# Post Provisioning Tool – Admin Guide

## 1. Purpose

The Post Provisioning Tool automates post-imaging device setup tasks, such as setting hostnames, applying tags, assigning profiles, and recording device details. This guide covers setup, configuration, and distribution for administrators.

## 2. Requirements

* Windows 10 or 11 (x64)
* Local admin rights on test devices
* Workspace ONE UEM admin access
* API credentials with required permissions (REST API + Device Management)
* Download access to GitHub release package

## 3. Folder Structure

The application must be run from: C:\ProgramData\PostProvisioningTool

All files must be stored here:

* PostProvisioningTool.exe (application)
* config.json (configuration file)
* DeviceLocation.csv (location dropdown data)
* DeviceTag.csv (tag IDs and names)
* Any other dependencies

## 4. Getting the App

1. Download the release ZIP from the provided GitHub Releases page.  
2. Extract all files to: C:\ProgramData\PostProvisioningTool  
3. Confirm the folder contains all required files listed above.

## 5. Accessing Admin Mode

Admin Mode is used to set up and change application settings before deployment.

1. Launch the Application – Run C:\ProgramData\PostProvisioningTool\PostProvisioningTool.exe  
2. Enter Admin Mode – Press Ctrl + A (or click 'Admin Mode' if visible) and enter the Admin Password.  
 Default: Omnissa (change before production).

## 6. Configuring the Admin Window

### API Configuration

|  |  |  |
| --- | --- | --- |
| Field | Description | Example |
| API URL | Workspace ONE API base endpoint | https://yourtenant.awmdm.com/API |
| Tenant Code | API tenant code for authentication | ABC123 |
| Username | API service account username | ws1api@company.com |
| Password | API account password (encrypted in config) | (enter securely) |

### Application Settings

|  |  |  |
| --- | --- | --- |
| Field | Description | Example |
| Admin Password | Password for Admin Mode | StrongPass! |
| Registry Path | Where the app stores config/status in Windows registry | SOFTWARE\PostProvisioningTool |
| Log File Path | Path to application log | C:\ProgramData\PostProvisioningTool\Logs\Log.txt |

### CSV Configuration (Upload or File Path)

The tool supports two methods for providing the Device Location CSV and Device Tag CSV:  
  
Option 1 – Upload: Click 'Upload' in Admin Mode and select the file from your local system. The file will be embedded in config.json.  
Option 2 – File Path: Enter a UNC or local path (e.g., \\server\share\DeviceLocation.csv or C:\SharedFiles\DeviceLocation.csv) so the app always pulls the latest version remotely.  
  
Best Practice for Remote File Path:  
- Use a read-only network share accessible by all target devices.  
- This allows updating CSVs without repackaging and redeploying the application.  
- Keep filenames consistent to avoid breaking the reference.

## 7. How the App Works

1. Technician Launch – Runs from C:\ProgramData\PostProvisioningTool\PostProvisioningTool.exe and loads dropdown data from CSVs.
2. User Selection – Technician enters ID, location, device tag, and optionally device profile.
3. Provisioning – Applies tag via WS1 API, sets hostname (if configured), assigns profile, writes status to log and registry.
4. Completion – App closes when finished.

## 8. Preparing for Workspace ONE Distribution

Step 1 – Package the App:  
Place all required files into C:\ProgramData\PostProvisioningTool Zip the folder as PostProvisioningTool.zip

Step 2 – Upload to Workspace ONE:  
1. Go to Apps & Books → Native → Add Application  
2. Select Windows (Win32)  
3. Upload PostProvisioningTool.zip  
4. Set app details and deployment options with install command:

powershell.exe -ExecutionPolicy Bypass -Command "if (!(Test-Path 'C:\ProgramData\PostProvisioningTool')) {New-Item -Path 'C:\ProgramData\PostProvisioningTool' -ItemType Directory -Force}; Expand-Archive -Path 'PostProvisioningTool.zip' -DestinationPath 'C:\ProgramData\PostProvisioningTool' -Force; Start-Process 'C:\ProgramData\PostProvisioningTool\PostProvisioningTool.exe'"

5. Once validated and tested, you can add this into your onboarding workflow. In my testing this was one of the final steps before I handed it off to an end user.

## 9. Troubleshooting

|  |  |  |
| --- | --- | --- |
| Issue | Cause | Fix |
| App starts in Limited Mode | config.json missing | Run Admin Mode and save settings |
| API errors | Bad URL/credentials | Verify API URL and service account |
| Tag not applied | Bad Tag ID in CSV | Correct CSV entry |
| App closes immediately | Runtime error | Check log file in Logs folder |