KUARIO

XPP

KUARIO Print Administrator manual

Date: July 21, 2026

Version: 2026.6

Classification: Admin manual

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher. Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks. While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

PUBLISHER
KUARIO B.V.

PRODUCTION
KUARIO B.V.

Contents

Document history **4**

Introduction **5**

Prerequisites **5**

Obtaining the KUARIO Print application **6**

Setup features **6**

Settings.json **6**

KUARIO Print deployment via Microsoft Intune **8**

1. Prerequisites *8*

2. Prepare the Installation Script *8*

3. Package the application *11*

4. Azure / Intune Portal Configuration *12*

KUARIO Print Deployment via Group Policy (GPO) **14**

1. Preparation: Network Share *14*

2. Logon Script *15*

3. Essential GPO Policies ("Security Bypass") *16*

4. Step-by-Step GPO Configuration *16*

5. Summary of GPO Logic *17*

Deploying KUARIO Print on macOS **18**

Deploying KUARIO on Linux (Debian/Ubuntu) **19**

1. Preparation *19*

2. Deploy *19*

3. Execution Command *19*

Logs **20**

Troubleshooting **21**

1. Core Service & Port Troubleshooting (Applies to All) *21*

2. Intune-Specific Troubleshooting *21*

3. GPO-Specific Troubleshooting *22*

4. Common Error Codes Table *23*

5. Quick Verification Checklist for IT Staff *23*

Support **23**

Document history

Revision Date	Revision List
28-01-2026	Initial document
23-03-2026	Added json options
01-04-2026	Refined PowerShell script for Intune
21-04-2026	Improved PowerShell script for Intune
14-07-2026	Rewrote PowerShell script for Intune

Introduction

KUARIO Print allows you to print from a Windows, Linux or macOS computer. Print jobs can then be physically printed by a KUARIO MFP Outlet.

To have access to the more advanced options of KUARIO Print, some configurations are available. This manual will address these configurations and will show how KUARIO Print can be deployed automatically on computers in network environments.

Please enjoy the advanced usage of KUARIO Print,

Prerequisites

KUARIO Print application
Administrator permissions

Obtaining the KUARIO Print application

The latest version of the KUARIO Print application can be downloaded from the KUARIO website: KUARIO || KUARIO Print Driver download

Setup features

After installation and setup, a configuration JSON file is generated. If the settings.json is already present prior to installation, onboarding will be skipped and the installer will run silently.
The config JSON file is located on the following locations

- Windows

  %programdata%\KUARIO Print\settings.json

- macOS

  ~/Library/Application Support/KUARIO Print/settings.json

- Linux (Ubuntu/Debian)

  ~/.config/KUARIO Print/settings.json

Settings.json

With a standard user installation, only two settings will be populated in the settings.json:

```
{
  "clientPublic":false,
  "autoUpdate":true
}
```

However, more options are available to pre-configure.
The following settings are configurable:

Setting	Description	Options
autoUpdate	Automatically check for updates (boolean)	true / false
deviceUUID	Randomly generated UUID used for login security (string)	'be1edf5e-b3ba-4c86-8d06-5da0d6ad114f'
clientPublic	The client will act like a public client (boolean)	true / false
qrLogin	Show a login QR, to allow login through QR codes (boolean)	true / false
jobCodeLogin	Show the job code login button (boolean)	true / false
jobCodeFixedUser	Static username/email for job-code login (string)	true / false
logoutTimer	Timer in seconds to auto logout a public user (int)	60
showQuitApp	Show the option to quit the app (boolean)	true / false
defaultColor	Force the job to a default color (string)	'color' / 'greyscale' / 'monochrome'
defaultDuplex	Force the job to a have duplex enabled (boolean)	true / false
defaultFinishingHide	Hide the finishing options in the job (boolean)	true / false
paperSizes	Supported paper sizes (array)	['iso_a4_210x297mm', iso_a5_148x210mm, 'iso_a3_297x420mm']
serverReadUrl	The url of the KUARIO API (string)	'api.kuario.com'
printerInstall	Install the printer on app boot (boolean)	true / false

IMPORTANT NOTE: for printer Install, local Administrator permissions are required.

KUARIO Print deployment via Microsoft Intune

The following section describes how to deploy the **KUARIO Print** application and its virtual printer to workstations where users do not have local administrator permissions. The deployment uses **Microsoft Intune (Win32 App)** to run the installation with System privileges while targeting the configuration to the active user's profile.

1. Prerequisites

- **Files Needed:** KUARIO-Print-x64.exe and your specific settings.json.
- **Tool Needed:** Microsoft Win32 Content Prep Tool (IntuneWinAppUtil.exe).
- **Target:** Windows 10/11 64-bit devices enrolled in Intune.

---

2. Prepare the Installation Script

Create a folder named **C:\KuarioPackage**.

Place inside this folder:

- The KUARIO Print installer (**KUARIO-Print-x64.exe**)
- The populated **settings.json** file
- A new file named **Install-Kuario-Intune-Detection.ps1**
- A new file named **Install-Kuario-Intune.ps1** and paste in the PowerShell code below (this makes sure the Intune detection works)

PowerShell

```
# Detection Script for Intune
$PrinterName = "KUARIO Print"

if (Get-Printer -Name $PrinterName -ErrorAction SilentlyContinue) {
    # Intune looks for any standard output to standard output (STDOUT) as
success
    Write-Host "Found it!"
    exit 0
} else {
    # Exiting with a non-zero code tells Intune it is not installed
    exit 1
}
```

What the following PowerShell script does:

This script places the settings.json in the user's folder, which triggers the application's built-in silent installation mode. It then installs the application in user context, starts the local printer service and installs the printer as .SYSTEM

PowerShell

```
# Intune Machine-Wide Deployment Script for KUARIO Print (V11)

$ConfigDir = "$env:ProgramData\KUARIO Print"
$InstallerName = "kuario-print-x64.exe"
$AppPath = "$env:ProgramFiles\kuario-print\kuario-print.exe"
$Port = 10130

# 1. Identify Logged-In User
$ExplorerProcess = Get-CimInstance Win32_Process -Filter "Name='explorer.exe'" | Select-Object -First 1
if (!$ExplorerProcess) {
    Write-Output "No user currently logged in. Exiting so Intune can retry when a user is present."
    exit 1
}
$OwnerInfo = Invoke-CimMethod -InputObject $ExplorerProcess -MethodName GetOwner
$TaskPrincipalId = "$($OwnerInfo.Domain)\$($OwnerInfo.User)"


# 2. Pre-Flight Cleanup: Nuke any Ghost Printers
Write-Output "Checking for orphaned KUARIO Print installations..."
$GhostPrinter = Get-Printer -Name "KUARIO Print" -ErrorAction SilentlyContinue
$GhostRegPath = "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Print\Printers\KUARIO Print"

if ($GhostPrinter -or (Test-Path $GhostRegPath)) {
    Write-Output "Ghost printer detected. Forcing cleanup..."
    Remove-Printer -Name "KUARIO Print" -ErrorAction SilentlyContinue
    if (Test-Path $GhostRegPath) {
        Remove-Item -Path $GhostRegPath -Recurse -Force -ErrorAction SilentlyContinue
    }
    Restart-Service -Name Spooler -Force
    Start-Sleep -Seconds 3
    Write-Output "Cleanup complete. Spooler restarted."
}


# 3. Pre-Configure Firewall as SYSTEM (App Whitelist)
Write-Output "Configuring Firewall Rules..."
$fwRuleName = 'KUARIO Print Connector'
$existingRule = Get-NetFirewallRule -DisplayName $fwRuleName -ErrorAction SilentlyContinue

if (-not $existingRule) {
    try {
        New-NetFirewallRule -DisplayName $fwRuleName `
            -Description 'Allow KUARIO Print executable inbound traffic' `
            -Direction Inbound -Action Allow -Program $AppPath `
            -Profile Any -Enabled True | Out-Null
        Write-Output "Firewall rule '$fwRuleName' created for application"
    } catch {
        Write-Output "WARNING: Failed to create firewall rule: $_"
    }
} else {
    Write-Output "Firewall rule '$fwRuleName' already exists - skipping"
}


# 4. Disable 'Let Windows manage my default printer' via User Policy Injection
Write-Output "Enforcing User Policy to disable Windows default printer management..."
$UserSID = (New-Object System.Security.Principal.NTAccount($OwnerInfo.Domain,
$OwnerInfo.User)).Translate([System.Security.Principal.SecurityIdentifier]).Value

$ActiveUserPolicyPath = "Registry::HKEY_USERS\$UserSID\SOFTWARE\Policies\Microsoft\Windows NT\Printers"
if (!(Test-Path $ActiveUserPolicyPath)) { New-Item -Path $ActiveUserPolicyPath -Force | Out-Null }
Set-ItemProperty -Path $ActiveUserPolicyPath -Name "DisableDefaultPrinterManagement" -Value 1 -Force

$ActiveUserTogglePath = "Registry::HKEY_USERS\$UserSID\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows"
if (!(Test-Path $ActiveUserTogglePath)) { New-Item -Path $ActiveUserTogglePath -Force | Out-Null }
Set-ItemProperty -Path $ActiveUserTogglePath -Name "LegacyDefaultPrinterMode" -Value 1 -Force

reg.exe load "HKU\DefaultUserHive" "C:\Users\Default\NTUSER.DAT" 2>&1 | Out-Null
$DefaultUserPolicyPath = "Registry::HKEY_USERS\DefaultUserHive\SOFTWARE\Policies\Microsoft\Windows
NT\Printers"
if (!(Test-Path $DefaultUserPolicyPath)) { New-Item -Path $DefaultUserPolicyPath -Force | Out-Null }
```

```
Set-ItemProperty -Path $DefaultUserPolicyPath -Name "DisableDefaultPrinterManagement" -Value 1 -Force
[gc]::Collect()
[gc]::WaitForPendingFinalizers()
reg.exe unload "HKU\DefaultUserHive" 2>&1 | Out-Null


# 5. Create ProgramData folder and stage settings.json
Write-Output "Creating ProgramData folder and staging settings.json..."
if (!(Test-Path $ConfigDir)) { New-Item -ItemType Directory -Path $ConfigDir -Force | Out-Null }
Copy-Item -Path ".\settings.json" -Destination "$ConfigDir\settings.json" -Force


# 6. Run the installer silently for all users
Write-Output "Running the installer silently for all users..."
$InstallerArgs = "/allusers /S"
Start-Process -FilePath ".\$InstallerName" -ArgumentList $InstallerArgs -Wait -NoNewWindow


# 7. Launch App as SYSTEM to bypass UAC and initialize IPP Server
Write-Output "Launching KUARIO Print as SYSTEM to initialize IPP Server..."
Start-Process -FilePath $AppPath -WindowStyle Hidden -RedirectStandardOutput "$env:TEMP\kuario_out.log" -
RedirectStandardError "$env:TEMP\kuario_err.log"


# 8. Wait for the port to open
Write-Output "Waiting for port $Port to open..."
$PortReady = $false
$Stopwatch = [System.Diagnostics.Stopwatch]::StartNew()

while ($Stopwatch.Elapsed.TotalSeconds -lt 180) {
    if ((Test-NetConnection -ComputerName localhost -Port $Port -ErrorAction SilentlyContinue).TcpTestSucceeded) {
        $PortReady = $true
        break
    }
    Start-Sleep -Seconds 5
}


# 9. Wait for Printer & Kill SYSTEM App
if ($PortReady) {
    Write-Output "Port $Port is open. Waiting for the app to register the printer with Windows..."

    $PrinterReady = $false
    $Stopwatch.Restart()
    while ($Stopwatch.Elapsed.TotalSeconds -lt 60) {
        if (Get-Printer -Name "KUARIO Print" -ErrorAction SilentlyContinue) {
            $PrinterReady = $true
            break
        }
        Start-Sleep -Seconds 3
    }

    if (-not $PrinterReady) {
        Write-Output "App took too long. Installing printer manually as fallback..."
        Add-Printer -Name "KUARIO Print" -IppURL "ipp://localhost:$Port/" -ErrorAction SilentlyContinue
    } else {
        Write-Output "Success: Printer automatically registered by the app."
    }

    Write-Output "Killing SYSTEM instance of the app so it doesn't block the user..."
    Stop-Process -Name "kuario-print" -Force -ErrorAction SilentlyContinue

    Write-Output "Waiting for Port $Port to be fully released..."
    while ((Test-NetConnection -ComputerName localhost -Port $Port -ErrorAction
SilentlyContinue).TcpTestSucceeded) {
        Start-Sleep -Seconds 2
    }

# 10. Persistent Run Keys (Ensures the app runs and defaults on future reboots)
    Write-Output "Configuring persistent auto-start and default printer logic for future reboots..."

    $HelperScriptPath = "$ConfigDir\SetDefaultPrinter.cmd"
    $HelperScriptContent = @"
@echo off
```

```
:: Wait 5 seconds for the desktop and print spooler to fully load
timeout /t 5 /nobreak >nul

:: Set KUARIO Print as the default printer using the native Windows Print UI
rundll32.exe printui.dll,PrintUIEntry /y /n "KUARIO Print"
"@

    Set-Content -Path $HelperScriptPath -Value $HelperScriptContent -Encoding Ascii -Force
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"  -Name
"SetKuarioDefaultPrinter" -Value "`"$HelperScriptPath`"" -Force
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"  -Name "KUARIOPrintApp" -
Value "`"$AppPath`"" -Force


# 11. Launch App and Set Default Printer for the Active User (Zero-Touch Handoff)
    Write-Output "Setting default printer and starting KUARIO Print for the currently logged-in user..."
    $TaskName = "LaunchKuarioUser"

    # Action 1: Force the default printer instantly in the user's session
    $ActionPrinter = New-ScheduledTaskAction -Execute "rundll32.exe" -Argument 'printui.dll,PrintUIEntry /y /n "KUARIO
Print"'

    # Action 2: Launch the application
    $ActionApp = New-ScheduledTaskAction -Execute $AppPath

    $Principal = New-ScheduledTaskPrincipal -UserId $TaskPrincipalId -LogonType Interactive

    # Register the task with both actions in sequence
    Register-ScheduledTask -TaskName $TaskName -Action $ActionPrinter, $ActionApp -Principal $Principal -Force |
Out-Null

    Start-ScheduledTask -TaskName $TaskName
    Start-Sleep -Seconds 3
    Unregister-ScheduledTask -TaskName $TaskName -Confirm:$false

    Write-Output "Success! KUARIO Print deployed and running."
    exit 0
} else {
    Write-Output "Fail: Port $Port never opened. Printer not installed."
    Stop-Process -Name "kuario-print" -Force -ErrorAction SilentlyContinue
    exit 1
}
```

3. Package the application

You must use the **Microsoft Win32 Content Prep Tool** (IntuneWinAppUtil.exe) to wrap these files.

Open Command Prompt as Administrator and run the Prep Tool:

```
DOS

IntuneWinAppUtil.exe -c "C:\KuarioPackage" -s "Install-Kuario-Intune.ps1" -o
"C:\KuarioOutput"
```

[!NOTE] This command creates a file named **Install-Kuario-Intune.intunewin** in your output folder. This is the only file you will upload to Azure.

4. Azure / Intune Portal Configuration

Log in to the **Microsoft Intune Admin Center** and follow these steps:

Step 1: Add the App

1. Navigate to **Apps** > **Windows** > **Add**.
2. Select **Windows app (Win32)** from the dropdown and click **Select**.
3. **App package file:** Click the folder icon and upload the *Install-Kuario-Unified.intunewin* file.
4. **App Information:** Fill in the Name (e.g., *KUARIO Print*) and Publisher.

Step 2: Program Settings

This is the most critical section. It ensures the script runs with the necessary power.

Field	Command Prompt / Selection
Install command	powershell.exe -ExecutionPolicy Bypass -File Install-Kuario-Intune.ps1
Uninstall command	powershell.exe -Command "Remove-Printer -Name 'KUARIO Print'"
Install behavior	System
Device restart behavior	App install may force a device restart
Return codes	Leave as default (0 = success)
Custom Detection Rule	The Install-Kuario-Intune-Detection.ps1 script

---

Step 3: Requirements

- **Operating system architecture:** x64
- **Minimum operating system:** Windows 10 21H1 (or whichever version matches your fleet)

Step 4: Detection Rules

Intune needs a way to verify the Printer installation was successful, so it doesn't try to install it every time the user logs in. This script looks for the KUARIO Print printer.

1. **Rules format:** Manually configure detection rules.
2. **Script file:** Install-Kuario-Intune-Detection.ps1

Step 5: Assignments

1. **Required:** Add a Group of users or devices to force the installation automatically.
2. **Available:** Add a Group if you want users to be able to install it manually via the **Company Portal**.
3. **End user notifications:** (Optional) Set to **Hide all toast notifications** for a completely silent experience.

Phase 3: Execution Logic (How it works)

When the deployment starts on the workstation, the following sequence occurs automatically:

1. **Permissions:** Intune starts the PowerShell script as **NT AUTHORITY\SYSTEM** (Full Admin).
2. **Config Placement:** The script detects the active user and places settings.json in their profile.
3. **Silent Install:** The script runs KUARIO-Print-x64.exe. Because the JSON is already there, the installer runs silently.
4. **Handoff:** The script uses a one-time Scheduled Task to launch the .exe in the **User Context**.
5. **Printer Install:** Once the user-side app opens **Port 10130**, the script finishes by running the Add-Printer command to map the virtual printer.

KUARIO Print Deployment via Group Policy (GPO)

While **Microsoft Intune** is the modern cloud standard, **Group Policy Objects (GPO)** remain the most common method for on-premises Active Directory environments.

Because GPOs run **Startup Scripts** as the **Local System** account, you can perform the administrative installation of KUARIO Print and then "wait" for a user to log in to complete the setup.

1. Preparation: Network Share

GPOs require a central location where all computers can access the installation files.

1. Create a folder on your file server (e.g., C:\Deploy\Kuario).
2. **Share** the folder (e.g., \\Server\Kuario$).
3. **Permissions:** * **Share Permissions:** Everyone - Read.
    - **NTFS Permissions:** "Domain Computers" and "Domain Users" - Read & Execute.
4. Copy your **three files** into this folder: KUARIO-Print-x64.exe, settings.json, and the script below.

---

2. Logon Script

Save this script in your network share. It handles the Admin install first, then waits for a user to log in to finish the printer mapping.

PowerShell

```
<#
.SYNOPSIS
    KUARIO GPO Deployment Script
    Runs as SYSTEM during Startup to handle Admin tasks.
#>

# 1. Install Application (Administrative Context)
Write-Output "Checking for KUARIO installation..."
$LocalAppDir = "C:\Users\Public\KuarioInstalled.txt"  # Simple marker file

if (!(Test-Path $LocalAppDir)) {
    # Run installer from network share
    Start-Process -FilePath "\\Server\Kuario$\KUARIO-Print-x64.exe" -Wait
    New-Item -Path $LocalAppDir -ItemType File -Value "Installed"
}

# 2. Wait for User Logon
Write-Output "Waiting for an active user session..."
$LoggedInUser = $null
while ($null -eq $LoggedInUser) {
    $LoggedInUser = (Get-CimInstance Win32_Process -Filter "Name='explorer.exe'" |
                Invoke-CimMethod -MethodName GetOwner -ErrorAction
SilentlyContinue).User | Select-Object -First 1
    if ($null -eq $LoggedInUser) { Start-Sleep -Seconds 10 }
}

# 3. Deploy User Settings
$UserConfigPath = "C:\Users\$LoggedInUser\AppData\Roaming\KUARIO Print"
if (!(Test-Path $UserConfigPath)) { New-Item -ItemType Directory -Force -Path
$UserConfigPath }
Copy-Item "\\Server\Kuario$\settings.json" -Destination
"$UserConfigPath\settings.json" -Force

# 4. Printer Mapping (Your verified logic)
$IppURL = "ipp://localhost:10130/"; $PrinterName = "KUARIO Print"
$Stopwatch = [System.Diagnostics.Stopwatch]::StartNew()
while ($Stopwatch.Elapsed.TotalSeconds -lt 120) {
    if ((Test-NetConnection -ComputerName localhost -Port 10130 -ErrorAction
SilentlyContinue).TcpTestSucceeded) {
        Add-Printer -Name $PrinterName -IppURL $IppURL
        break
    }
    Start-Sleep -Seconds 5
}
```

3. Essential GPO Policies ("Security Bypass")

Since 2021, Windows blocks non-admins from installing any printer drivers (the "PrintNightmare" patch). You **must** enable these policies in your GPO to allow the script to work.

Policy 1: Disable Admin-Only Driver Restriction

- **Path:** Computer Configuration > Policies > Administrative Templates > Printers
- **Setting: Restrict printer driver installation to administrators**
- **Value: Disabled** (This allows the script's Add-Printer command to succeed).

Policy 2: Point and Print Restrictions

- **Path:** Computer Configuration > Policies > Administrative Templates > Printers
- **Setting: Point and Print Restrictions**
- **Value: Enabled**
- **Security Prompts:** * "When installing drivers for a new connection": **Do not show warning or elevation prompt**.
    - "When updating drivers": **Do not show warning or elevation prompt**.

---

4. Step-by-Step GPO Configuration

1. **Create GPO:** Open gpmc.msc, create a new GPO named **"App - KUARIO Print"**, and link it to your Workstations OU.
2. **Add Startup Script:**
    - Navigate to: Computer Configuration > Policies > Windows Settings > Scripts > Startup.
    - Go to the **PowerShell Scripts** tab.
    - Click **Add** and paste the UNC path: \\Server\Kuario$\GPO-Install-Kuario.ps1.

3. **Configure Registry (Optional but Recommended):** To ensure the driver restriction is truly off, add this Registry Preference:
    - **Path:** Computer Configuration > Preferences > Windows Settings > Registry
    - **Key:** HKLM\Software\Policies\Microsoft\Windows NT\Printers\PointAndPrint
    - **Value Name:** RestrictDriverInstallationToAdministrators
    - **Value Type:** REG_DWORD
    - **Value Data:** 0

---

5. Summary of GPO Logic

- **Startup:** The computer boots and runs your script as **Local System** (full admin). It installs the software from the share.
- **Logon:** The script remains running in the background. As soon as the user logs in, it detects their folder, drops the settings.json, and maps the printer.
- **Permissions:** Because the script was initiated by the **System** at boot, it retains the permission to "inject" the printer into the user's session without a UAC prompt.

Deploying KUARIO Print on macOS

To deploy applications to macOS workstations in 2026, the industry standard is to utilize **Apple Business Manager (ABM)** integrated with an **MDM provider (Jamf, Kandji, or Mosyle)**. This allows for **Zero-Touch Provisioning**, where apps are deployed silently via the **VPP (Volume Purchase Program)** or as managed **.pkg installers** with Root privileges, removing the need for local administrator credentials during the process.

Deploying KUARIO on Linux (Debian/Ubuntu)

To deploy KUARIO Print, **Ansible** is a widely used and well-known deployment tool for your Linux workstations (Ubuntu/Debian).

1. Preparation

Place your files (kuario.deb, settings.json) on your Admin Workstation or a central file server.

2. Deploy

Copy and paste this YAML logic. It handles the "Sudo" requirements automatically, so the user doesn't get a password prompt.

YAML

```yaml
---
- name: Deploy KUARIO Print to Linux Workstations
  hosts: workstations
  become: yes  # Runs as Root (Admin)

  tasks:
    - name: Create KUARIO config directory in user profile
      file:
        path: "/home/{{ ansible_user }}/.config/kuario-print"
        state: directory
        owner: "{{ ansible_user }}"
        mode: '0755'

    - name: Copy settings.json to the workstation
      copy:
        src: ./settings.json
        dest: "/home/{{ ansible_user }}/.config/kuario-print/settings.json"
        owner: "{{ ansible_user }}"

    - name: Install KUARIO Print package
      apt:
        deb: "/tmp/kuario-print-linux.deb"
        state: present

    - name: Setup IPP Printer (equivalent to your PS1 script)
      shell: |
        lpadmin -p "KUARIO_Print" -v "ipp://localhost:10130/" -E -m everywhere
      become: yes
```

3. Execution Command

From your admin terminal, run: ansible-playbook -i hosts deploy_kuario.yml

Logs

Log files are generated when a warning or error has occurred.

After the max file size (128kb) has been reached, the log file (main.log) is renamed to main.old.log and a new empty main log is created. The previous old.log will be overwritten.

The log file is located on the following locations

- Windows

  %appdata%\KUARIO Print\logs\main.log

- MacOS

  ~/Library/Application Support/KUARIO Print/logs/main.log

- Linux

  ~/.config/KUARIO Print/logs/main.log

Troubleshooting

1. Core Service & Port Troubleshooting (Applies to All)

If the software installs but the "KUARIO Print" printer is missing, the local service or the IPP port is likely the culprit.

- **Check Port 10130:** The virtual printer depends on the KUARIO service listening on port 10130.
    - **Command:** Test-NetConnection -ComputerName localhost -Port 10130
    - **If Fail:** The KUARIO background app is not running. Ensure kuario-print.exe is active in Task Manager under the user's session.
- **Check settings.json Placement:** The app will not start correctly or link to the user's account if the JSON is missing.
    - **Path:** %ProgramData%\KUARIO Print\settings.json
- **Manual Printer Install Test:** If the port is open but the script fails, try adding it manually in a PowerShell window to see the exact error:
    - **Command:** Add-Printer -Name "Test-Kuario" -IppURL "ipp://localhost:10130/"

---

2. Intune-Specific Troubleshooting

Intune runs in a hidden "System" session, which is the most common cause of "silent hangs."

- **View Intune Management Extension (IME) Logs:** This is the "Black Box" of Intune deployment.
    - **Path:** C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementExtension.log
    - **What to look for:** Search for [Win32App] or the name of your script. It will show you the exact exit code (e.g., 0x80070005 for Access Denied).

- **The "Silent Hang" Test:** If the status in Intune stays "Pending" or "In Progress" for hours, the installer is likely showing a hidden popup window.
    - **Fix:** Ensure your settings.json is being placed *before* the installer runs, as this triggers the silent mode.
- **Testing as SYSTEM:** To replicate Intune exactly, use **PsExec** (psexec -i -s powershell.exe) as described in the previous manual. If the script fails in the PsExec window, it will fail in Intune.

---

3. GPO-Specific Troubleshooting

GPO failures are usually related to network timing or "PrintNightmare" security blocks.

- **Check Startup Script Permissions:** Ensure the \\Server\Kuario$ share gives **Read** permissions to **"Domain Computers."** (Startup scripts run as the computer, not the user).
- **PrintNightmare Security Blocks:** Windows often blocks drivers from being added by scripts.
    - **Check:** Does the printer install if you run the script manually as an Admin, but fail via GPO?
    - **Fix:** Ensure the **Point and Print Restrictions** policy is set to "Disabled" or "Do not show warning" in the GPO.
- **Script Execution Policy:** GPO may block the script if the execution policy is too strict.
    - **Check:** Run Get-ExecutionPolicy on the workstation. It should be Bypass or RemoteSigned.
- **Wait for Network:** If the script fails with "File not found," the PC might be trying to run the script before the Wi-Fi/Ethernet is connected.
    - **GPO Fix:** Enable Computer Configuration > Administrative Templates > System > Group Policy > Specify startup policy processing wait time (Set to 60 seconds).

---

4. Common Error Codes Table

Error Code	Meaning	Common Fix
0x80070005	Access Denied	Script is running in User context instead of System, or the share permissions are wrong.
1603	Fatal error during installation	Often means a previous version is half-installed. Try a manual uninstall first.
0x80070002	File Not Found	The path to KUARIO-Print-x64.exe in the script is incorrect or the network share is unreachable.
1	Generic Script Failure	The Port 10130 timeout was reached. The app took too long to start.

---

5. Quick Verification Checklist for IT Staff

If a user reports the printer is missing, run these three commands in order:

1. Test-Path "$env:ProgramData\KUARIO Print\settings.json" (Must be **True**)
2. Get-Process "kuario-print*" (Must return a running process)
3. Get-Printer -Name "KUARIO Print" (If False, re-run the Add-Printer logic)

Support

When issues occur or if you have any technical inquiries, please log a ticket at KUARIO support: [KUARIO : Help & Support]