

Process of Building up an Application Packaging

Trupti M. Hake*
M.E. (Computer Engg.) pursuing,
Sinhgad College
Pune (M.S.) India
*truptihake@yahoo.com

Pravin S. Metkewar
Assoc Professor,
Symbiosis International University,
Pune – (M.S.) India

Abstract: Application packaging bundles applications and compatibility patch for operating systems into a single file called a distribution unit (.msi), which makes it easier to deploy and install them on user's computers. Packaging reduces the total cost of ownership (TCO) for the customers by enabling them to efficiently install and configure the applications. This results in an application package, which provides the product with newer capabilities like advertising features without installing them, installing products on demand, add user customizations etc. Now a days there is new development in this technology i.e., Application virtualization which has been introduced. In this paper we have presented architecture and process of application packaging, its advantages and how it also reduces the overall cost.

Keywords: .msi, Package Request Form (PRF), Application Compatibility Report (ACR)

I. INTRODUCTION

Application packaging has many benefits for application developers and users. Traditional or legacy software installations in the past were developed in a variety of ways. Setup developers designed their install programs to concentrate on their own products which often impacted on programs that were already installed on the PC. Application Packaging is a process which customizes software as per user's or client's requirement. The word customize includes adding a new feature to it or deleting any unwanted feature from it. In this wrapping, the software installation, re-installation and removal are customized according to user's needs. The process of packaging is more like a client oriented technology.

Application Packaging is the process of creating distributable bundles of application files, registry keys, INI files and all other entities that are needed for an application to function correctly in a Windows operating system environment. These bundles are called application packages. Package is a file that contains an application files and

metadata required to create and manage instances of the application.

Microsoft Application Virtualization, known as App-V, is the new name for the technology formerly known as SoftGrid from Softricity. App-V is a set of products that allow enterprises to perform virtualization at an application level rather than at server level. App-V solves application conflict issues with standard, "off-the-shelf", software packages and enables centralized management of virtual application packages.

It will install and run the applications without breaking other applications. Execute multiple versions of the same application on the same machine. Make applications that are not multi-user or multi-tenant aware instead work the way you want them to work.

The goal of packaging an application is to reduce to zero technical knowledge required to install and use the application (no database setup, no web server setup, no application configuration, etc.) The actual package will install through a single click from the user.

A. Microsoft Software Installer (MSI) and Installation files

An .msi is an installation package containing all the information that Windows® Installer needs to install or uninstall an application. The information usually stored in an .msi file would be the applications files, shortcuts, registry keys, security enhancements etc.

Properly setting up all necessary installation files reduces software errors. Some examples of necessary installation extension files are Windows Installer database (.msi), executable file (.exe) and Windows Installer patch (.msp). All basic but necessary installation files should be created to install, run, update and uninstall the desired application properly. For Windows operating systems, Windows Installer provides reliable management of application files that can distribute and run managed files on

any Windows computer. Closely work with all engineers to be sure that all necessary files were included.

B. Why packaging:

With legacy installations in the past, there might be two applications using the same file, but one might be designed to run on an older version. This led to “.DLL hell” where the installation of one application would cause other applications to stop working. Inevitably the process of ‘rolling back’ or uninstalling the program resulted in system down time and - more severe cases - having to rebuild the PC completely and reinstall all your software.

Several years ago Microsoft set out to address these and other issues. One of the emerging solutions was the development of Windows® Installer technologies. The Windows® Installer allows the efficient installation and configuration of applications. The installer can also provide your installations with new capabilities not previously available with legacy installations. Some of the added capabilities are:-

- Advertise applications without installing them until required
- Self-repair an application if corruption occurs
- Upgrade applications quicker and more efficiently
- Better management of shared resources and
- Customize your software installations to your organizations requirements.

One of the most talked about benefits of MSI is Self Repair. Also referred to as Self-Healing, applications packaged within an MSI have the ability to restore deleted files and other elements of their setup. By utilizing these added capabilities and efficiencies your organization can expect to achieve a lower Total Cost of Ownership.

II. LITERATURE SURVEY

In a modern organisation, it is quite common to have hundreds of applications installed on client computers – each with their own unique installation and configuration requirements. Whilst some of these requirements are simple to accommodate, others can be complex and this often leads to difficulties for the organisation to deploy and support. An effective way to resolve this problem is through automating the application installation and configuration process, often referred to as **Application Packaging**. This process results in a consistent and reliable application deployment experience. **Application Packaging** helps organisations manage software for desktop and server systems more

efficiently by streamlining the configuration and **software deployment** processes and helps to reduce on-going management costs [6].

Application packaging is viewed as a critical component for efficiently managing software on servers, desktops and notebook systems. By streamlining software install, uninstall, patching, upgrading and repairing, application packaging can help reduce costs associated with each phase of the application lifecycle. In particular, application packaging is designed to reduce costs and improve efficiency during the deployment and post-deployment phases. Such benefits typically depend on having a stable environment in which packages are distributed automatically by enabling fast, standardized software installations; it is designed to minimize desk-side visits by support staff. Business disruptions caused by software failure are avoided; there by helping to reduce costs for IT support and business operations, respectively. When implemented as an IT best practice, application packaging can help create a cost-effective software repository that is in line with overall business priorities. With application packaging strategy, organizations can help reduce administrative costs while providing business benefits. This approach enables administrators to set and enforce corporate software configuration standards [1].

Application packaging is one of the most important software management tasks. IT departments oversee in order to maintain a stable and productive end user environment. Desktop applications must work as expected, be configured and updated to the required level and be available from different platforms in a consistent, dependable manner. They have art of application packaging, allowing for significant cost savings and time reduction for both IT and business operations [4].

In any organization the system administrator knows that the variability in end users' desktop results in increased maintenance cost. Packaging/Repackaging applications help eliminate issues through customizing an installation so that it behaves in a consistent manner adhering to the company's standards for mass deployment. Braves Technologies solutions for application packaging help enterprises manage the ever growing volumes of applications efficiently and reduce the cost associated with application management [3].

In turn this minimizes the frequency of administrative errors during installation. By specifically using MSI, the Windows Installer service offers features that help application deployment. These include powerful self-repair and rollback capabilities that are designed to

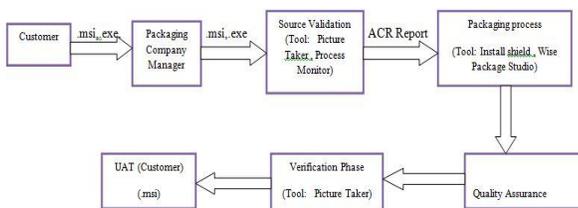
dramatically reduce the occurrence of deployment-related desktop software problems.

Application compatibility when moving to a newer operating environment has always been a challenge for all organizations [2] Most of the application incompatibilities arise due to:

- Change in the security model imposed by the new Operating System
- Adoption of newer Industry standards which invalidates older mechanisms
- More restrictions brought forward in the kernel to prevent unlawful attacks
- Restrictions in inter-process communication

III. ARCHITECTURE OF APPLICATION PACKAGING

The process of application packaging goes through different phases and different tools are used.



When customer sends the application to packaging factory, it may be present in different forms like it can be .msi, .exe, loose files, .exe, .msi extracting .msi etc. so to package this different type of source need to follow below steps.

- Source validation and ACR Report Formation
- Packaging Process
 - Capture
 - Editing
 - Testing
- Quality Assurance
- Verification Phase
- User Acceptance Test (UAT)

IV. SOURCE VALIDATION AND ACR REPORT FORMATION

To move further basically we have to test the application compatibility for different windows versions like

Windows XP 32bit, Windows XP 64bit, Windows 7 32bit, Windows 7 64bit etc. So in ACR report formation we have to follow:

- Application Installation Result.
- Application Functional Result.

A. Application Installation Result:

Need to test what exactly application installs like services, drivers, kernel mode drivers, registries etc. Whether this drivers works properly for the windows version which is required for customer. What is the installation directory of the application whether it gets installed in C:\Program files\ or somewhere else i.e., which is installdir of application in technical terms. If it is .msi then we have to check the event viewer message for it, if the installation is successful then the given message is “Application installed successfully”.

If Application shows any Program Compatibility Assistant (PCA), User Access Control (UAC) message during installation or launching the shortcut then this needs to be document as well since it may require remediation. And the process also involves with providing the solution by which it will not appear at user end that is called Remediation. This is can be followed by using “Microsoft Application Compatibility Toolkit”.

B. Application Functional Result

Functionality testing intends to check whether the application requires any backend database connectivity or any network connection. And also to check the application functionality as per the requirement mentioned in the PRF or the base OS behavior. Base OS means for which application has been manufactured.

The entire test results have to be documented in the ACR report. Depending upon the above test result ACR report status can be concluded, which shows application compatibility for target OS.

Following are the different status of the ACR report

- Green – Compatible (32-BIT)
- Green* - Compatible (32-BIT) (limited functionality testing)
- Amber - Application requires a fix to work on Windows 7
- Amber*- Application requires fix (limited functionality testing)
- Red - Not Compatible
- Grey – Unable to Install the application

Tools used for ACR Report Formation:

- Picture Taker
- Process Monitor
- Microsoft Application Compatibility Toolkit

V. PACKAGING PROCESS

As per the ACR report status of the application package moves to the packager. If the given source is provided in loose files or in .exe format then it requires to be captured and make it in the .msi format. As .msi have many advantages over .exe.

Following tools are available to package the application.

- Install Shield
- Wise Package Studio

These are the basic tools to complete the packaging process. These tools have many sub tools for different purposes like capturing, validation, comparing etc.

In packaging basic standards are applied. Like junk files, folders, registry keys are removed from the package. If any required remediation is mentioned then that is applied. If there are any post and pre installation configurations mentioned in the PRF, any permission settings needs to be included in the package then those are also implemented.

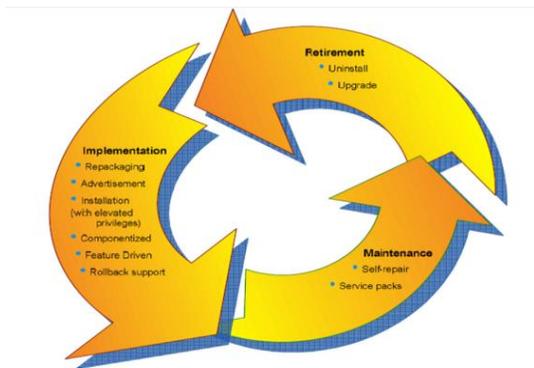


Figure 1: Packaging - Process life cycle

VI. QUALITY ASSURANCE(QA)

After packaging the application, maintaining the quality of application in terms of customer's need is very necessary. So quality analysis is one of the important phases of packaging. This step basically does a rigorous check for all the general standards and any client specific requirements had should have been implemented. All the documentation

like "Readme", PRF are filled properly and kept at exact location.

VII. VERIFICATION PHASE

After the QA phase is completed successfully, the stage comes where it's verified that the application gets installed successfully and also there is clean uninstallation of the application. All the services, driver gets installed and functions properly as with source. After creating .msi, basic stage is to check whether the application behavior meets client requirements.

VIII. USER ACCEPTANCE TEST (UAT)

This is the final stage from client side, as per the requirement whether the application is running properly or not. If not then packaging team have to follow the same process again.

IX. COST AND CLIENTELE

Minimizing Cost

Eliminating certain software errors in application packaging helps reduce long-term costs. The initial application production process costs less before the application is released to the company, clients and customers. Take the time to create an effective strategy that covers all possible problem areas where costs are predicted to rise. For example, when application users run into errors or even simply have questions about the application, it will cost to hire a customer service support team.

Maximizing Clientele

Save enough money to make necessary repairs after the application has been used by customers. After an application is released, one marketing tactic for gaining additional clientele is to continue support for the application. For example, if an application is for a banking software client and if it has failed arbitrarily, the employee might instinctively contact. By providing 24 by 7 supports after the application releases, any customer can directly contact the application distributor regarding any problem at any time.

X. ADVANTAGEOUS SCENARIOS

- Customize Applications to suit the user needs.
- Simplify the Installation and Un-installation Procedures.
- Saves Time in both Installation and Un-installation.
- Once packaged, applications can be quickly installed on a range of desktops in multiple

locations, saving administrative costs, minimizing support and repair expenditures.

- Have a great flexibility of obtaining the lost files through a phenomenon called Self-Heal, this reduces the down time of application. If a critical file (e.g., DLL or .EXE file) that is part of the distribution may be corrupted or is deleted, the user can be prompted to repair the installation by presenting the original .MSI distribution. Additionally, if the installation media is available (e.g., on a network share), the repair simply happens automatically.
- Can be advertised. So that on demand installation could take place.
- Upgrading of the application can be done with ease.
- Clean installation and Un-Installation is achieved by a process called Roll-Back.
- Simplifies management of new user set-up along with the revision and distribution of software repairs and new applications to existing users. Application recovery can also be improved.
- Helps eliminate uncontrolled software downloads and installation, enables applications to be safely removed and reduces non-business traffic on a corporate network.
- Using .MSI format can automate software distribution process and ensure that the installation doesn't break other applications that have already been installed.
- Application is installed via an OS service.
- Windows Installer provides an application programming interface (API) that lets programmers and administrators see whether a specific application is installed on a machine.
- Scriptable API. This whips together a VBScript to help us with the MSI file manipulations. The API to manipulate MSI files is so powerful that it can create, validate and update packages, trigger installs and uninstalls, examine the MSI repository data on computers, and perform some custom actions.
- Served installs. Because MSI files can be housed in a share point and delivered via a server, we can keep our installation files all in one place or move them around -- closer to the users if necessary.

CONCLUSION

In earlier days, Application Packaging was a part of software development process. Now a days it has separate unit like software development, testing, maintenance etc. as different operating systems evolved and available in the market. Due to emerging trends and technologies, this is the

need of the day to package the application and deliver it to the client at door step. This paper accomplishes to understand what are the stages involved in packaging the application and also to emphasize on tools to be used as per the client requirement and technology support. Advantageous scenarios have been remarked in section (X) in order to minimize cost and time respectively. Finally Industry can maximize clientele in order to generate more revenue with best standards suitable in the market.

REFERENCES

- [1]. SaiKuricheti, AparnaVolety and SandeepNayak (Sept 2009), Best practices in Application Compatibility, published by Infosys company.
- [2]. J Ukkakouletsis (Nov 2005), the Basics of Application Packaging: Best Practices for Enabling Reduced Software Management Costs.
- [3]. http://www.bravestechologies.com/braves_technologies_datafiles/app_pack_benefits.htm.
- [4]. http://services.atea.com/services_uk/services/application-packaging.aspx
- [5]. <http://www.symantec.com/connect/articles/application-packaging-basics-and-benefits>
- [6]. <http://www.2way.net.au/services.html>
- [7]. Application Packaging Developer's Guide Sun Microsystems, Inc. 4150 Network Circle Santa Clara, CA 95054 U.S.A. PartNo: 817-0406-13 April 2009