FP-International Journal of Computer Science Research (IJCSR)
Volume 2, Issue 2, Pages 60-63, June 2015

OP: Optimization of Application Packaging Design

Nataraj.J
M.Tech, SITE School
VIT University Vellore- 632014
natarajlogin@gmail.com

Prof. KamalaKannan .J
SITE School, VIT University
Vellore-632014
jkamalakannan@vit.ac.in

ABSTRACT
Software Installation process is the vital part of Software product delivery in the IT industry. Application packaging is vague to implement in every release of software product. In prior work, people readily worked and remade the software package and its relevant issues to eliminate an installation issue. Even though that has been taken more time consumption, effort handling, technician required and loss of cost resource from the product. The proposed one will eliminate the repackaging process using Package Transformation technique to resolve the issue while install and release the package. Then the package deployed in an install shield to get NMS application.

Keywords
MSI Transformation, Windows Installer, Installshield, NMS application, Supporting Files

I. Introduction

The Network Management System is allowing the hardware/software developer to manage an entity of Software modules within the large networking system. The NMS application likes OpenNMS, NMIS, iPECS, and 8770 and so on. Each application has delivered using application packaging process. Generally the software release is the way to get an application in varies modified version. Each version of the software can build using application packages. Installation developer can collect an updated package and perform the build process. Generally the software application has two kinds of files such as Setup files and Supporting files. The Setup files are the functional part of an application. It performs core activities of software application. The software modification is updated and released as further version whenever can build the application package. Once if found any issue in an application, have to hot fixes of those issues and released as minor or major upgrades based on the product code and version. The functional part of setup files modification has done using patch production. After the collection of patch, we can build the application packaging and release the minor or major version of the NMS application.

Another kind of files like supporting files can handle the setup files activities throughout the installation process. If an installation process is interrupted due to the supporting files, can't fix the issues to continue an installation process.

In existing work, can remade the application package once again and fix the solution to continue an installation process. Its take more time consumption, technician effort and wastage of money and other resources. An application packaging issues can be solved by using Package Transformation techniques to solve the issues which are coming from the supporting files. Then resume the build process and release appropriate NMS product version.

This paper follows: Section 2 describes the related works, Section 3 explained the proposed system, and Section 4 shows the Design and Implementation and Section 5 conclude the result. Section 6 has references to improve the enhanced features and analyze the existing data.

II. Related Works

The development of NMS application has varies phase to deploy the software. It's follow an agile software development process. In prior work have done as follows

Application packaging could build up the product and release appropriate version using Application Re-packing techniques (1). The application packaging can remade the entire software package and again start to build the process. The package contains the collection of application files that can be used to build and released as NMS application version. This may be minor or major upgrade of product version based on the modified binaries or added features.

The software deployment method (2) based on the Installation package can perform the production process of large clustered application. There are three methods to deploy package installation such as Disk-based deployment, behavior based deployment and package based deployment. It consumed less time and effort to complete the build process when compare to prior work of packaging process.

The deployment of application package (3) in NMS application release is most challenging task. The network management system provides the infrastructure for building enterprise-level server-side distributed java

components. They implemented web-based java application using maven tool. It can drag the package build and release the product.

a) Application Packaging mechanism

Desktop application management techniques are being developed to help enterprises administer their existing PC investments more efficiently, reduce end-user support costs, and minimize end-user business disruptions. One such development is application packaging, which is emerging as a critical component of an overall software configuration management strategy.

Application packaging involves the preparation of standard, structured software installations targeted for automated deployment. Automated installations, or packages, must meet the installation requirements for a specific environment: corporate standards for software usage and desktop design, multiple languages, regional issues, and software related support issues. In addition, pack-ages must prepare for both commercial software and applications developed in-house.

To enable this level of application management, Microsoft now provides the Microsoft Windows Installer (MSI) services a part of its desktop operating systems. Starting with Microsoft Windows 2000, the windows Installer has been included in the ease desktop OS.

For earlier OS versions, windows Installer can be downloaded from Microsoft's Website. This database-driven service resides on workstations and controls he installing, uninstalling, patching, and repairing of software. Input into the Windows Installer is an .msi formatted file, which is further explained in the "Understanding Microsoft Windows Installer".

b) Release state

When deploying new applications (4), many administrators consider an automated approach to software installation as an attractive option to help save time and help reduce compatibility issues. However, the testing needed to verify that a new application will not cause an old application to fail can be difficult, if not impossible, in an uncontrolled environment. Although business disruptions caused by a new application deployment often cannot be measured directly, impaired end-user productivity can be costly. So window installer can help to address the configuration management.

III. Proposed System

The Software package is the collection of binaries to build and release the version of the software application. Binaries files like .dll, .cpp, .jar, .war, .sql, .php, .asm. These binaries are maintaining in the repository.

- Repository - Source Cod Management
- Build the SCM binaries → package
- Patch production: use fix to an issue and check-in to the repository and compile it.

During the package production, there are two types of file has setup files and supporting files. The setup files have main functional module of NMS application. To track the installation, the supporting files are used. The Supporting files have the list of log files which include the details of setup files execution.

a) Source Code Management

Usually many of the developers in an organization can develop their source code and reported in repository tool like bitbucket, projectlocker, cloudForge, GitHub and so on. It can recover from roduct failure. It have source code repository backup.

Then we can build the package using maven tool, Installshield and so on. Then can get the source code from the repository tool and perform the packaging using package transformation techniques.

It gives idea can help to give elucidation of an installation and release interruption. We can give patch to the supporting files and resume the installation trigger.

b) System Architecture

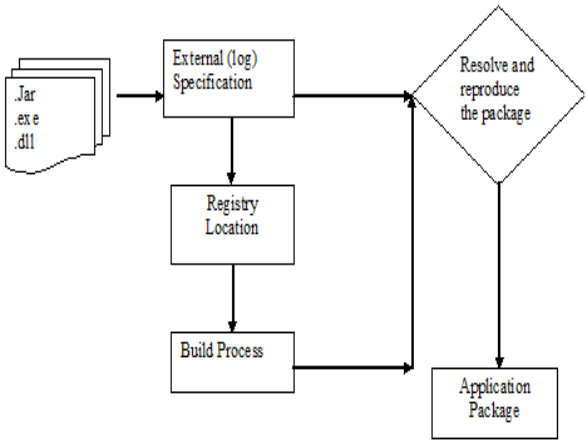


Fig1. System architecture of NMS Application package

Once the binaries are triggered by maven tool, the package will start the build process. It may reduce working effort and cost. The system architecture explains as following section

In the existing system, the package production has interrupted in the setup files. If the supporting files of

log are failed, we can solve issues and again start the package build from an initial state. It can avoid the failure of installation and time consumption.

IV. Implementation

c) Package Transformation

A building software package is an essential part of product delivery to the customer. Software product principle - Continuous Integration process The Source Code management have maintain application development code in mercurial branch repository. The binaries are built by maven tool.

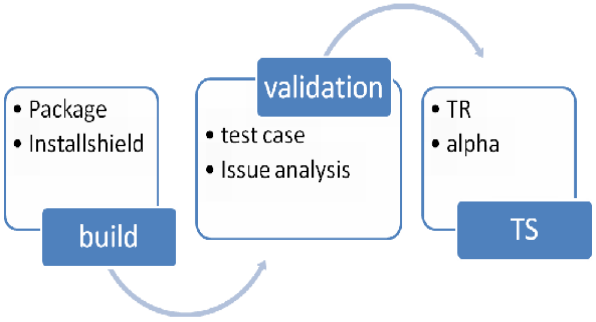
- Package – release.zip, propertise.zip, log.zip
- Package can be tracked by log files(.ini, .bat, .vbs)

Installshield is used to rapidly build, test, and deploy installations that target Windows-based systems. An Existing system has deployed the application package using AI installation tool. This tool has some of the drawback as

i) Manual testing more cost, maintenance effort during the package builds.
ii) It can't perform enough operation in the Virtual machine

So we can analyze the problematic moment and implement the software package in the Installshield 2014 version to enhance the product features. The process of creating Install script has basic MSI files and made it as MST installation projects, and creating global installations of the windows application.

If the failure occurred in validation state or technical support, again the package can build and test from the developer to technical support. This is too long process and implementation is difficult. This can be avoided using package transformation techniques.



Finally package transformation build completed and goes to installation design.

Using .MSI format can automate software distribution process and ensure that the installation doesn't break other applications. 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.

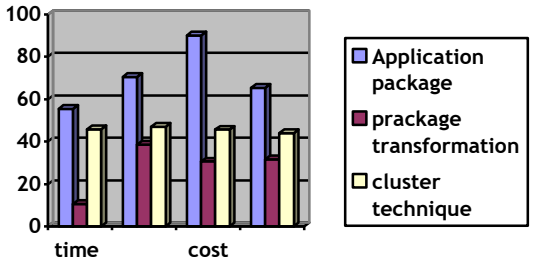


Fig2. Graphical output of packing system

Application Packaging was a part of software development process. 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. The client requirement and technology support.

V. Conclusion

The Software package has released many products with different version in the modern life. It's more helpful to deliver the target application to the customer. The package transformation technique can eliminate the customer-side problem. It can reduce the time consumption, effort, cost and other product related issues. It can automatically test and can release the target version. In the future, can enhance the package release technique to implement dynamically resolving an installation package and delivered it.

Reference

[1] Alan Dearle. Software Deployment, Past, Present and Future. Future of Software Engineering (FOSE'07).

[2] Peter H. Hughes, Jakob Sverre Løvstad. A generic model for quantifiable software deployment. International Conference on Software Engineering Advances (ICSEA 2007).

[3] Meriem Belguidoum, Fabien Dagnat. Analysis of deployment dependencies in software Components. SAC'06, April, 23-27, 2006, Dijon, France.

[4] Meriem Belguidoum, Fabien Dagnat. Dependability in Software Component Deployment. 2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX'07).

[5] Bowen Alpern, Joshua Auerbach, Vasanth Bala, Thomas Frauenhofer, Todd Mummert, Michael Pigott. PDS: A Virtual Execution Environment for Software Deployment. VEE'05, June 11–12, 2005, Chicago, Illinois,USA.

[6] Robert Dickau, MSI Installation Design Issues and Best Practices, IA_WP MSI Design, Oct11.

[7] Xu Zhang.Research and Application of Automated Software Delivery System [D] Zhejiang: Zhejiang University.

[8] Brian Elliott Finley. VA SystemImager. Proceedings of the4th Annual Linux Showcase & Conference, Atlanta Atlanta, Georgia, USA October 10 –14, 2000.

[9] N. Panlilio-Yap, D. Ho, Deploying software estimation technology and tools: the IBM SWS Toronto Lab experience, in: Proceedings of the 9th International Forum on COCOMO and Software Cost Modeling, University of Southern California, Los Angeles, October, 1994.