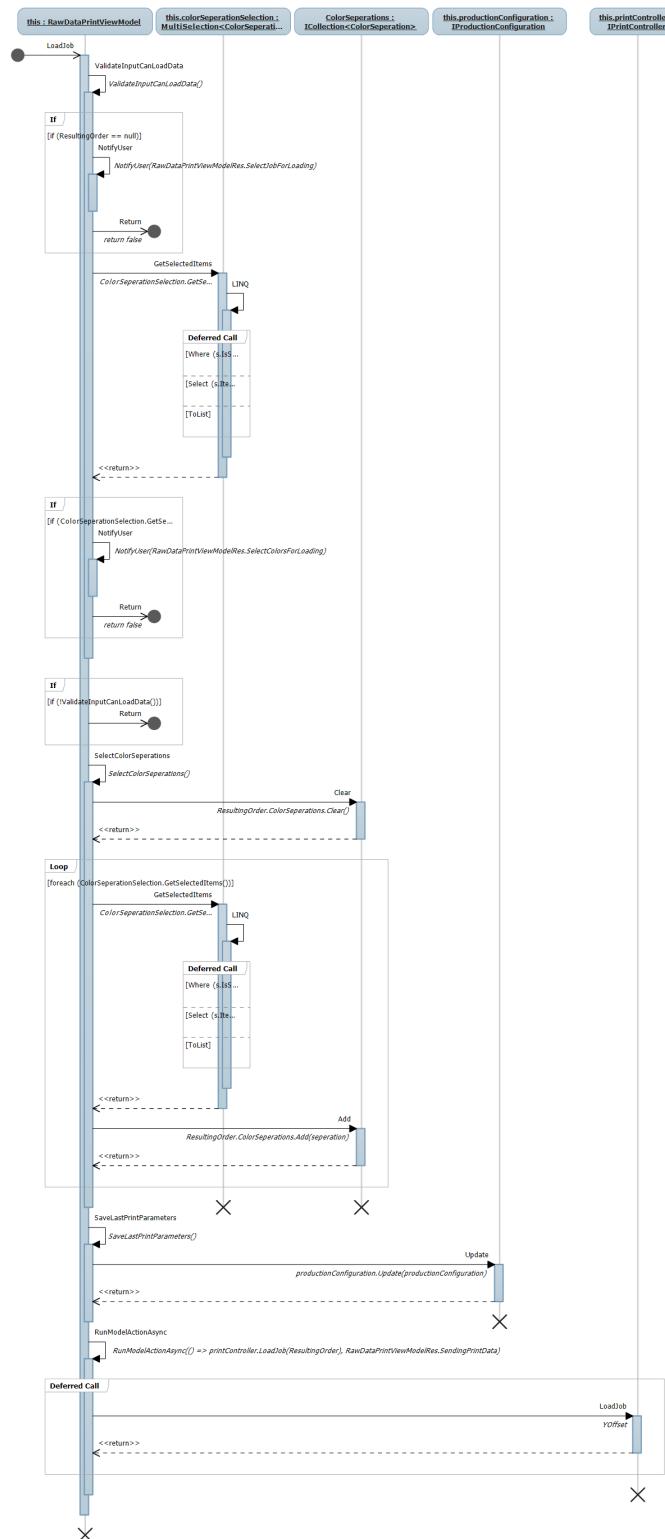


Figure 64: shows the Load Job function flow2

11.2 Printing View for Operators

An operator performs the actions in the following sequence.

1. Select the Production Job to print.
2. Choose all colors or selected colors (Cyan, Red, Yellow, Black).
3. Set Parameters like, Number of copies, Is Seamless Print, Y-Offset, Notes.
4. Choose Load Job and then Print Job or simply Load and Print Job.
5. Stop the print if we want to cancel the current job.

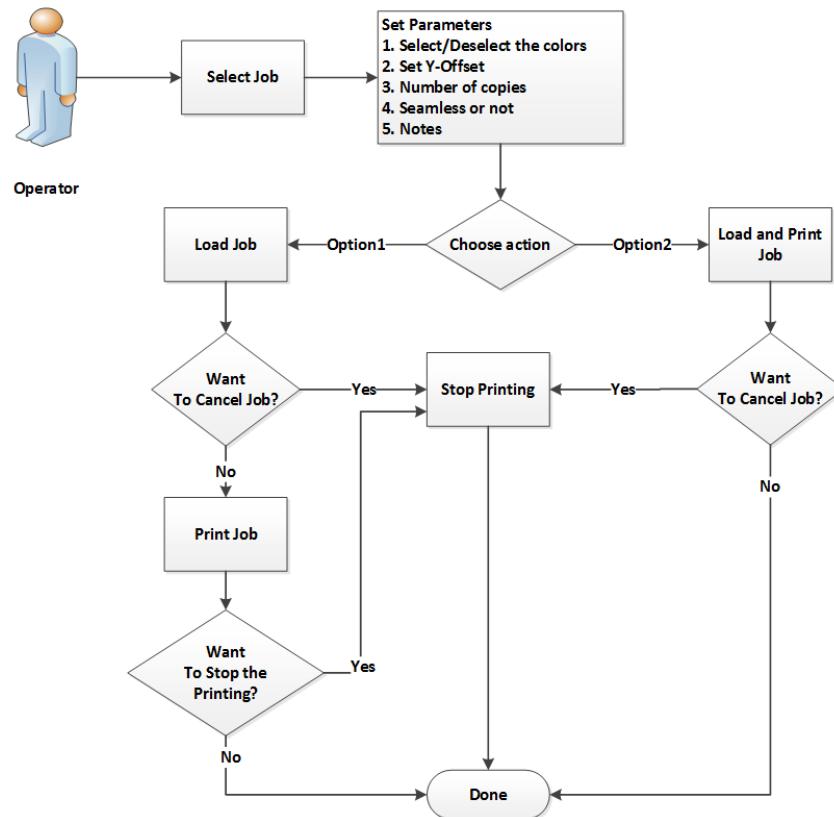


Figure 65: shows the operator sequence of actions for printing

11.3 Printing View for Service Technician

The service technician needs to check the following configuration in order to perform printing operation.

- Printer.Config

- PrintUnit.Config
- Production.Config
- TestPattern.Config
- ConnectionString.Config
- DataLoading.config
- PrintheadSettings.Config
- App.config from PrintAgent
- App.config from Main UI

11.3.1 Printer.Config for printing

The service technician needs to check if the Printer.config is configured correctly.

```

<?xml version="1.0" encoding="utf-8"?>
<Printer printingDirection="Width"
..... machineIdentifier="Palis2250"
..... isScanPrinter="false">

...<!-- currentStandbyMode possible values => Standby, Production or None-->
...<!-- productionStandbyTimeout: 0 disables the timer-->
<standby currentStandbyMode="Standby" productionStandbyTimeout="00:59:00">
...<!--Strategy types possible values => Spit, MeniscusActivation-->
...<standbyStrategy type="Spit" executionDuration="00:00:01" maintenanceInterval="01:00:00" />
...<productionStandbyStrategy type="Spit" executionDuration="00:00:01" maintenanceInterval="01:00:00" />
</standby>

...<colors>
...<color id="Cyan" displayName="Cyan" argb="#8800FFFF">
...<printheads>...</printheads>
...</color>

...<color id="Red" displayName="Rot" argb="#FF000000">
...<printheads>...</printheads>
...</color>

...<color id="Yellow" displayName="Gelb" argb="#88FFFF00">
...<printheads>...</printheads>
...</color>

...<color id="Black" displayName="Schwarz" argb="#88000000">
...<printheads>...</printheads>
...</color>
...</colors>
</Printer>
```

Figure 66: shows the Printer.Config for printing

11.3.2 PrintUnit.Config for printing

The service technician needs to check if the PrintUnit.Config is configured correctly.

```

<printunit
    ..currentMode="DefaultMode"
    ..downloadedLocalRipFolder="C:\Spool\"/>
    ..isStepperReferenceMovementEnabled="true"
    ..printdataFolder="",\RipMetadata"
    ..showStartPrintNodes="true"
    ..testpatternFolder="C:\Spool\Testpattern"
    ..waveformFolder=""
    ..maxFaultPrintHeadTemperature="40"
    ..minimalTemperatureExcedanceDuration="00:00:05"
    ..printHeadAlarmUptime="00:00:40">

    ...<printnodes>
        ...<printnode>
            ..colorId="Cyan"
            ..connectedPrintheads="20"
            ..daisyChainStart="1"
            ..ipAddress="10.0.50.166"
            ..macAddress="74-D4-35-16-54-8E"
            ..maintenanceServiceConfiguration="MaintenanceService2"
            ..meteorConfiguration("\\10.0.50.167\MeteorConfig\meteor.cfg"
            ..printAgentConfiguration="PrintAgentNode1"
            ..printdataFolder="\10.0.50.166\Transfer"
            + ..deltaForUpperwarning="1.2"
            + ..deltaForLowerwarning="2"/>

        ...<printnode>
            ..colorId="Red"
            ..connectedPrintheads="20"
            ..daisyChainStart="1"
            ..ipAddress="10.0.50.167"
            ..macAddress="74-D4-35-1F-C5-32"
            ..maintenanceServiceConfiguration="MaintenanceService2"
            ..meteorConfiguration("\\10.0.50.167\MeteorConfig\meteor.cfg"
            ..printAgentConfiguration="PrintAgentNode2"
            ..printdataFolder="\10.0.50.167\Transfer"
            + ..deltaForUpperwarning="1.2"
            + ..deltaForLowerwarning="2"/>

        ...<printnode>
            ..colorId="Yellow"
            ..connectedPrintheads="20"
            ..daisyChainStart="1"
            ..ipAddress="10.0.50.168"
            ..macAddress="74-D4-35-1E-DC-31"
            ..maintenanceServiceConfiguration="MaintenanceService2"
            ..meteorConfiguration("\\10.0.50.168\MeteorConfig\meteor.cfg"
            ..printAgentConfiguration="PrintAgentNode3"
            ..printdataFolder="\10.0.50.168\Transfer"
            + ..deltaForUpperwarning="1.2"
            + ..deltaForLowerwarning="2"/>

        ...<printnode>
            ..colorId="Black"
            ..connectedPrintheads="20"
            ..daisyChainStart="1"
            ..ipAddress="10.0.50.169"
            ..macAddress="74-D4-35-1F-5D-58"
            ..maintenanceServiceConfiguration="MaintenanceService2"
            ..meteorConfiguration("\\10.0.50.169\MeteorConfig\meteor.cfg"

```

Figure 67: shows the PrintUnit.Config for printing

11.3.3 Production.Config

The service technician needs to check if the Production.Config is configured correctly.

```

<?xml version="1.0" encoding="utf-8".?>
<Printer_Production
    ..lastRipData="xx.bmp"
    ..numberOfCopies="25"
    ..isSeamless="false"
    ..showSeamlessOption="true"
    ..isAlignToCenter="true"
    ..showNumberOfCopies="true"
    ..yOffset="10"/>

```

Figure 68: shows the Production.Config for printing

11.3.4 TestPattern.Config

The service technician needs to check if the TestPattern.Config is configured correctly.

```

<?xml version="1.0" encoding="utf-8" ?>
<Printer.TestPattern>
  <testPatterns>
    <add
      purpose="NozzleCheck"
      jobNameUsed="TargetUsf5x5Pixels"
      numberOfCopies="2"/>
    <add
      purpose="CartridgeCalibration"
      jobNameUsed="TargetUsf5x5Pixels"
      numberOfCopies="2"/>
    <add
      purpose="PrintheadCalibration"
      jobNameUsed="TargetUsf5x5Pixels"
      numberOfCopies="2"/>
    <add
      purpose="DensityCalibration"
      jobNameUsed="TargetUsf5x5Pixels"
      numberOfCopies="2"/>
    <add
      purpose="AdjustPaperThickness"
      jobNameUsed="AdjustPaperThickness"
      numberOfCopies="2"/>
  </testPatterns>
</Printer.TestPattern>

```

Figure 69: shows the TestPattern.Config for printing TestPatterns

11.3.5 ConnectionString.Config

The service technician needs to check if the ConnectionString.Config is configured correctly.

```

<?xml version="1.0" encoding="utf-8" ?>
<connectionStrings>
  <add
    name="MachineContext"
    providerName="System.Data.SqlClient"
    connectionString="
      Server=.\SQLEXPRESS;
      Database=palistest;
      integrated Security=true;..";
      MultipleActiveResultSets=true;"/>
</connectionStrings>

```

Figure 70: shows the ConnectionString.Config

11.3.6 DataHandling.Config

The service technician needs to check if the DataHandling.Config is configured correctly.

```
<?xml version="1.0" encoding="utf-8" ?>
<dataHandling>
    <PreparationQueueBufferCapacity>4</PreparationQueueBufferCapacity>
    <DataProducerQueueCapacity>4</DataProducerQueueCapacity>
    <LocalTrunk>C:\PrintAgent</LocalTrunk>
    <DirectoryToRamDisk>E:\PrintAgent</DirectoryToRamDisk>
    <DirectoryToTtpSimFile>C:\Program Files\TTP\Meteor\SimFiles</DirectoryToTtpSimFile>
    <IsReverseScanStripeActivated>false</IsReverseScanStripeActivated>
    <IsCroppingActivated>false</IsCroppingActivated>
    <IsTransposeActivated>true</IsTransposeActivated>
</dataHandling>
```

Figure 71: shows the DataHandling.Config for printing

11.3.7 PrintheadSettings.Config

The service technician needs to check if the PrintheadSettings.Config is configured correctly.

```
<?xml version="1.0" encoding="utf-8" ?>
<!--<printheadsSetting printheadLayout="Single" ReversePrintHead="true">
    <printheads>
        <printhead headID="0" pccNum="1" />
    </printheads>
</printheadsSetting>-->

<!--<printheadsSetting printheadLayout="PreloadUnified" ReversePrintHead="true" .>
    <printheads>
        <printhead headID="0" pccNum="1" />
        <printhead headID="1" pccNum="2" />
        <printhead headID="2" pccNum="4" />
        <printhead headID="3" pccNum="3" />
    </printheads>
</printheadsSetting>-->

<!--<printheadsSetting printheadLayout="PreloadUnified" ReversePrintHead="false">
    <printheads>
        <printhead headID="0" pccNum="1" />
        <printhead headID="1" pccNum="2" />
        <printhead headID="2" pccNum="3" />
    </printheads>
</printheadsSetting>-->
```

Figure 72: shows the PrintheadSettings.Config for printing

11.3.8 App.config from PrintAgent

The service technician needs to check if the App.Config from PrintAgent is configured correctly.

```

..<WcfConfiguration>
...<endPoints>
... ...< endPoint·name="PrintAgentService"·uri="net.tcp://localhost:8083/PrintAgent"·isDebugEnabled="true"·/>
...</endPoints>
..</WcfConfiguration>

..<startup>...</startup>

..<nozzleActivation·configSource="Config\nozzleActivation.config"·/>

..<jobPreparationEngine·configSource="Config\jobPreparationEngine.config"·/>
..<monitor·configSource="Config\monitor.config"·/>
..<testMachine·configSource="Config\testMachine.config"·/>
..<dataHandling·configSource="Config\dataHandling.config"·/>
..<printUnitSetting·configSource="Config\printUnitSetting.config"·/>
..<Printer.Infrastructure.Changelog·pathToChangelog="./Changelog.txt"·/>
..<!--·Configuration·for·the·print·heads.-->
..<printheadsSetting·configSource="Config\printheadsSetting.config"·/>
..<printAreaSetting·configSource="Config\printAreaSetting.config"·/>
..<!--·Configuration·for·the·call·back.-->
..<printEngineCallbackSetting·callbackInterval="3000"·/>
..<!--·Configuration·the·logging.-->
..<loggingConfiguration·configSource="Config\logging.config"·/>
..<!--·Navigation·configuration·only·a·dummy.-->
..<Printer.Infrastructure.Navigation·pathToNavigationFile="Config\Navigation.xml"·/>
..<PrintEngine.PrintAgentHost.MultipleInstance·canAppRunMutipleInstances="true"·/>
..<system.serviceModel>
...<!--·Configurations·for·bindings·used·by·the·clients-->
...<bindings·configSource="Config\system.serviceModel.bindings.config"·/>
...<!--·Client·configuration-->
...<client·configSource="Config\system.serviceModel.client.config"·/>
...<!--·Behaviors·configuration-->
...<behaviors·configSource="Config\system.serviceModel.behaviors.config"·/>

..</system.serviceModel>
...

```

Figure 73: shows the App.Config for PrintAgent

11.3.9 App.config from Main UI

The service technician needs to check if the App.Config from Main UI is configured correctly.

```

<WcfConfiguration>
  <endPoints>
    <EndPoint name="PrinterStatusClient" uri="net.tcp://localhost:8010/PrinterStatus" isDebugEnabled="true" />
    <EndPoint name="PrinterStatusService" uri="net.tcp://localhost:8010/PrinterStatus" isDebugEnabled="true" />
    <EndPoint name="PrinterConfigurationClient" uri="net.tcp://localhost:9001/PrinterConfigurationService" isDebugEnabled="true" />
    <EndPoint name="PrinterConfigurationService" uri="net.tcp://localhost:9001/PrinterConfigurationService" isDebugEnabled="true" />
    <EndPoint name="CleaningService" uri="net.tcp://localhost:9000/CleaningService" isDebugEnabled="true" />
    <EndPoint name="CleaningClient" uri="net.tcp://localhost:9000/CleaningService" isDebugEnabled="true" />
    <EndPoint name="UserManagementService" uri="net.tcp://localhost:9002/UserManagementService" isDebugEnabled="true" />
    <EndPoint name="UserManagementClient" uri="net.tcp://localhost:9002/UserManagementService" isDebugEnabled="true" />
    <EndPoint name="PrintheadCalibrationService" uri="net.tcp://localhost:9003/PrintheadCalibrationService" isDebugEnabled="true" />
    <EndPoint name="PrintheadCalibrationClient" uri="net.tcp://localhost:9003/PrintheadCalibrationService" isDebugEnabled="true" />
    <EndPoint name="TestPatternService" uri="net.tcp://localhost:9004/TestPatternService" isDebugEnabled="true" />
    <EndPoint name="TestPatternClient" uri="net.tcp://localhost:9004/TestPatternService" isDebugEnabled="true" />
    <EndPoint name="PrintEnginePowerSupplyService" uri="net.tcp://localhost:9005/PrintEnginePowerSupplyService" isDebugEnabled="true" />
    <EndPoint name="PrintEnginePowerSupplyClient" uri="net.tcp://localhost:9005/PrintEnginePowerSupplyService" isDebugEnabled="true" />
    <EndPoint name="DensityCalibrationService" uri="net.tcp://localhost:9006/DensityCalibrationService" isDebugEnabled="true" />
    <EndPoint name="DensityCalibrationClient" uri="net.tcp://localhost:9006/DensityCalibrationService" isDebugEnabled="true" />
    <EndPoint name="CartridgeCalibrationService" uri="net.tcp://localhost:9007/CartridgeCalibrationService" isDebugEnabled="true" />
    <EndPoint name="CartridgeCalibrationClient" uri="net.tcp://localhost:9007/CartridgeCalibrationService" isDebugEnabled="true" />

    <EndPoint name="PrintAgentNode1" uri="net.tcp://10.0.50.166:8083/PrintAgent" isDebugEnabled="true" />
    <EndPoint name="PrintAgentNode2" uri="net.tcp://10.0.50.167:8083/PrintAgent" isDebugEnabled="true" />
    <EndPoint name="PrintAgentNode3" uri="net.tcp://10.0.50.168:8083/PrintAgent" isDebugEnabled="true" />
    <EndPoint name="PrintAgentNode4" uri="net.tcp://10.0.50.169:8083/PrintAgent" isDebugEnabled="true" />

    <EndPoint name="PrinterPositioningService" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true" />
    <EndPoint name="PrinterPositioningClient" uri="net.tcp://localhost:9008/PrinterPositioningService" isDebugEnabled="true" />
    <EndPoint name="CurrentOrderService" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true" />
    <EndPoint name="CurrentOrderClient" uri="net.tcp://localhost:9009/CurrentOrderService" isDebugEnabled="true" />

    <EndPoint name="ComputedDensityCalibrationService" uri="net.tcp://localhost:9010/ComputedDensityCalibrationService" isDebugEnabled="true" />
    <EndPoint name="ComputedDensityCalibrationClient" uri="net.tcp://localhost:9010/ComputedDensityCalibrationService" isDebugEnabled="true" />

    <EndPoint name="DensityProfileService" uri="net.tcp://localhost:9011/DensityProfileService" isDebugEnabled="true" />
    <EndPoint name="DensityProfileClient" uri="net.tcp://localhost:9011/DensityProfileService" isDebugEnabled="true" />

    <EndPoint name="UserTransactionService" uri="net.tcp://localhost:9012/UserTransactionService" isDebugEnabled="true" />
    <EndPoint name="UserTransactionClient" uri="net.tcp://localhost:9012/UserTransactionService" isDebugEnabled="true" />

```

Figure 74: shows the App.Config for MainUI

12 Conclusion

The result of the analysis are presented in this section, what we could infer from the above report and what are the future work that can be done.

1. What new information we got from the analysis?

- we hope for the reader of this document, could get a better understanding of various process involved in Palis2250 software system.
- This document is covering different stakeholder view points to give a comprehensive overview about the software system.
- This document can be used as a basis for further analysis who would like to build tools and applications on top it.