**WIPRO ASSIGNMENT – 3**

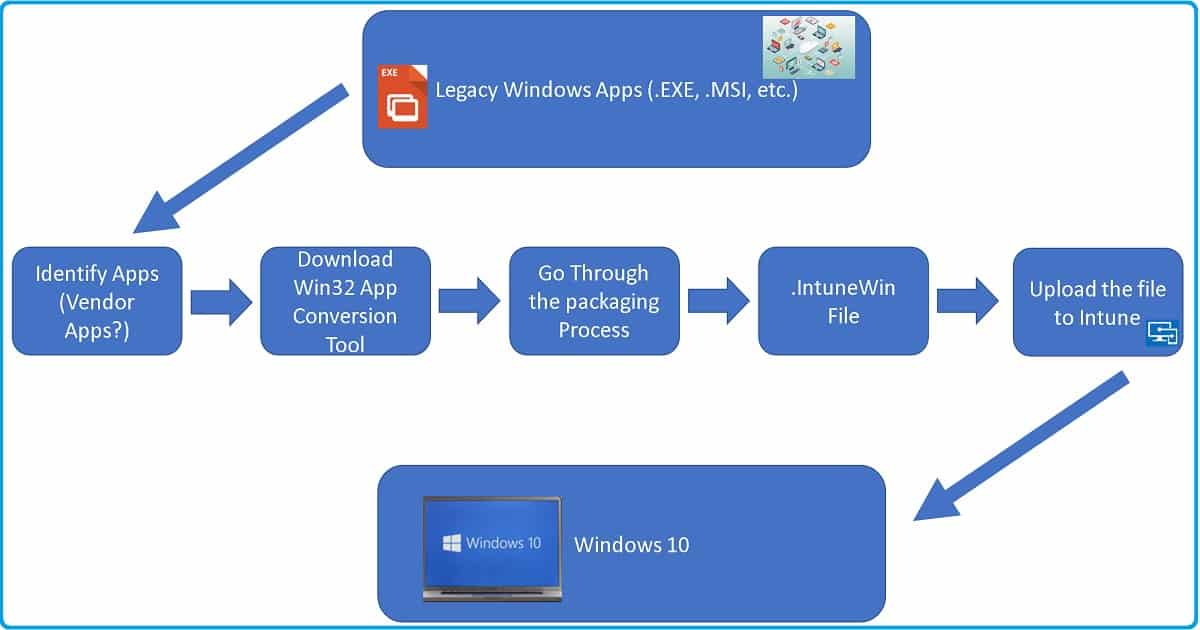
* **Submitted By:** Anuj Bose
* **User ID:** 34757
* **Batch Name:** WiproNGA\_DWS\_B5\_25VID2550
* **Date:** 7th August, 2025
* **Email ID:** anujbose01012004@gmail.com

**Intune Conversion – Compatible Version To Upload To Intune**

To deploy a Win32 app via Intune, you need to use the Microsoft Win32 Content Prep Tool, which converts your application's installation files into a .intunewin format that Intune can deploy. The minimum supported operating system for this deployment is Windows 10 version 1607 or later. Additionally, target devices must be enrolled in Intune through methods such as Microsoft Entra join, hybrid domain join, or Group Policy enrollment.

Here's a breakdown of the process:

* **Prepare your application**: Gather all necessary installation files (e.g., .exe, .msi, config files) into a single folder.
* **Run the Win32 Content Prep Tool**: Open Command Prompt, run IntuneWinAppUtil.exe, and follow the prompts to specify the source folder, setup file, and output location.
* **Generate the .intunewin file**: The tool creates a .intunewin package containing the app files and metadata needed for Intune deployment.
* **Upload to Intune**: Go to the Intune admin center, choose **Apps > Windows > Add**, select **App type: Windows app (Win32)**, and upload the .intunewin file for deployment.



**Important Considerations:**

* **File Size:** Ensure your package size doesn't exceed 8 GB.
* **Silent Installation:** The tool will likely require you to provide the silent installation command-line arguments for the application.
* **Uninstall Command:** You'll need to provide the command used to uninstall the application.
* **Minimum Windows Version:** Your devices need to be running Windows 10 version 1607 or later.
* **Intune Management:** Devices need to be enrolled in Intune management.

**Interactive And Non-Interactive Applications**

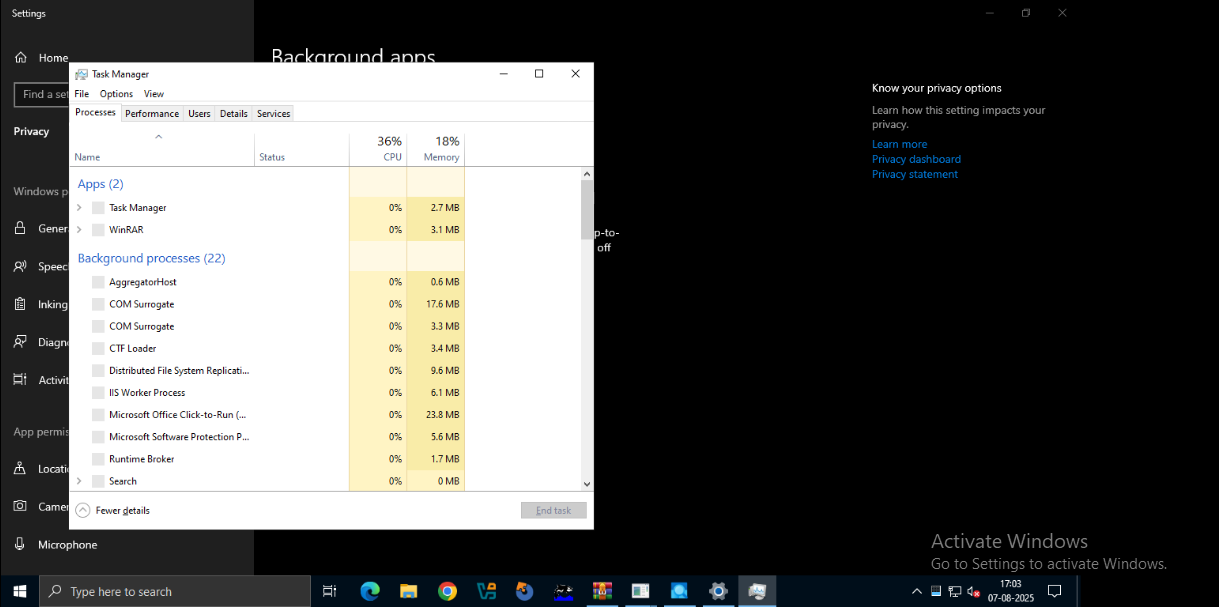
In Windows, interactive applications are programs that require user input and typically offer a graphical user interface (GUI), allowing users to click, type, and interact—for example, web browsers or word processors. In contrast, non-interactive applications run in the background without any user interaction or visible interface. These are often system services or tasks like software updates or printing that operate automatically.

**Interactive Applications**

* **User Interaction:** Require user input like clicks, typing, or menu selection.
* **GUI:** Feature a graphical interface for visual interaction.
* **Examples:** Web browsers, word processors, media players, and games.

**Non-Interactive Applications**

* **Background Execution:** Run without a visible GUI, performing tasks automatically in the background.
* **Services:** Often implemented as Windows services handling tasks like printing or network management.
* **Automated Tasks:** Commonly used for updates, virus scans, backups, and other routine operations.
* **Examples:** System services, scheduled tasks, and background data processing.



**Required And Available App Assignments**

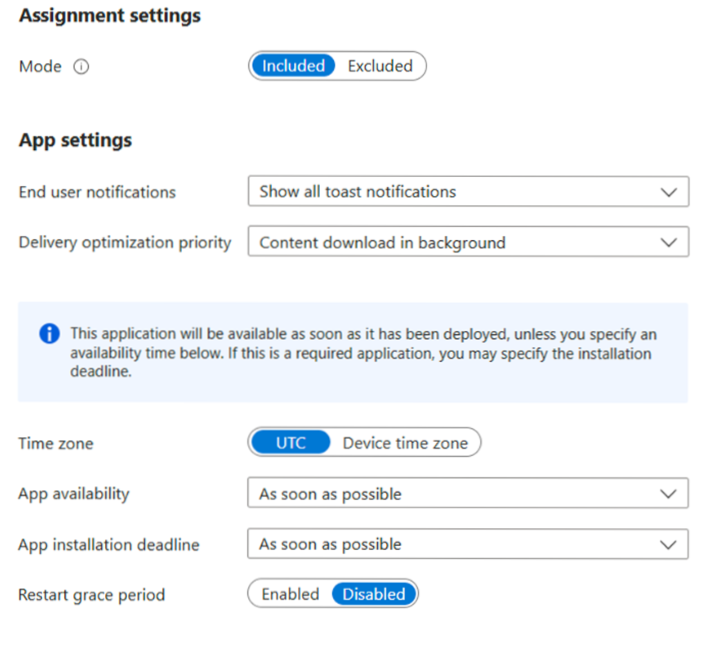
When you set an application's assignment, you're setting whether the app will be **Required**, **Available for enrolled devices**, **Available with or without enrollment**, or **Uninstalled**. Each of these options has additional suboptions, such as availability, installation deadline, and restart grace period.

**Required Assignments**

* **What it means:** The app installs automatically on assigned devices or for targeted users without requiring any user action.
* **Use case:** Ideal for critical or essential apps (e.g., antivirus, VPN, Microsoft Word) that should be present on all managed devices.
* **Behavior:**
  + Installs silently in the background.
  + Can be scheduled with a specific install time or deadline.
  + Installation progress can be monitored through Intune reports.
* **User experience:** No interaction needed; the app appears already installed when the user accesses the device.

**Available Assignments**

* **What it means:** The app is optional and made available in the Company Portal for users to install manually when needed.
* **Use case:** Suitable for non-essential or optional apps like developer tools, training software, or department-specific applications.
* **Behavior:**
  + The app is not auto-installed.
  + Users can browse the Company Portal and install it at their convenience.
* **User experience:** Users open the Company Portal, locate the app, and choose to install it whenever they need it.



**Process Flow for an Application on Windows client via IME service.**

**1. Polling**

* IME (Intune Management Extension) regularly checks Intune servers for new or updated app assignments.
* Keeps the client device in sync with deployment changes.

**2. Detection**

* **Before Installation:** Detection rules verify if the app is already installed.
* **After Installation:** Rules are re-checked to confirm successful setup.
* **Rule Types:** Can be based on registry entries, file presence, or running processes.

**3. Installation**

* If not present, IME begins the installation process.
  + **Download & Unpack:** .intunewin file is downloaded and extracted to a staging folder.
  + **Command Execution:** Installs the app using admin-defined commands (e.g., msiexec).
  + **Monitoring:** Installation is tracked; timeouts result in failure.

**4. Detection & Notifications (Post-Install)**

* **Re-detection:** Confirms the app is properly installed.
* **Notifications:** Users get toast messages indicating success or failure.
* **Toast Customization:** Supports icons, custom text, and actions.
* **Reboot (if needed):** Device may restart based on Intune policy for changes to apply.

**Registries with respect to LOB and Win32Apps**

In Microsoft Intune, the Windows Registry is used to deploy and manage LOB and Win32 applications. It is also a hierarchical Windows configuration information database.

1. **Windows Registry Role**

* Saves application and system settings, preferences and configurations.
* Structures data using keys: folders and values: entries.

1. **LOB Apps**

* Line-of-business applications.
* Use the registry to:
* Save preferences and settings.
* Maintain app functionality across devices.
* Allow Intune configuration management.

1. **Win32 Apps**

* Traditional Windows desktop applications.
* Depend greatly on the registry for:
* Track installation progress.
* Config management (store config data).
* Manage app setting.

1. **Intune (manage both LOB and Win32 apps)**

* Intune employs registry information to:
* Ensure an application is installed (according to detection rules).
* Enforce specific app configurations.

1. **Registry Editing with Caution**

* Manual editing must be done carefully.
* Incorrect changes can lead to instability (of system) or app problems.

Windows Registry is critical to LOB and Win32 application management in Intune for detection, validation and configuration control support for secure enterprise app deployment. Change Language and summarize in points

**Specific Registries with Application GUID which give you the status of Installation / Uninstallation.:**The Windows Registry stores key details about installed software, such as unique identifiers (GUIDs) used to verify installation status or assist with removal. This is especially valuable when managing **Line-of-Business (LOB)** or **Win32 applications** using Microsoft Intune or manual MSI processes.

**i. Common Registry Paths for Applications**

* **Per-machine installs** (available to all users):  
  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
* **Per-user installs** (specific to the current user):  
  HKCU\SOFTWARE\Microsoft\Windows\Uninstall

**ii. GUID (Globally Unique Identifier)**

* Each application entry has a **GUID**, a 32-character unique string.
* This GUID appears as the **sub-key** under the Uninstall registry path.
* It helps determine if the app is present or to trigger its removal.

**iii. Using GUID for Uninstallation**

* You can uninstall a program by running the following command:  
  msiexec.exe /x {GUID} /QN
* **Parameters explained:**
  + /x {GUID}: Uninstalls the application using its unique identifier
  + /QN: Performs the uninstallation silently (no user interaction)
  + /L\*V "C:\Client-uninstall\desktop-uninstall.log": (Optional) Saves a detailed log of the process
* **Example:**  
  If the registry shows {80890A63-01AA-40D3-A2E9-B3E214735151}, then:  
  msiexec.exe /x {80890A63-01AA-40D3-A2E9-B3E214735151} /QN

**iv. Log Files (Optional but Helpful)**

Windows Event Logs can be used to confirm whether an app was successfully installed or removed. These logs contain:

* **Timestamps** – Show exactly when each event took place.
* **Event Types** – Define whether it’s an error, warning, info, or audit success/failure.
* **Severity Levels** – Indicate the seriousness of the event (e.g., critical, error, info).
* **Descriptions** – Offer insights like error codes, affected systems, or user actions.
* **Event IDs** – Unique numbers tied to specific types of events, useful for quick searches.

