

## Self-Repair

### Meaning:

Self-repair is a feature of MSI-based installers (Windows Installer) that automatically fixes missing or corrupted application components when a user launches the app.

## Self-Repair (Self-Healing)

- MSI checks KeyPath files at launch
- If missing, Windows Installer repairs
- Triggered by advertised shortcuts, COM errors, etc.
- Ensures application integrity automatically

## How Self-Repair Works

1. App shortcut launches
2. MSI verifies KeyPath files/registry
3. If missing → triggers repair
4. Files restored from MSI cache



Tip: Use KeyPath only for essential components

## Active Setup

### Meaning:

Active Setup is a Windows mechanism that allows certain user-specific

### Role in Packaging:

Used to install user profile components (like HKCU registry ent

## Active Setup - Overview

- Used to apply user-specific settings
- Runs once per user at login
- Common in enterprise & VDI environments



Good for setting HKCU keys or copying user profile files

## Active Setup - How It Works

1. Installer writes to HKLM Active Setup
2. At login, Windows compares with HKCU
3. If GUID missing → executes StubPath command
4. Adds settings to user profile

## Excel Add-ins

### Meaning:

Excel Add-ins are extensions that add features to Excel, typically via .xla, .xlam, or .dll files.

## Excel Add-ins

 Types:

- VBA Add-ins (.xlam, .xla)
- COM Add-ins (.dll)
- Automation Add-ins (via registry)

Used to extend Excel functionality in enterprise apps



## Packaging VBA Add-ins

### Steps:

1. Copy to AppData or Program Files
2. Modify registry:  
HKCU\...\Excel\Options  
Value: /R path\addin.xlam

## Packaging COM Add-ins

### Steps:

1. Copy .dll to location
2. Register with regsvr32
3. Add registry:  
HKLM\...\Excel\Addins\YourAddin  
LoadBehavior = 3

## Key Points for Packaging

File Placement: Copy to %ProgramFiles%, %AppData%, or %LocalAppData%.

Registration (usually via registry):

COM Add-ins:

HKLM\Software\Microsoft\Office\Excel\Addins\{AddInName}

Excel Options:

HKCU\Software\Microsoft\Office\Excel\Options

Silent Installation:

Ensure the add-in is registered automatically without user input.

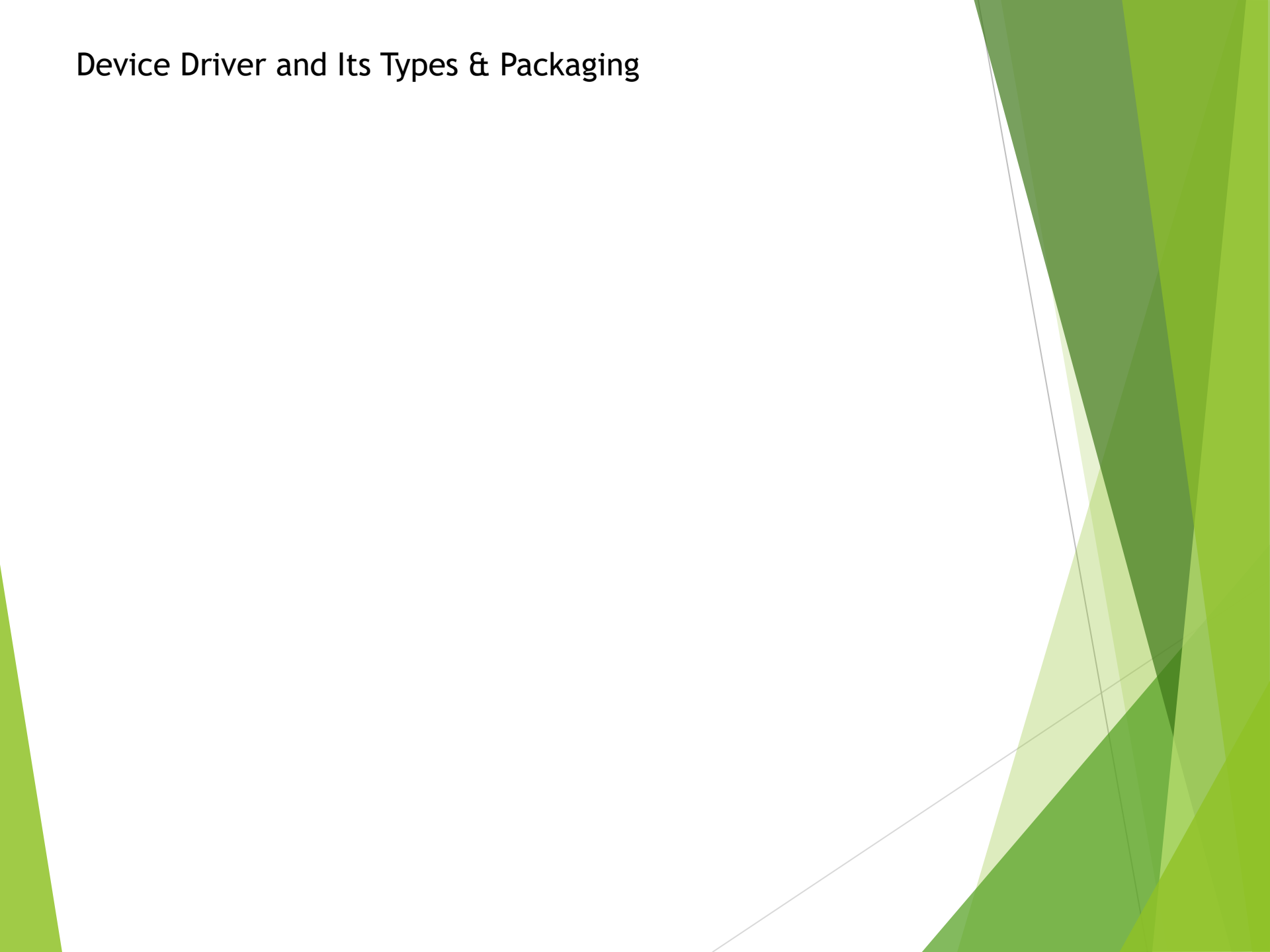
Sometimes uses custom scripts or Active Setup to apply per-user settings.

## Conclusion

- Self-Repair keeps apps intact
- Active Setup ensures user config
- Excel Add-ins enhance app packaging

 Mastering these ensures reliable enterprise deployments

# Device Driver and Its Types & Packaging



# What is a Device Driver?

- A device driver is a program that allows the OS to communicate with hardware.
- Acts as a translator between hardware and software.
- Each hardware component needs its own driver.

# Functions of a Device Driver

- Communicates with hardware devices
- Converts OS instructions into hardware actions
- Handles data transfers
- Ensures hardware compatibility

# Types of Device Drivers

1. Kernel-mode drivers
2. User-mode drivers
3. Virtual device drivers
4. Installable drivers
5. Firmware drivers



## Kernel-Mode Drivers

- Runs in the kernel space of the OS
- High privilege level access
- Examples: disk drivers, display drivers
- Risk: Can crash the system if faulty

## User-Mode Drivers

- Operates in user space with limited access
- Safer - cannot crash the OS
- Examples: USB and printer drivers

# Virtual Device Drivers

- Simulate hardware behavior
- Used in virtual machines
- Example: VirtualBox or VMware drivers

## Installable Drivers

- Loaded dynamically when required
- Examples: Plug-and-Play device drivers (USB flash drives)

# Firmware Drivers

- Embedded in hardware
- Works independently of the OS
- Example: BIOS/UEFI drivers

# Packaging of Device Drivers

Allows drivers to be:

- Installed
- Updated
- Distributed across systems

# Driver Packaging Formats

- INF Files (Windows)
- .SYS Files (Windows)
- .DEB / .RPM (Linux)
- PKG / DMG (macOS)

# Driver Packaging Tools

- Windows: Device Driver Wizard, WDK
- Linux: DKMS, make, insmod
- macOS: Kext Utility



# Driver Installation Process

1. Detect hardware
2. Match with available driver
3. Install driver package
4. Configure settings
5. Load driver into memory

## Use Cases of Packaged Drivers

- Mass deployment
- OEM preloading
- Updates via OS-specific services

## Conclusion

- Drivers are essential for hardware
- Different types for different functions
- Packaging simplifies distribution and updates

# Self-Repair (MSI) - Extra Insights

- ▶ Widely used in enterprises to reduce helpdesk incidents
- ▶ Limitations: Only repairs components defined as KeyPaths
- ▶ Example: If a DLL goes missing, launching Word triggers self-repair automatically
- ▶ Best Practice: Define KeyPaths carefully to avoid unnecessary repairs

# Active Setup - Extra Insights

- ▶ Similar to Group Policy Preferences but lighter
- ▶ Can impact logon performance in VDI if misconfigured
- ▶ Example: Automatically create Outlook profiles for each new user
- ▶ Best Practice: Use only for critical HKCU entries, not heavy scripts

# Excel Add-ins - Extra Insights

- ▶ Use Cases: Finance macros, BI tool connectors, enterprise reporting tools
- ▶ Deployment Challenges: Per-user vs machine-wide installations
- ▶ Best Practice: Digitally sign COM add-ins for security
- ▶ Version Control: Keep add-ins updated across enterprise systems

# Device Drivers - Extra Insights

- ▶ Critical for OS deployment - storage and network drivers must be present
- ▶ Security Concern: Unsigned drivers pose major risks
- ▶ OEM Drivers: Custom features but may lag in updates
- ▶ Universal Drivers: Provided by OS vendor, more generic but stable

# Driver Packaging - Extra Insights

- ▶ Automation Tools: SCCM, Intune, MDT help distribute drivers
- ▶ Windows: INF and CAB files commonly used for deployments
- ▶ Linux: DKMS ensures drivers are recompiled after kernel updates
- ▶ Best Practice: Test drivers in staging before mass deployment