# Windows Autopilot
# Hardware Hash Extraction Procedure

Tech to School — Internal Technician Guide

---

For batch extraction of Windows Autopilot hardware identifiers from refurbished and non-distribution-channel devices. Includes Go-Box HID automation for hands-free extraction and live MKL progress tracking.

| Detail | Value |
|---|---|
| Applicable To | Windows 10/11 Pro or Education devices with TPM 2.0 |
| Estimated Time (Go-Box) | ~45 seconds per device (hands-free after plug-in) |
| Estimated Time (USB Script) | 1-2 minutes per device (after initial setup) |
| Estimated Time (Manual) | 3-5 minutes per device |
| Output | CSV + live upload to MKL portal (or CSV-only fallback) |
| Deliverable to Customer | CSV file for upload to their Microsoft Intune admin center |

### Three Extraction Methods

| Method | Speed | Tech Effort | Best For |
|---|---|---|---|
| A: Go-Box HID (recommended) | ~45 sec/device | Plug in, walk away | Batch orders 50+ |
| B: USB Script | 1-2 min/device | Type one command | Medium batches, no Go-Box |
| C: Manual PowerShell | 3-5 min/device | Type 7 commands | Single device, troubleshooting |

# 1. Overview

Windows Autopilot allows schools to pre-register devices in their Microsoft Intune tenant so that when a device is powered on for the first time, it automatically configures itself with the school's policies, apps, and settings — zero-touch deployment.

What we provide is the **hardware hash extraction** — pulling a unique device identifier from each Windows machine and compiling it into a CSV file. The customer's IT team then uploads this CSV to their Intune admin center to complete registration.

## What the Customer Needs

The customer must have:

- Microsoft 365 Education A3 or A5 licensing (includes Intune)
- An active Azure AD / Entra ID tenant
- An IT administrator who can upload the CSV to Intune

## Device Requirements

- Windows 10 or Windows 11 (Pro, Education, or Enterprise)
- TPM 2.0 module (standard on devices manufactured after ~2016)
- Device must be able to boot to Windows OOBE or desktop
- Network connectivity (Ethernet recommended for batch work)

# 2. Go-Box HID Automation Setup

The Go-Box (go-box.com) is a USB HID device that emulates a keyboard — it sends scripted keystrokes to any device it's plugged into. We program it with the exact key sequence needed to open PowerShell at OOBE, extract the hardware hash, and POST it to MKL for live tracking.

**Why Go-Box over USB script?** The technician doesn't need to type anything. Plug in the Go-Box, it types the commands automatically, and the hash uploads to MKL. No hunting for drive letters, no typos, no missed steps.

## Prerequisites

- Go-Box Chrome device (go-box.com) with custom script capability
- Device must be connected to network (Ethernet or Wi-Fi) BEFORE plugging in Go-Box
- MKL portal endpoint configured for hash collection (see Section 6)

### *Step 1: Connect to Go-Box Dashboard*

Power on the Go-Box. Connect to its self-broadcast Wi-Fi network from any browser (laptop, phone, or tablet) to access the configuration dashboard.

### *Step 2: Create the Autopilot Extraction Script*

In the Go-Box dashboard, create a new custom automation template. Program the following keystroke sequence. Each line represents keystrokes the Go-Box will type, with delays between steps to allow commands to execute.

**Keystroke Sequence:**

```
Keystroke Sequence for Go-Box Template
======================================

STEP 1: Open command prompt at OOBE
  Keys:  Shift+F10
  Delay: 3 seconds (wait for cmd window to open)

STEP 2: Launch PowerShell
  Type:  powershell
  Keys:  Enter
  Delay: 3 seconds (wait for PS prompt)

STEP 3: Set TLS and execution policy
  Type:  [Net.ServicePointManager]::SecurityProtocol =
         [Net.SecurityProtocolType]::Tls12;
         Set-ExecutionPolicy -Scope Process -ExecutionPolicy
         RemoteSigned -Force
  Keys:  Enter
  Delay: 2 seconds

STEP 4: Install NuGet provider (silent)
  Type:  Install-PackageProvider -Name NuGet
         -MinimumVersion 2.8.5.201 -Force | Out-Null
  Keys:  Enter
  Delay: 10 seconds (downloads NuGet on first device)

STEP 5: Trust PSGallery and install extraction script
  Type:  Set-PSRepository -Name PSGallery
```

```
          -InstallationPolicy Trusted;
          Install-Script -Name Get-WindowsAutopilotInfo -Force
  Keys:  Enter
  Delay: 15 seconds (downloads script on first device)

STEP 6: Extract hash and POST to MKL
  Type:  (see full PowerShell one-liner below)
  Keys:  Enter
  Delay: 20 seconds (extraction + upload)

STEP 7: Shutdown device
  Type:  shutdown /s /t 5
  Keys:  Enter
```

### *Step 3: The Go-Box PowerShell One-Liner (Step 6)*

This is the command the Go-Box types in Step 6. It extracts the hash, captures the serial number, and POSTs everything to the MKL portal API. Program this as a single typed line in the Go-Box template:

```
$t=Join-Path $env:TEMP 'ap.csv';
Get-WindowsAutopilotInfo -OutputFile $t;
$s=(Get-WmiObject Win32_BIOS).SerialNumber;
$d=Get-Content $t -Raw;
try {
  Invoke-RestMethod -Uri 'https://mkl.techtoschool.com/api/autopilot/hash'
    -Method POST -ContentType 'application/json'
    -Body (@{serial=$s;csv=$d;timestamp=(Get-Date -Format o)}
    | ConvertTo-Json)
} catch {
  Write-Host "Upload failed - saving locally" -ForegroundColor Yellow
};
Write-Host "DONE: $s" -ForegroundColor Green
```

*Note: The Go-Box types this character by character. Ensure the one-liner is entered exactly as shown, on a single line (no line breaks). The semicolons separate commands. The try/catch ensures the device still shuts down even if the MKL upload fails.*

### *Step 4: Set Delays*

Configure delays between each step in the Go-Box template. The delays above are conservative — on subsequent devices (after NuGet and the script are cached), Steps 4 and 5 complete in 1-2 seconds. You can create two templates:

- **AUTOPILOT-FIRST:** Longer delays (for the first device in a batch, when packages need to download)
- **AUTOPILOT-FAST:** Shorter delays (for subsequent devices, packages already cached)

### *Step 5: Save and Test*

Save the template on the Go-Box. Test on a single device first — watch the Go-Box type each command and verify it completes successfully. Check the MKL dashboard to confirm the hash appeared. Adjust delays if commands are executing before the previous one finishes.

# 3. Per-Device Procedure — Go-Box Method

Once the Go-Box is programmed, the per-device procedure is minimal:

**Step 1.** Power on the device. Wait for the OOBE screen.

**Step 2.** Connect the device to the network:

- **Ethernet (preferred):** Plug in a cable. OOBE auto-detects it.
- **Wi-Fi:** Navigate through the OOBE Wi-Fi setup screen before proceeding.

**Step 3.** Plug the Go-Box USB cable into the device.

**Step 4.** Walk away. The Go-Box will:

- Press Shift+F10 to open the command prompt
- Launch PowerShell
- Install required packages (first device only)
- Extract the hardware hash
- Upload it to MKL
- Shut down the device automatically

**Step 5.** Unplug the Go-Box when the device powers off. Move to the next device.

## Assembly-Line Workflow (Recommended for 50+ Devices)

- **Station 1 — Boot + Network:** Power on device, plug in Ethernet
- **Station 2 — Go-Box:** Plug in Go-Box, let it run (~45 sec)
- **Station 3 — Stack:** Unplug when powered off, box/stack device

With two technicians and two Go-Boxes, you can pipeline 4-6 devices simultaneously. Monitor progress live on the MKL dashboard from any phone or laptop.

**Important: First Device vs. Subsequent Devices**

The first device in each batch takes longer (~2-3 minutes) because it must download NuGet and the Get-WindowsAutopilotInfo script from the internet. **Subsequent devices skip these downloads if you don't fully shut down between extractions** (the packages are cached in the session). For maximum speed, use the AUTOPILOT-FIRST template on device #1, then switch to AUTOPILOT-FAST for the rest of the batch.

However, since Step 7 shuts down the device, packages won't persist to the next device. For the fastest pipeline, consider removing the auto-shutdown and having the tech type **shutdown /s /t 0** manually after confirming SUCCESS, or keep the conservative delays.

# 4. USB Script Setup — Method B (Fallback)

If a Go-Box is not available, use a USB drive with the extraction script. The tech must type one command per device instead of zero, but the script handles everything else.

### *Step 1: Prepare a USB Drive*

- Use any USB flash drive (1 GB or larger)
- Format as FAT32 or exFAT
- Label it: **AUTOPILOT**

### *Step 2: Create the Extraction Script*

Save the following as **Extract-AutopilotHash.ps1** on the USB drive:

```
# Extract-AutopilotHash.ps1
# Autopilot Hardware Hash Extraction — Tech to School
# Run from USB at OOBE (Shift+F10) or from an admin PowerShell prompt

$ErrorActionPreference = "Stop"

# Find the USB drive
$usb = Get-Volume | Where-Object { $_.FileSystemLabel -eq "AUTOPILOT" } |
       Select-Object -First 1
if (-not $usb) {
    $usb = Get-Volume | Where-Object { $_.DriveType -eq "Removable" } |
           Select-Object -First 1
}
if (-not $usb) {
    Write-Host "ERROR: USB drive not found." -ForegroundColor Red
    pause; exit 1
}
$usbPath = "$($usb.DriveLetter):\"
$outputDir = Join-Path $usbPath "AutopilotHashes"
if (-not (Test-Path $outputDir)) { New-Item -Path $outputDir -ItemType Directory | Out-Null }

$csvFile = Join-Path $outputDir "autopilot-devices.csv"

Write-Host ""
Write-Host "====================================" -ForegroundColor Cyan
Write-Host "  Autopilot Hash Extraction Tool"     -ForegroundColor Cyan
Write-Host "  Tech to School"                      -ForegroundColor Cyan
Write-Host "====================================" -ForegroundColor Cyan
Write-Host ""

$serial = (Get-WmiObject -Class Win32_BIOS).SerialNumber
Write-Host "Device Serial: $serial" -ForegroundColor Yellow

# Install dependencies if needed
Write-Host "Checking for Get-WindowsAutopilotInfo..." -ForegroundColor Gray
$needsInstall = -not (Get-InstalledScript -Name Get-WindowsAutopilotInfo `
                      -ErrorAction SilentlyContinue)
if ($needsInstall) {
    Write-Host "Installing Get-WindowsAutopilotInfo from PSGallery..."
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
    Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force | Out-Null
    Set-PSRepository -Name PSGallery -InstallationPolicy Trusted
    Install-Script -Name Get-WindowsAutopilotInfo -Force
```

```
}

# Extract the hash
Write-Host "Extracting hardware hash..." -ForegroundColor Cyan
$tempCsv = Join-Path $env:TEMP "autopilot-temp-$serial.csv"
Get-WindowsAutopilotInfo -OutputFile $tempCsv

if (Test-Path $tempCsv) {
    # Save to USB
    $content = Get-Content $tempCsv
    if (-not (Test-Path $csvFile)) {
        $content | Out-File -FilePath $csvFile -Encoding UTF8
    } else {
        $content | Select-Object -Skip 1 | Out-File -FilePath $csvFile -Append -Encoding UTF8
    }

    # Also try to upload to MKL
    try {
        $csvData = Get-Content $tempCsv -Raw
        $body = @{serial=$serial;csv=$csvData;timestamp=(Get-Date -Format o)} | ConvertTo-Json
        Invoke-RestMethod -Uri 'https://mkl.techtoschool.com/api/autopilot/hash' `
            -Method POST -ContentType 'application/json' -Body $body
        Write-Host "Uploaded to MKL portal" -ForegroundColor Green
    } catch {
        Write-Host "MKL upload skipped (offline or endpoint not configured)" `
            -ForegroundColor Yellow
    }

    Remove-Item $tempCsv -Force
    $count = (Get-Content $csvFile | Measure-Object).Count - 1

    Write-Host ""
    Write-Host "SUCCESS: Hash extracted for $serial" -ForegroundColor Green
    Write-Host "Total devices in CSV: $count" -ForegroundColor Green
    Write-Host "CSV location: $csvFile" -ForegroundColor Gray
} else {
    Write-Host "ERROR: Hash extraction failed for $serial" -ForegroundColor Red
}

Write-Host ""
Write-Host "Remove USB and proceed to next device, or type 'exit' to close."
Write-Host ""
pause
```

### *Step 3: Create a Quick-Launch Batch File*

Save the following as **RUN-ME.bat** on the USB root:

```
@echo off
echo Starting Autopilot Hash Extraction...
powershell -ExecutionPolicy Bypass -File "%~dp0Extract-AutopilotHash.ps1"
pause
```

## USB Per-Device Steps

**1.** Boot device to OOBE. Press **Shift + F10** for command prompt.

**2.** Insert AUTOPILOT USB drive.

**3.** Type: **D:\RUN-ME.bat** (or E:) and press Enter.

4. Wait for green SUCCESS message.
5. Remove USB, power off, next device.

# 5. Manual PowerShell — Method C (Troubleshooting)

For single devices or when USB/Go-Box methods have issues.

**Step 1.** Boot to OOBE. Press **Shift + F10** to open command prompt.

**Step 2.** Run the following commands one at a time:

```
powershell
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned -Force
Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force
Set-PSRepository -Name PSGallery -InstallationPolicy Trusted
Install-Script -Name Get-WindowsAutopilotInfo -Force
Get-WindowsAutopilotInfo -OutputFile C:\autopilot.csv
```

**Step 3.** Copy **C:\autopilot.csv** to a USB drive.

**Step 4.** Manually merge individual CSVs (keep one header row, append data rows).

# 6. MKL Live Progress Tracking

When using the Go-Box or USB method with network connectivity, each device's hash is POSTed to the MKL portal in real-time. This gives you a live dashboard showing fulfillment progress.

## What You See on MKL

- Live counter: **247 / 400 devices processed**
- Each device's serial number and timestamp as it comes in
- Duplicate detection (if a device is accidentally processed twice)
- One-click CSV export when the batch is complete
- Downloadable customer-ready CSV with proper formatting

## API Endpoint

```
POST https://mkl.techtoschool.com/api/autopilot/hash
Content-Type: application/json

{
  "serial": "5CG1234ABC",
  "csv": "Device Serial Number,Windows Product ID,Hardware Hash\n5CG1234ABC,...",
  "timestamp": "2026-03-19T14:30:00-07:00"
}
```

The endpoint accepts the raw CSV content from Get-WindowsAutopilotInfo and stores it in the MKL database. The dashboard page aggregates all received hashes per batch and provides the combined CSV download.

## Offline Fallback

If the device has no internet or the MKL endpoint is unreachable, the USB method still saves hashes to the USB drive locally. You can upload the CSV to MKL manually afterward. The Go-Box method requires network connectivity to be useful (otherwise there's nowhere to store the data since the Go-Box can't write files).

# 7. After Extraction — Preparing the Deliverable

### *Step 1: Verify the Data*

If using MKL tracking, check the dashboard for the final count. If using USB, open **AutopilotHashes/autopilot-devices.csv**. Verify:

- Row count matches the number of devices processed
- Three columns: **Device Serial Number**, **Windows Product ID**, **Hardware Hash**
- No blank rows or duplicate serial numbers
- Hardware Hash column contains a long Base64 string (4,000+ characters per entry)

### *Step 2: Export and Name the File*

Download from MKL or rename the USB CSV:

```
Autopilot-Hashes-[CustomerName]-[YYYY-MM-DD].csv
```

### *Step 3: Send to Customer*

Email the CSV to the customer's IT administrator along with the upload instructions (see Section 8).

# 8. Customer Upload Instructions (Include With Delivery)

Include the following instructions when sending the CSV to the customer. This can be copied into an email:

**How to Register Your Devices in Windows Autopilot**

1. Sign in to the **Microsoft Intune admin center** at intune.microsoft.com

2. Navigate to: **Devices > Enrollment > Windows Autopilot > Devices**

3. Click **Import** and select the CSV file we provided

4. Wait for the import to complete (can take 15-40 minutes for large batches)

5. Verify all devices appear in the Autopilot devices list

6. Assign a **Deployment Profile** to the imported devices (if you haven't already set a default)

7. When a device is powered on and connected to the internet, it will automatically detect your Autopilot profile and begin zero-touch setup

**Note:** Devices must be connected to the internet (Wi-Fi or Ethernet) during first boot for Autopilot to activate. If a device has already been set up, it will need to be reset (Settings > System > Recovery > Reset this PC) to trigger Autopilot.

## 9. Time Estimates by Order Size

| Order Size | Go-Box | USB Script | Manual | Notes |
|---|---|---|---|---|
| 25 devices | ~25 min | ~45 min | ~2 hours | Any method works |
| 50 devices | ~45 min | ~1.5 hours | ~4 hours | Go-Box recommended |
| 100 devices | ~1.5 hours | ~3 hours | ~8 hours | Assembly line + Go-Box |
| 200 devices | ~3 hours | ~5-6 hours | ~16 hours | Two Go-Boxes, two techs |
| 400 devices | ~5-6 hours | ~10-12 hours | ~32 hours | Two Go-Boxes, 1 day |

*Go-Box estimates assume ~45 seconds per device with assembly-line pipelining. First device in each batch takes 2-3 minutes for package downloads. Ethernet connectivity assumed.*

## 10. Troubleshooting

| Issue | Solution |
|---|---|
| Shift+F10 does nothing at OOBE | Try Fn+Shift+F10 on laptops. Some devices disable this — boot from a Windows PE USB instead. |
| Go-Box keystrokes are too fast | Increase delays between steps in the Go-Box template. Watch the first device to calibrate timing. |
| Go-Box opens wrong window | Ensure the device is at the OOBE screen (not past it). The Shift+F10 shortcut only works at OOBE. |
| MKL upload fails (Go-Box) | Device has no internet. Connect Ethernet BEFORE plugging in Go-Box. Check that OOBE network setup is complete. |
| Script says "USB drive not found" | Check the drive letter. Type **wmic logicaldisk get name** to find it. The drive label must be AUTOPILOT. |
| NuGet/PSGallery install fails | Device needs internet. Connect Ethernet or join Wi-Fi first: **netsh wlan connect name="NetworkName"** |
| Get-WindowsAutopilotInfo errors | Ensure you are running as admin. At OOBE the command prompt is already elevated. |
| Hardware hash is empty/short | Device may not have TPM 2.0. Run **tpm.msc** to check. Non-TPM devices cannot use Autopilot. |
| CSV has duplicate entries | Same device was processed twice. MKL dashboard flags duplicates. Remove before sending to customer. |

| | |
|---|---|
| Device stuck at "Just a moment..." | Normal OOBE behavior. Wait 1-2 min. If it proceeds past OOBE, run script from Settings > search "PowerShell" > Run as Admin. |

## 11. Network Connectivity at OOBE

Network connectivity is required for: (1) downloading the extraction script (first device only), and (2) uploading hashes to MKL. For Go-Box method, the device **must** be on the network before plugging in the Go-Box.

- **Ethernet (preferred):** Plug in a cable. OOBE auto-detects wired connections. Fastest and most reliable.
- **Wi-Fi from OOBE:** Complete the Wi-Fi setup step in OOBE before pressing Shift+F10 or plugging in Go-Box.
- **Wi-Fi from command prompt:**

```
netsh wlan show networks
netsh wlan connect name="TTS-Workshop" ssid="TTS-Workshop"
```

- **USB tethering:** Connect a phone via USB with hotspot enabled — Windows picks it up as Ethernet.

*Tip: For large batches, set up a table near an Ethernet switch. Plug each device into the switch before starting. This is the most reliable and fastest configuration.*

## 12. Quick Reference Card

Print this section and keep at the workstation:

**Go-Box Method — Quick Steps**

**1.** Power on device, wait for OOBE screen
**2.** Plug in Ethernet cable
**3.** Plug in Go-Box USB cable
**4.** Walk away — Go-Box does everything
**5.** Device shuts itself down when done
**6.** Unplug Go-Box, move to next device

Check MKL dashboard for live progress count
First device takes ~2-3 min (package download), rest take ~45 sec

**USB Script Method — Quick Steps**

**1.** Power on device, wait for OOBE screen
**2.** Press **Shift + F10** to open command prompt
**3.** Plug in AUTOPILOT USB drive
**4.** Type: **D:\RUN-ME.bat** (or E:\RUN-ME.bat) and press Enter
**5.** Wait for green SUCCESS message
**6.** Remove USB, power off, next device

If Shift+F10 doesn't work: try **Fn+Shift+F10**
If no internet: plug in Ethernet before running script
Drive letter unknown: type **wmic logicaldisk get name**