**Post Provisioning Tool v1.0 – Release Notes**

**Version:** 1.0  
**Build Date:** June 2, 2025  
**Executable:** PostProvisioningTool.exe  
**Platform:** Windows 10 / 11 (64-bit)  
**Framework:** .NET 6 (self-contained, no installation required)

**Overview**

The **Post Provisioning Tool** is a lightweight Windows desktop application designed to automate post-imaging device configuration and validation for Workspace ONE-managed endpoints.

It guides technicians through key post-deployment steps and ensures each device:

* Has a valid hostname
* Is registered with the correct Workspace ONE tag
* Passes basic connectivity and enrollment checks
* Is documented in logs and the Windows Registry

**Core Functionality**

1. **Dynamic UI Workflow**
   * Dropdowns auto-adjust based on CSV location data (up to 5 levels)
   * Technician and hostname inputs are required
   * Only shows relevant fields for minimal confusion
2. **Connectivity & Serial Detection**
   * Validates internet access (google.com) with timeout
   * Detects serial number automatically via PowerShell/WMI
3. **Workspace ONE Integration**
   * Calls /api/mdm/devices?searchby=Serialnumber&id={serial} to verify enrollment
   * Applies Workspace ONE device tag via API
   * Fails fast with clear status if device is not found or tagging fails
4. **Hostname Validation**
   * Enforces hostname rules: max 15 chars, no spaces/specials (hyphens allowed)
   * Logs hostname changes and optionally triggers reboot
5. **Registry Writes**
   * Writes technician name, location, hostname, and date to:
   * HKLM\SOFTWARE\Omnissa\PostProvisioningTool
6. **Logging**
   * Appends all actions to:
   * C:\ProgramData\AirWatch\UnifiedAgent\Logs\PostProvisioningTool\ProvisioningLog.txt
   * Logs include timestamps, status messages, serial number, API URLs, and response metadata
7. **Admin Mode (Ctrl+Shift+A)**
   * Password-protected (default: Omnissa)
   * Allows override actions, config editing, CSV re-imports, and API connectivity tests
8. **Limited Mode**
   * Triggered if config.json is missing or malformed
   * Blocks provisioning and clearly alerts the user
   * Banner displayed and logs capture reason

**Setup Instructions**

**1. Locate PostProvisioningTool.exe in C:\ProgramData\PostProvisioningTool**

No installation required.

**2. Prepare config.json**

Place alongside the .exe. This file should contain:

* Base64-encoded credentials (password, tenant code, admin password)
* Raw plaintext values for API URL, username, paths, and embedded CSVs
* JSON keys:
  + ApiUrl, Username, EncodedPassword, EncodedTenantCode, EncodedAdminPassword,  
    DeviceLocationCsv, DeviceTagCsv, LogFilePath, RegistryPath

**3. Launch the tool**

Run as standard user or elevated depending on your environment's registry and network policies.

**System Requirements**

| **Requirement** | **Minimum** |
| --- | --- |
| OS | Windows 10 or 11 (x64) |
| .NET Runtime | Not required (self-contained) |
| Workspace ONE UEM | API endpoint must be externally reachable |
| PowerShell | Included with Windows |
| Internet | Required to proceed |

**Troubleshooting**

| **Symptom** | **Possible Cause** | **Resolution** |
| --- | --- | --- |
| 🔴 Limited Mode | Missing or invalid config.json | Verify structure and values; test in Admin Mode |
| 🚫 "Tag assignment failed" | Device not found or credentials invalid | Confirm serial number is enrolled; check API creds |
| ❌ App won’t run | Missing dependency or blocked by antivirus | Ensure .exe is unblocked and not quarantined |
| ⏱️ Internet check timeout | No network or DNS resolution | Confirm internet access to google.com within 5 seconds |
| 🔐 Admin Mode won’t unlock | Wrong password | Default is Omnissa; reset via config.json if needed |