**Name**: Sajiya

**User ID:** 34764

**Email ID:** sajiyadl888@gmail.com

**Practice Activity 3: Assignment**

**Interactive and Non-Interactive Applications, Intunewin Conversion, IME Process Flow, and Registries for LOB and Win32 Apps**

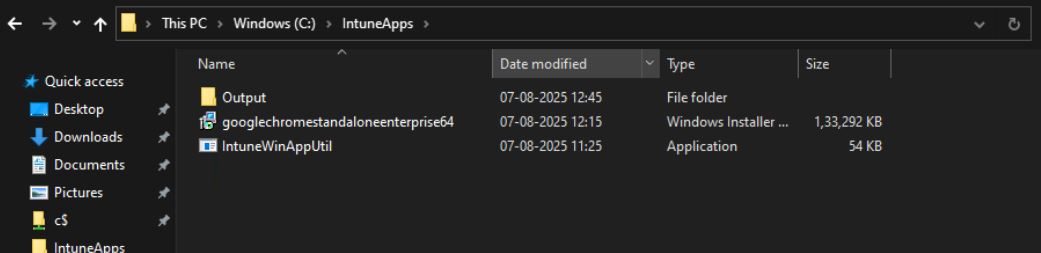
**1. Interactive and Non-Interactive Applications**

Some applications are meant to be used directly by a person — like when you're clicking buttons, typing into forms, or making choices. These are called **interactive applications**. They show a visible window and wait for you to do something, like opening a browser, playing music, or installing a game.

On the other hand, there are apps that run by themselves in the background and don’t need any input from the user. These are called **non-interactive applications**. They don’t pop up or ask for anything — they just do their job quietly, like checking for software updates or scanning for viruses.

In short, interactive apps ask for your attention, while non-interactive ones just do their work silently.

**2.** **Intunewin conversion - Compatible version to upload to Intune**



To deploy a Win32 app through Intune, the first thing to do is collect all the files needed to install the app. This includes the main installer file like .exe or .msi, and any other support files if required. After that, Microsoft provides a tool called **IntuneWinAppUtil.exe**, which is part of the Win32 Content Prep Tool. Open Command Prompt, run this tool, and it will ask for some basic information like the folder path where your installer is saved, the setup file name, and the output folder. Once you enter all that, the tool will create a new file with the .intunewin extension. This file is ready to be uploaded to the **Intune admin center**. In Intune, go to the **Apps** section, choose **Windows apps (Win32)**, and upload your .intunewin file. You’ll also need to type in the install and uninstall commands — usually in silent mode so it runs without bothering the user (like using /quiet or /silent switches). Also, make sure the app size is not more than **8 GB**, and it will only work on devices running **Windows 10 version 1607 or newer** that are already enrolled in Intune.

**Process Flow for an Application on Windows Client via IME Service**

When an application is deployed using Intune, the **Intune Management Extension (IME)** handles the entire installation process on the user’s Windows device. It begins with **polling**, which means the device regularly checks with the Intune service to see if any new apps or policies are assigned. If a new app is found, the IME service uses **detection rules** to check if the app is already installed. These rules can look for specific files, folders, registry keys, or MSI product codes. If the app isn’t found, IME proceeds to **download and install** the app using the command given by the admin, usually in silent mode. After the installation finishes, the detection rules are run **again** to make sure the app was installed successfully. If everything is fine, the device reports success back to Intune, and a **toast notification** (a small pop-up) appears on the screen to inform the user whether the app was installed successfully or if it failed. In some cases, a **restart** might be required to complete the setup, depending on the app and policies applied. This process helps Intune manage app deployment smoothly and automatically.

**Registries with Respect to LOB and Win32 Apps**

In Intune, both **LOB (Line-of-Business)** apps and **Win32** apps often use the **Windows Registry** to store or check important information about the app. The registry is like a hidden database in Windows where apps and the system save settings, such as install status, configurations, and user preferences. When deploying an app through Intune, detection rules can be set to look for specific **registry keys or values** to confirm whether the app is already installed. For example, if a registry key says that version 1.0 of the app is installed, Intune knows it doesn’t need to install it again. LOB apps, which are usually custom apps for an organization, and Win32 apps, which are packaged using .intunewin, both make use of registry paths. Admins can also configure registry entries during or after installation to apply special settings. Since the registry controls how apps behave and how the system responds, making changes to it should always be done carefully. Mistakes in the registry can cause apps to misbehave or even affect the Windows system itself.