Intune Application Packaging and Deployment – Full Training Guide

This guide is designed to teach you everything about application packaging and deployment using Microsoft Intune, starting from scratch. Each topic includes theory, real-world context, examples, hands-on steps, and logs to observe.

---

1. Intune Free Trial Subscription – How to Enroll

📚What is Intune?

Microsoft Intune is a **cloud-based service** that helps organizations manage devices and apps securely. It's part of Microsoft Endpoint Manager.

🩲Why Use Intune?

- Manage apps and updates
- Secure company data
- Apply policies to devices

🛠️Steps to Enroll in Free Trial:

1. Visit: https://www.microsoft.com/en-us/microsoft-365/business/microsoft-intune
2. Click **"Try for free"**
3. Sign in with your **Microsoft account** (or create one)
4. Complete company name and admin info
5. Setup your **Microsoft 365 admin center**
6. Go to **Microsoft Endpoint Manager Admin Center**: https://endpoint.microsoft.com

   💡 Tip: Use a test domain (like "TestOrg.onmicrosoft.com")

🧪Try It:

- Enroll one VM or test laptop to Intune.
- Explore: Devices > Windows > Enrollment > Windows Enrollment.

---

2. Walkthrough over Intune Applications (Windows)

🗂️Types of Applications in Intune:

App Type	Description
LOB	Line of Business – native .msi apps
Win32	Complex apps (.exe, zipped, custom scripts)
Microsoft Store Apps	Install directly from Microsoft Store
Web Link	Add a link like an app shortcut

🔎Where to Find Them:

- Go to **Apps > Windows > Add**
- Choose the type: Win32, LOB, Microsoft Store

🧪Try It:

- Click **Apps > Windows > Add**
- Choose **App Type: Line-of-business app**
- Upload any `.msi` installer

---

3. LOB and Win32 Apps

🧰What's the Difference?

Feature	LOB (.msi)	Win32 (.exe/.zip)
Format	.msi	.exe, zipped folders
Complexity	Basic installation	Advanced logic (scripts)
Packaging	Upload directly	Must convert to `.intunewin`

📌Real-World Examples:

- LOB: 7-Zip `.msi` installer
- Win32: Adobe Reader with custom silent install

🧪Hands-On:

1. Download `.msi` of 7-Zip from: https://www.7-zip.org
2. Add it as **LOB App** to Intune.
3. Download `.exe` of VLC Player.
4. Package VLC as **Win32** using IntuneWinAppUtil.

---

4. Intunewin Conversion (Packaging EXE into Intune Format)

🧰Tool Used:

IntuneWinAppUtil.exe from Microsoft.

🧪Steps:

1. Download tool: https://github.com/Microsoft/Microsoft-Win32-Content-Prep-Tool
2. Create a folder `C:\AppSource\VLC`
3. Put your `vlc.exe` there
4. Create `install.cmd` (with silent install command)
5. Run:

```
IntuneWinAppUtil.exe
```

1. Input:
2. Source folder: `C:\AppSource\VLC`
3. Setup file: `install.cmd`
4. Output folder: `C:\Output`
5. It creates: `vlc.intunewin`

🔗Upload this to Intune as a **Win32 App**

---

5. Interactive vs Non-Interactive Applications

🧪What It Means:

- **Interactive**: Needs UI/user input
- **Non-Interactive**: Silent install, no user needed

🔗In Intune:

- Always prefer **silent (non-interactive)** apps for remote deployment

  Example:

```
install.cmd:
start /wait vlc.exe /S
```

---

6. App Assignments (Required vs Available)

Assignment Type	Behavior
Required	Automatically installs on target devices
Available	Shows in Company Portal for user to install manually

🧪Hands-On:

- Assign 7-Zip (LOB) as **Required** to a user group
- Assign VLC (Win32) as **Available** to same group

---

7. Groups, Dynamic Queries, Users

Types of Groups:

- **Assigned Groups** – Manually add users/devices
- **Dynamic Groups** – Auto-populated based on rules

🧪Example Dynamic Query:

```
(device.deviceOSType -eq "Windows") and (device.deviceOSVersion -contains "10")
```

Use it in Azure AD > Groups > New Group > Dynamic Membership Rules

---

8. IME Process Flow (App install)

📦Intune Management Extension (IME)

- Background service on client devices that handles **Win32 app installations**.

🔦Flow:

1. Device syncs with Intune
2. IME downloads app content
3. Runs install command (e.g. install.cmd)
4. Checks detection rules
5. Logs result

📍 Path: `C:\Program Files (x86)\Microsoft Intune Management Extension`

---

9. Registries for Win32 and LOB

Why Registry?

Used for:

- **App detection** (is it already installed?)
- **App removal commands**

🧪Example Detection Rule:

Check if registry key exists:

```
HKEY_LOCAL_MACHINE\SOFTWARE\7-Zip
```

---

10. GUID-specific Registry Keys

📌Each Intune app has its own GUID under registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IntuneManagementExtension\Win32Apps\{App-GUID}
```

- Tracks install status, error codes, etc.

Use `regedit.exe` to explore

---

11. Log File Locations

📝Important Logs:

Path	Purpose
`C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementExtension.log`	Main app install log
`C:\Windows\CCM\Logs`	If co-managed with SCCM

Use Notepad++ or CMTrace to read logs

---

12. Company Portal

🪧What is It?

An app users install to:

- View available apps
- Install self-service apps
- See compliance status

🧪Install:

- Download from Microsoft Store
- Sign in with work account

---

13. Syncing Device and Company Portal

🕐When to Sync:

- New app published
- Policy changed
- User reports delay

🧪Sync Steps:

- Settings > Accounts > Access work or school > Select > Info > Sync
- Or use Company Portal > Settings > Sync

---

14. Log File Breakdown

🔍What to Look For:

Open: `IntuneManagementExtension.log`

- `Starting download for app` – Download begins
- `Running install command line` – Command executing
- `Detection rule passed` – App installed successfully
- `Exit code` – Use this to find errors

🖍️Common Exit Codes:

- 0 – Success
- 1 – General failure

- 1603 – MSI install error (check command)

PowerShell for Application Packaging and Management – Full Beginner Guide

This section will teach you the **basics and advanced use** of PowerShell commands (cmdlets) that are critical in application packaging, deployment, and system management.

Each topic is simplified with examples and hands-on practice steps.

1. Introducing to Cmdlets

📌What is a Cmdlet?

A **cmdlet** is a lightweight PowerShell command used to perform specific tasks in Windows.

Example:

```
Get-Process
```

This shows all running processes.

🧪Try It:

- Open PowerShell
- Type `Get-Service` → Lists all services

2. The PowerShell Pipeline

🕐What is a Pipeline?

The pipeline (`|`) lets you pass the output of one cmdlet to another.

Example:

```
Get-Process | Where-Object {$_.CPU -gt 100}
```

This shows only the processes using more than 100 CPU time.

3. Key Cmdlets

Cmdlet	Purpose
`Get-Process`	List running processes
`Get-Service`	List services
`Set-Service`	Start, stop, or pause a service
`Get-EventLog`	View Windows logs
`Stop-Process`	Kill a process

🧪Practice:

```
Stop-Process -Name notepad
```

4. WMI & PowerShell

📚What is WMI?

Windows Management Instrumentation (WMI) lets you query and manage Windows components.

🔧Example:

```
Get-WmiObject Win32_OperatingSystem
```

Shows OS info like version, build, etc.

5. Pipeline Filtering & Operators

🕗Common Filters:

```
Where-Object {$_.Status -eq "Running"}
```

Finds services that are running.

6. Input, Output & Formatting

📤Output Formatting:

```
Get-Process | Format-Table Name,CPU -AutoSize
```

This shows a clean table of process names and CPU usage.

7. Scripting Overview

What is a PowerShell Script?

A `.ps1` file with a set of commands.

Example:

```
# save as Hello.ps1
Write-Host "Hello from PowerShell"
```

Run it using:

```
.\Hello.ps1
```

8. Objects, Arrays, Variables

🕐Variables:

```
$name = "Wipro"
```

📚Arrays:

```
$apps = @("7zip", "VLC", "Zoom")
```

Loop through:

```
foreach ($app in $apps) { Write-Host $app }
```

9. Scope

🔒What is Scope?

Scope controls where a variable is visible (inside a function, script, etc.)

```
function test {
   $localVar = "inside"
   Write-Host $localVar
}
test
Write-Host $localVar  # error, not visible outside
```

10. More Operators

- `-eq` → equal to
- `-gt` → greater than
- `-like` → pattern match

```
if ($name -eq "Wipro") { Write-Host "Selected" }
```

11. Scripting Constructs

🔦Loops:

```
for ($i=0; $i -lt 5; $i++) { Write-Host $i }
```

🕯️ If/Else:

```
if ($a -gt 10) { Write-Host "Big" } else { Write-Host "Small" }
```

12. Modularization

📦What is a Module?

A file or folder with reusable functions.

Create your own:

```
New-ModuleManifest -Path MyModule.psd1 -RootModule MyModule.psm1
```

Then import:

```
Import-Module .\MyModule.psd1
```

🔗 This concludes your PowerShell Advanced Briefing. You're now equipped to write scripts, manage services, and interact with WMI and Intune with PowerShell!

Would you like a PDF export, mock test, or interview prep based on this?