**Email:** meghanabadgujar07@gmail.com

**Name:** Meghana Samadhan Badgujar  
**Batch Name:** WiproNGA\_DWS\_B5\_25VID2550

**User ID:** 34737

**Batch ID:** 25VID2550

**Date:** 7th August 2025

**Assignment Topic:**

* **Intunewin Conversion – Compatible Version to Upload to Intune**

**- An .intunewin file is a packaged version of a Windows application used for deploying Win32 apps through Microsoft Intune.**

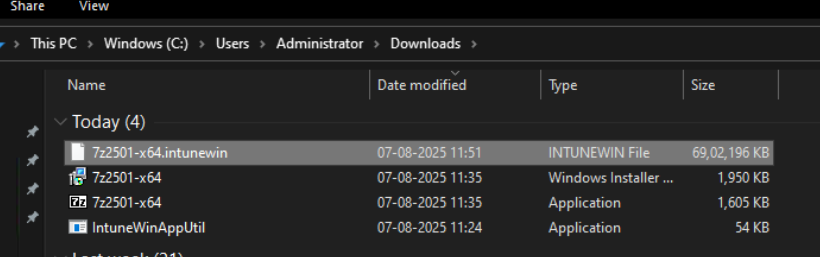
**- It is created using the Win32 Content Prep Tool (IntuneWinAppUtil.exe), which bundles the application's install files and metadata into a format that Intune can understand and deploy silently to client devices**.

**To deploy Win32 applications via Microsoft Intune, the application must be converted into a .intunewin format using Microsoft’s Win32 Content Prep Tool.**

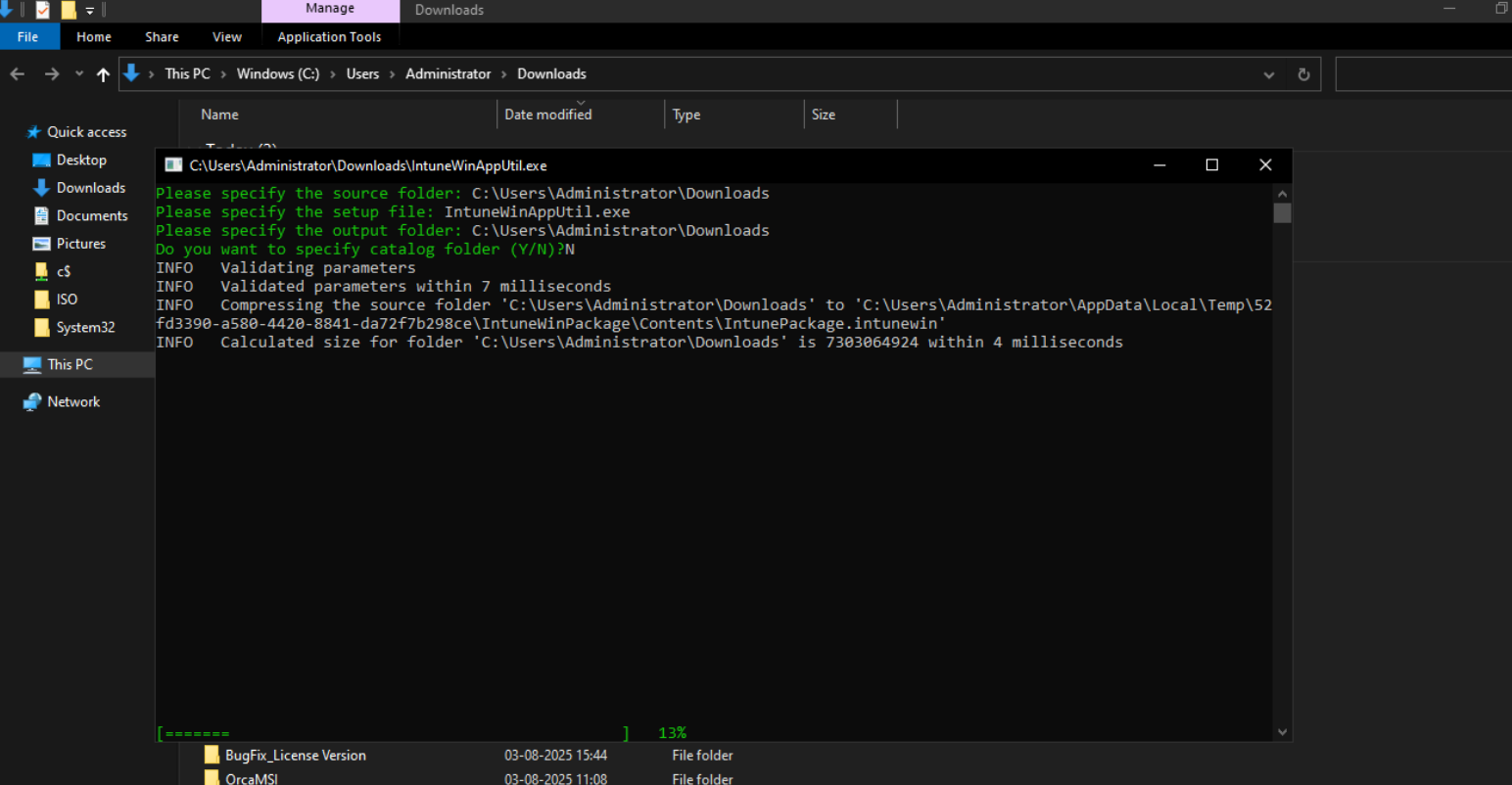
**Steps to Create a .intunewin File:**

1. **Prepare Application Files:**Collect all the necessary setup files for installation (e.g., setup.exe, .msi, etc.).
2. **Download and Run IntuneWinAppUtil.exe:**This tool is available from Microsoft. Run it via command prompt and provide:

* Source folder
* Setup file (e.g., setup.exe)
* Output folder (where .intunewin will be saved)

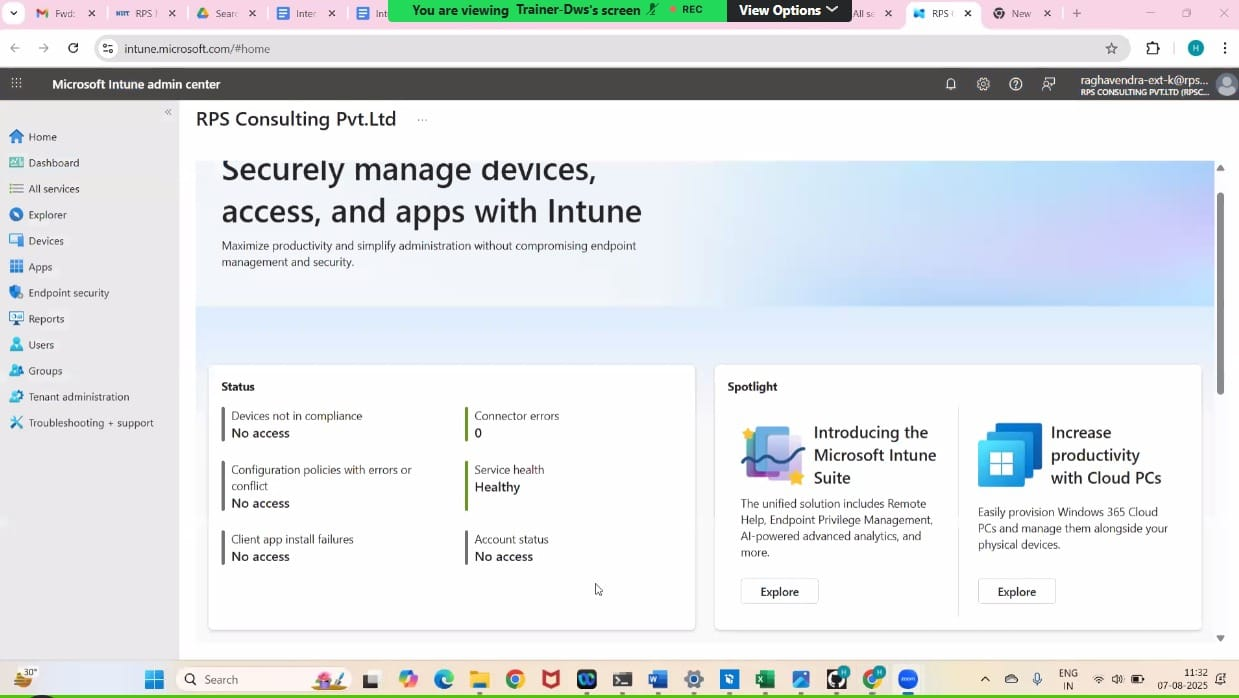


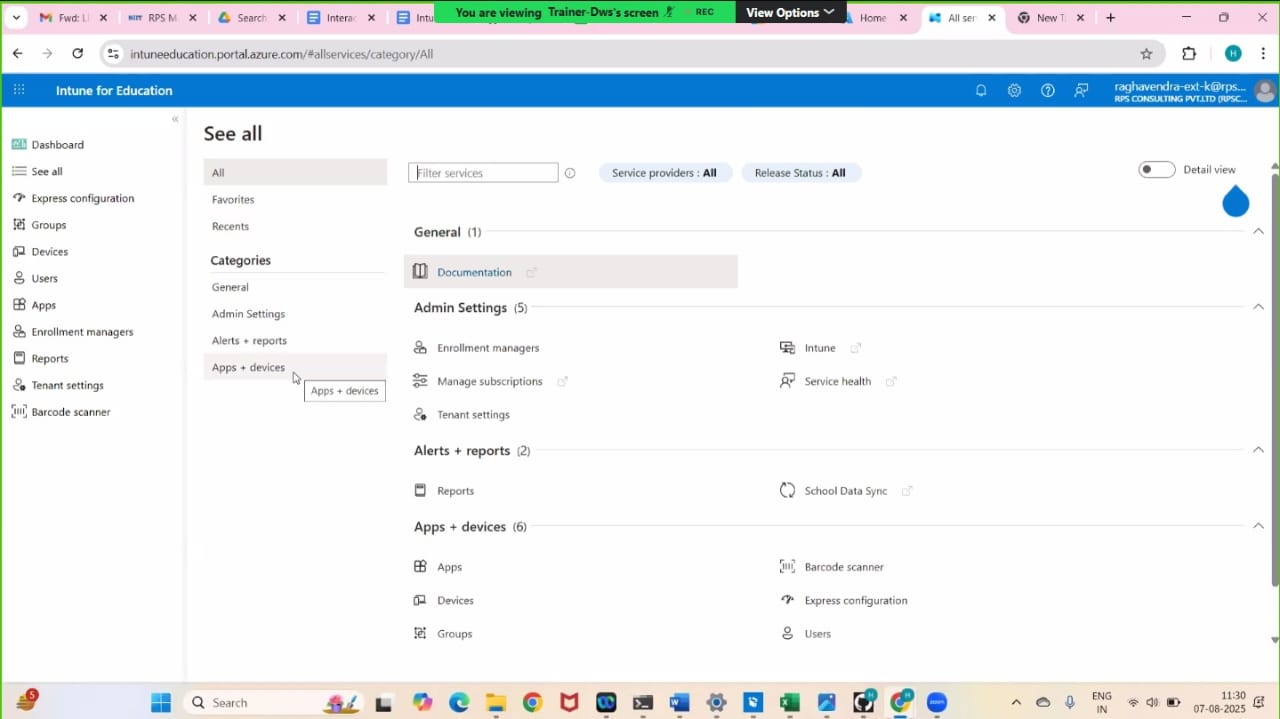
1. **Conversion Output:**The tool packages the app into a .intunewin file that includes metadata and binaries.

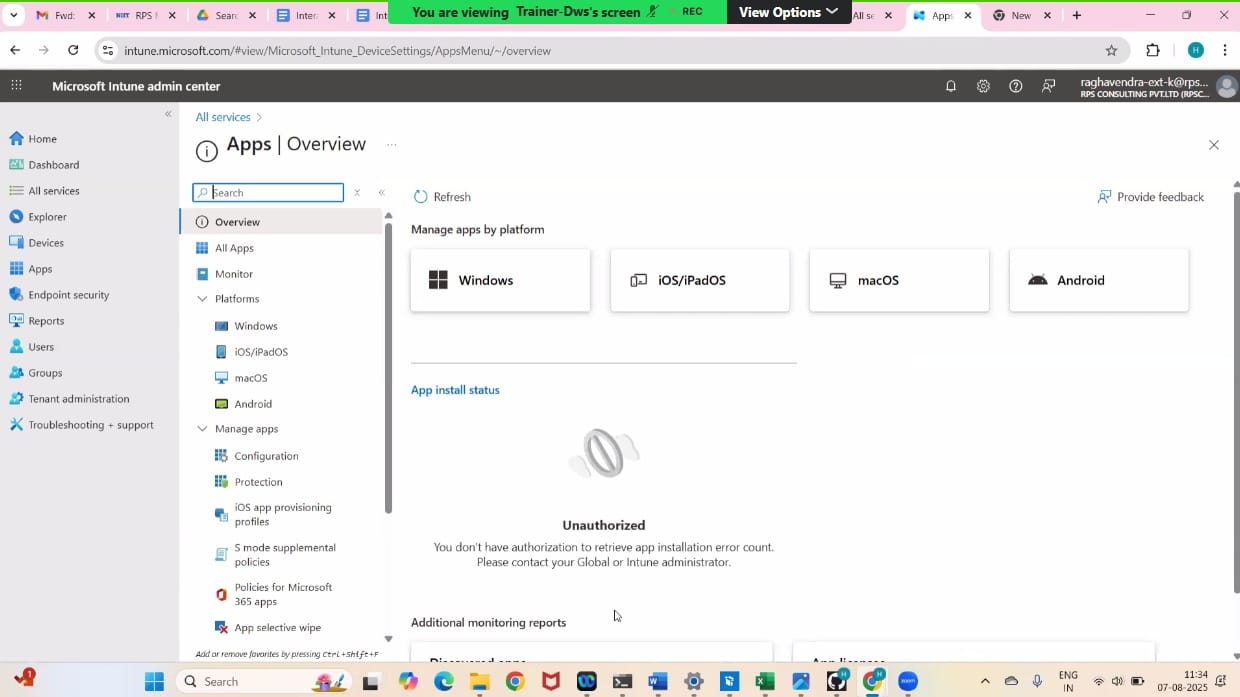
****

1. **Upload to Intune Admin Center:**

* Go to Apps > Windows > Add App
* Choose App Type: Windows app (Win32)
* Upload the .intunewin file
* Specify install/uninstall commands and detection rules







* **Interactive and Non-Interactive Applications**

1. **Interactive Applications:**

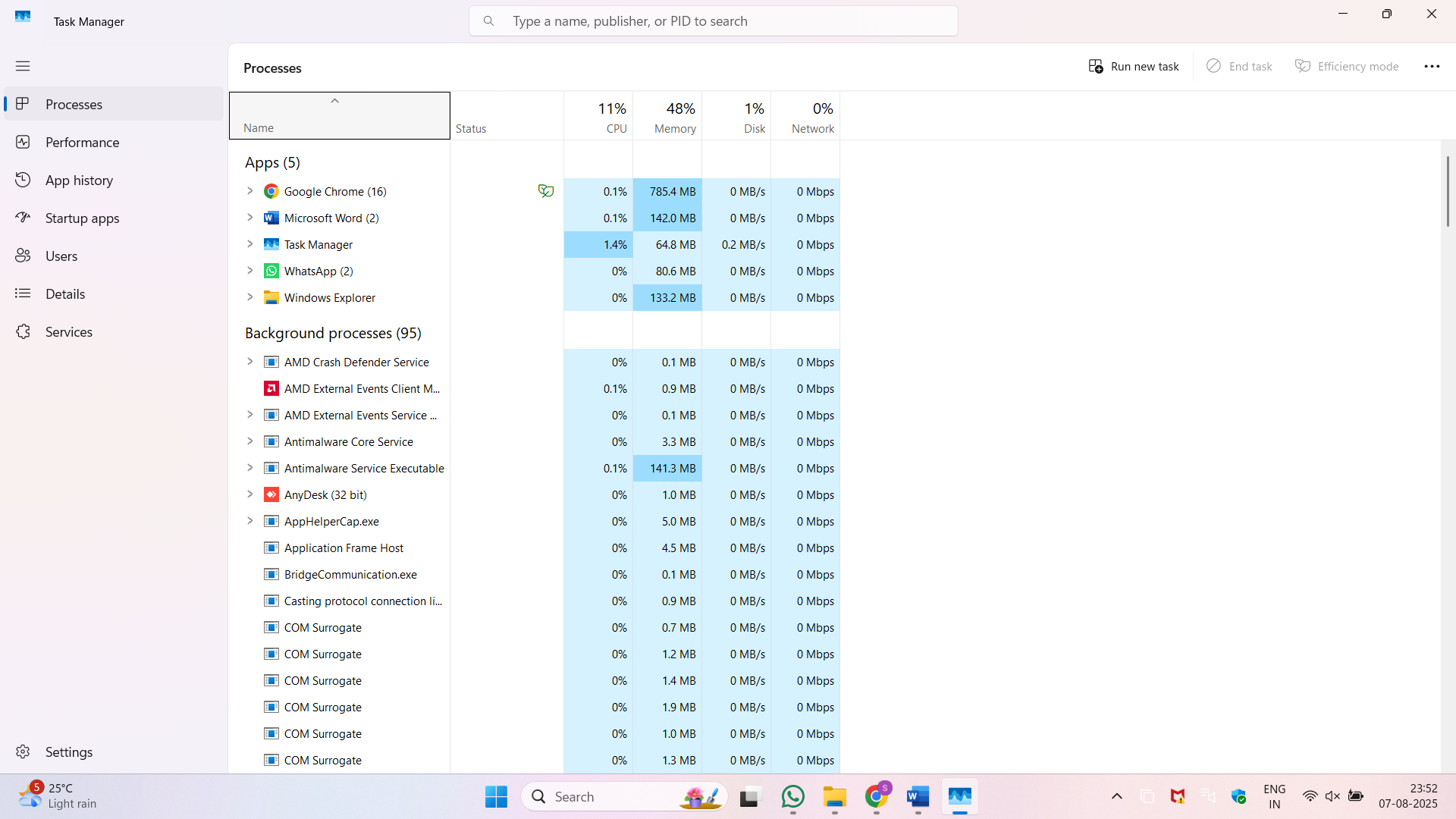
Interactive applications require user interaction during execution. They are designed with Graphical User Interfaces (GUI) to enable real-time user input such as clicking buttons, filling forms, or selecting options.

* Require **user input** (e.g., mouse clicks, keyboard entry)
* Have a **GUI (Graphical User Interface)**
* Examples: MS Word, VLC Media Player, Web Browsers

1. **Non-Interactive Applications:**

Non-interactive applications run in the background without needing any user input during their execution. They usually perform automated or scheduled tasks.

* Run in the **background** without user interaction
* Often **do not have GUI**
* Examples: Antivirus scans, print spoolers, update services



**Interactive App**

**Non-Interactive App**

* **Required vs. Available App Assignments in Intune**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Assignment Type** |  |  | | --- | |  | | | **Behavior** | | --- |  |  | | --- | |  | |
| |  | | --- | | **Required** |  |  | | --- | |  | | App is automatically installed on targeted devices/users without user interaction |
| |  | | --- | | **Available** |  |  | | --- | |  | | App appears in Company Portal; users can install it on-demand |

* **Required = Silent deployment**
* **Available = User-driven deployment**
* **Process Flow for an Application on Windows Client via IME Service**

**When a Win32 application (.intunewin) is deployed through Microsoft Intune, the Intune Management Extension (IME) is responsible for handling the full lifecycle of that application on a Windows client.**

Here’s a step-by-step breakdown of the application flow:

**1. Polling Phase**

* The IME agent installed on the client machine polls the Intune service every 60 minutes (by default).
* It checks for:
  + New applications assigned to the user or device
  + Changes to existing application assignments
  + Updates or version changes
* This ensures the device stays in sync with Intune policies.

**2. Pre-Installation Detection**

* Before installing, IME uses detection rules defined in the Intune portal to check if the app is already present.
* Detection methods can include:
  + File existence (e.g., check if a .exe file exists)
  + Registry entries
  + Product code (for MSI-based apps)
* If detection indicates the app is already installed, installation is skipped.

**3. Installation Phase**

* If the app is not detected, IME begins the install process:
  + Downloads the .intunewin package from the Intune cloud
  + Unpacks and extracts it into a temporary folder
  + Executes the Install Command provided (e.g., msiexec /i app.msi /qn)
* The installation runs in either user or system context (as defined).
* If the installer takes too long, exceeds timeout, or returns an error code, the status is marked as failed.

**4. Post-Installation Detection**

* After the install completes, IME runs the same detection rules again.
* This step verifies whether the installation was successful.
* If detection passes, the app is considered successfully installed.
* If it fails, it's marked as a failed deployment.

**5. Toast Notifications**

* Based on the result, the user receives a toast notification:
  + App installed successfully
  + App failed to install
* These notifications can include:
  + Text (status message)
  + Icons
  + Optional action buttons (like retry or open)
* A device reboot may be required after installation if:
  + The app installer needs it
  + It's defined in the Intune deployment settings
* **Log Files and What They Include**

**Log files are system or application-generated files that record events, actions, and errors. They are critical for troubleshooting, auditing, and debugging.**

* **Key Components of Log Files:**

1. **Timestamps**
   * Shows the exact date and time when each event occurred
   * Helps track the sequence of events
2. **Event Type / Action**
   * Describes what happened (e.g., Installation Started, Detection Failed, User Login)
   * Helps identify the nature of the event
3. **Severity Level**
   * Indicates the importance or criticality of the event:
     + Informational, Warning, Error, Critical, etc.
4. **Descriptions / Messages**
   * Brief text explaining the event
   * Often includes error codes, status messages, or actions taken
5. **Event IDs / Codes**
   * Unique identifiers assigned to specific events
   * Useful for searching and filtering logs
6. **Source or Component Name**
   * Shows which service or module generated the log (e.g., IntuneManagementExtension, MSI Installer)
7. **User Context**
   * Indicates whether the action was performed by a user or system account
8. **Result or Outcome**
   * Indicates success, failure, retry, or timeout status
   * Often shown with codes like Exit Code 0 (Success) or Exit Code 1603 (Failure)
9. **File Paths / Registry Keys** 
   * Points to related files, folders, or registry locations involved in the event
10. **System or Device Info**
    * May include hostname, OS version, user ID, or IP address
    * Helps in identifying the affected device

* **Registry Entries for Application GUID and Status**

Each Win32 app deployed via Intune has a **unique Application GUID**.

**Status Tracking Registries:**

Located at:

HKLM\SOFTWARE\Microsoft\IntuneManagementExtension\Win32Apps\{App GUID}

You can find:

* **Installation Status** (Succeeded, Failed, etc.)
* **Exit Codes**
* **Detection Results**
* **Execution Context** (user/system)
* **Registries with Respect to LOB and Win32 Apps**

**Key Registry Paths for Installed Apps:**

* **System-wide (All Users):**  
  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
* **User-specific:**  
  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

These keys list details like:

* Display Name
* Install Location
* Uninstall Command
* Product GUID

