Name:- Raushan Kumar Sharma

User I’d:- 34745

Mail:- [raushankumarsharma01582000@gmail.com](http://raushankumarsharma01582000@gmail.com)

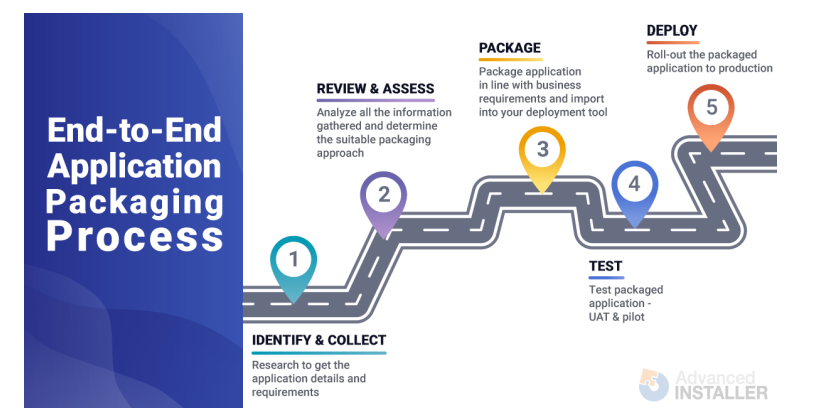
Topic Assignment:-

Application Packaging process:

The process of creating a standardized, deployable software installer to provide an improved end user experience.

End-to-End Application packaging process has five stages:-

* IDENTIFY AND COLLECT
* REVIEW AND ASSESS
* PACKAGE
* TEST
* DEPLOY



* Application Packaging Industry trends – What skills are companies looking for? Nowadays, when a company is looking to hire an Application Packager, they expect for that person alone to take care of the entire Application Packaging process. They have this expectation for both permanent employees and contract-based packagers.

There are two main reasons why companies take this approach:

 to cut down costs,

 to improve app packaging process efficiency by speeding up the process and reducing the time an application takes to be packaged and deployed to production devices – the more people are involved in the process, the slower the process is.

* Downside:-

Difficult to find the candidates with right skills, and who have fully understand the end-to-end packaging process.

* What is Application packaging and why do we need it?

Modern organizations commonly have hundreds to thousands of software products installed on client computers. Each software product comes with its own unique installation and configuration requirements, making the management of the software products a more complex process.

And it will become even more complicated with Windows 10 and Windows as a Service model. I will go into why later in this article.

Application Packaging helps organizations in need of a way to take the burden off their IT support teams while providing an improved end-user experience.

* Benefits:-

 ensures a consistent, stable and reliable standard environment;

 increases the efficiency of software management by streamlining the software deployment along with any customization needed;

 mitigates security issues;

 decreases risks for business disruption;

 helps reduce the on-going administration and support costs.

* Demerits:-

 Time consuming

 Compatibility Issues

 Dependency Problems

 Troubleshooting challenges

 Not suitable for all Apps

Note:- Application Packaging is not a miracle recipe for all problems.

* Does Application packaging solve all your problems?

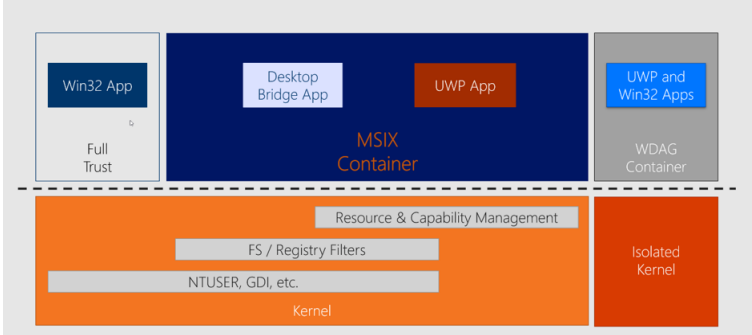
There are many companies that expect to resolve all their application related issues, including compatibility issues, just by packaging each and every one of them.

some compatibility issues that could be sorted by remediating the application and including the fixes within the package, this is not possible for the majority of them.

Solving application compatibility issues in packaging relies on containerized packaging solutions. That are

 App-V, which enables applications to run in their own container on the client computer;

 The new MSIX packaging format, which is based on the same concept of containerization as App-V



* ACT (Application Compatibility Toolkit) :-

Developed by Microsoft in 2010 t to help enterprises fix application compatibility issues that may occur due to the changes between Windows Operating System versions.when enterprises were looking to upgrade from Windows XP or Windows Vista to Windows 7. Since then, once Windows 10 was released, the majority of ACT functionalities were moved to Windows Analytics - which was recently replaced on 31st January 2020 by Desktop Analytics.

* End-to-End packaging process:-

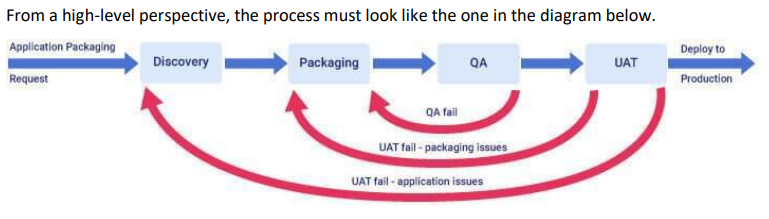
the end-to-end packaging process consists of 3 main steps:

1. Application Discovery

2. Application Packaging

3. UAT (User Acceptance Testing)

Organizations that care about the quality of their packages have also introduced an extra step: the review of the packages created by a senior packager (QA).



Application Discovery

The first step is the Application Discovery, and it consists of the following:

 validating the application source file

 ensuring that the application is fully functional within the organization environment and that it works as expected.

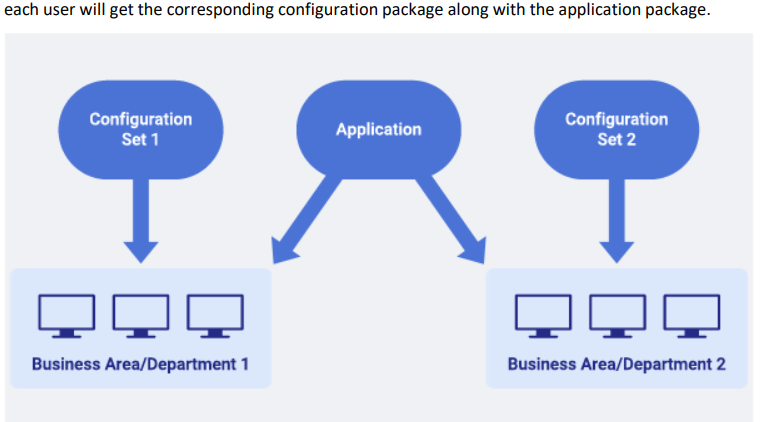
Through this step, all the requirements and details of the application are collected and recorded accordingly. Keep in mind that some applications may require less discovery than others. Also, sometimes vendor support is needed if the application fails to install or work as expected due to any compatibility issue or misconfiguration within the organization environment.

This is not the time to be superficial since the future package is going to be created based on the requirements and the application details recorded during the discovery.

RECOMMENDATION: Plan the application discovery carefully and make sure to get the correct outputs, which will be then used for packaging your application.

During the discovery phase, also gather the following information that could significantly affect the packaging process :

Is the application used differently by users from different business areas? Does each business area require different sorts of configurations?



Along with MSIX, Microsoft also released MSIX Modification Package - the packaging format meant to store the customizations of the application.

Application Packaging:-

Application Packaging covers the actual package creation based on the requirement and details gathered in the Discovery stage. There is not much left to be said here - depending on which packaging format you prefer to use within your organization, there are all sorts of best practices articles online .

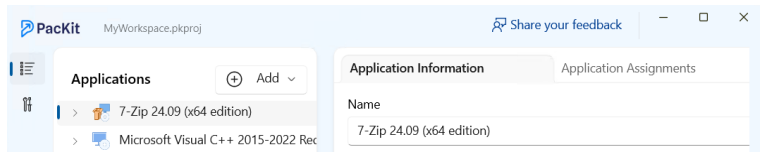
User Acceptance Testing Application:-

UAT is the step of the application packaging process before the packaged application is deployed to production and it consists of validating the package created and making sure that the packaged application is fully functional and works as expected – basically, it must behave the same as the vendor source file tested earlier in the Discovery stage.

Application Deployment:-

Application Deployment is the process of installing a package created using a software management tool such as Configuration Manager (formerly known as SCCM) or Intune.

* **Best Practices for User Acceptance Testing, Deployment and post Deployment**
* Always use VMs for Testing
* Avoid multiple patch Packages- repackage with fixes
* Run periodic housekeeping
* Implement application rationalization
* Maintain clean naming conventions and document everything
* Retire Uninstall Remove applications(in that order)
* Avoid manual installs in growing organizations



* **Tools & Formats**
* MSI /MSIX - Microsoft Windows Installer formats
* App-V - Virtual container for apps
* Configuration Manager (SCCM) - For scalable deployment
* Intune - Cloud-native device and app management
* PacKit - Automation tool for packaging, conversion & deployment
* Desktop Analytics - (Replaces ACT) for compatibility insights
* **Recommended Approaches**
* Prefer Application Model over Package Model in ConfigMgr
* Use Supersedence logic correctly when upgrading/replacing apps
* Keep apps up-to-date with the Windows Servicing Model (WaaS)
* Don't assume packaging solves all compatibility issues-use modern formats (MSIX/App-V) for isolation
* **Conclusion**
* Application packaging is key to streamlined software deployment in modern enterprises.
* Follow structured practices from discovery to deployment to decommissioning.
* Proper packaging reduces errors, cost, and downtime for IT infrastructure.

**MSIX Modification Packages**

* MSIX allows modular customization using Modification Packages.
* Useful when different business areas need different configurations layered over the same base app.

**Windows 10 vs Windows 11 in App Packaging Context**

**Windows 11 Advantages:**

* Modern UI & Snap Layouts.
* Better performance & security (TPM 2.0, Windows Hello).
* Smaller, faster updates.
* AI Assistant-Copilot.
* DirectStorage for gaming and better performance apps.

**Windows 10 Advantages:**

* Familiar interface.
* Wider hardware and software compatibility.
* Stable, cost-effective for legacy systems.  
    
  **Considerations:**
* Check app compatibility.
* Consider hardware specs for Windows 11.
* Older apps may perform better on Win 10.

Windows 11 built upon Windows 10’s foundation which offers more modern and refined user experience with enhanced features and performance. Both OS are compatible with a wide range of apps, including those within an “App Pack”.

**MSI (Window Installer) Contexts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Context** | **Runs As** | **Access Level** | **Use Case** |
| **User** | Current user | Limited to profile | Personal apps, user-specific tasks |
| **System** | SYSTEM account | Full system-wide | Global installs, services, scheduled tasks |
| **Admin** | Elevated privileges/permissions | Full + UAC prompt | Changes to services, drivers, etc. |

**Logon Scripts + Active Setup**

**1. Use of Active Setup**

* **Purpose**: Executes user-specific tasks at **user logon**
* **Action**: Copy files, update registry, run scripts within MSI.
* **Example**: Copy config files to AppData from machine-level location.

**2. Creating & Assigning Logon Scripts**

* **Script types**: .bat, .ps1 (PowerShell), .vbs – choose based on need.
* **Function**: Copy user files/settings from shared drive to user profile folder.
* **Assignment**: Use **Group Policy** to assign to:
  1. Individual users
  2. User groups
  3. Organizational Units (OUs)

**3. Deployment Strategies**

* **Group Policy**:
  1. Assign logon scripts by user/group/OU.
  2. Distribute MSI that uses logon scripts/Active Setup.
* **Software Distribution**:
  1. Tools like SCCM or Intune can also deploy MSIs with scripts.
* **Language Choice**:
  1. Use batch for basic logic.
  2. Use PowerShell for complex actions (preferred for modern systems).

**4. Example Scenario – Settings Deployment**

* **Need**: App needs to load user-specific settings at login.
* **Steps**:
  1. Add **Active Setup** in MSI to trigger script at login.
  2. Script copies settings from \\server\netlogon\MyApplication to %AppData%\MyApplication.
  3. Deploy via Group Policy or Software Distribution tool.

**5. Best Practices**

* **Error Handling**: Handle failures like network issues gracefully.
* **Security**: Protect sensitive file paths & credentials.
* **Testing**: Always test script in real user environments.
* **Documentation**: Record script logic, deployment method, and config.