International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 3, March - 2014

Automated Installer (Setup Creator)

Lokesh Mandesa[1], Ruchi Birla[2], Mangesh Bhapkar[3], Anuja Asalkar[4]
[1,2,3,4] - Students of B.E. Computer Engineering
PVG's College Of Engineering And Technology
Pune, India.

***Abstract*—An installer is handy with every software package that you download. It copies or/and updates files, writes registry keys, modifies the configurations, creates shortcuts etc. All this is done directly for the user with just little intervention from the user end. But at the developer's end, the software developer needs to provide his product with this installer for which several line coding is required. The automated installer replaces this coding process with a user interface that automatically generates the setup file with its installer (installation wizard) for the software developer.**

Keywords—Automation, Installer, Nsis, WiX.

I. INTRODUCTION

Many softwares are developed for various purposes. Every software requires installation to be done so as to function properly on end machines. Installation process is performed by means of a setup file which needs to be provided by the software developer. Creation of installer is a tedious process which includes lots of coding, thus at the cost of many resources (time, efforts).

Several line coding is required to create this installation using various technologies (language platforms) which is not an easy process as even for a software developer. Thus a software developer who is a beginner to create applications either requires help of another software developer who is PROFESSIONAL in creating the required installer or having an automated help.

The solution to this is the creation of an automated tool which takes input as files, banners, images, license agreements, etc. from the user and generates the installer thus easing this process. The automated setup creator system creates an installer for the end user who develops any particular application.

The paper is organized as follows. In section II introduces what is an Installer and the tasks it performs, in section III and IV, introduction to the different types of installation/executable files and the WiX and NSIS platforms to develop these installers is given. In section V we define the process of automation that we use to create the installers and thus conclude the paper in section VI following with future scope and then we discuss some of the advantages, shortcomings and eventually end up with references in the next sections.

II. INSTALLER

What is an installer

Installation of a computer program (software/device drivers/plugins) is the act of making the program ready for execution. An installer is a specialized computer program responsible for doing whatever is needed for executing the program on end user's machine because the process varies for each program and environment provided by each computer.

Functions of an installer

An installation program installs the files such as applications, drivers or other software on the particular machine. Once installed, the program can be executed again and again, without the need of it being re-installed every time. Some of the common operations performed by installer are:

- Making sure that necessary system requirements are met
- Checking for existing versions of the software
- Creating/updating program files and folders adding configuration data such as configuration files, Windows registry entries or environment variables
- Making the software accessible to the user, for instance by creating links, shortcuts etc
- Configuring components that run automatically, such as daemons or Windows services

Executable File Types

An executable file is the generated file for installation purpose of any software or an application. It comprises of all the necessary files and folders, registries, dll's, etc. required for proper functioning of the software/application.

Basically, there are two major types of executable files:

1) .msi – installer package file format used by windows.
2) .exe – executable file for various operating systems.
For eg. DOS, OpenVMS, Symbian, Windows etc.

These .msi or .exe files are generated using developing platforms like WiX (Windows Installer XML), NSIS (NullSoft Scriptable Installation System) respectively which are discussed in the next section.

Development Platforms

To develop installers there are some development platforms out of which there are two major platforms used:
WiX (Windows Installer XML) – WiX is a XML markup that is used for creation of installation packages for Windows-based software. The underlying technology is Windows Installer, which is the established standard for installing to any Windows operating system. WiX uses its defined XML set of tags having multiple attributes that are used to create a final setup (.msi) file. Tags for all the functionalities of an installer like changing registries, creating shortcuts, checking versions,

IJERT

performing repair of corrupted file, etc. is done by using various pre-defined tags. It also uses algorithms for the compression so as to create a setup file of a minimal size[5]. But though it gives all the functionalities a large amount of code is written for the creation and the developer must also be completely versed with the WiX script. And thus a lot of resources (time, efforts) can be saved if this process can be automated.

NSIS (NullSoft Scriptable Install System) - is used for creating installers and is open source. A special script i.e. NSIS script is used for this purpose. It also consists of various compression techniques which are used for creating the installer. It is also compatible with most of the Windows versions and adds a small overhead to the installer data. The installers generated through NSIS give the setup in the form of .exe file which the user can run for installing the application. It provides features for registering the keys, creating shortcuts, etc [6]. But with this one needs to have a complete knowledge of the NSIS script for creation of installers. This is time consuming so there is a need for automation. With automation installers can be built even without knowing the script.

Process Of Automation

As we have seen earlier writing of code for installers is a tedious job. In this section we present a means of automating it. Here we would be using a series of user friendly GUI screens for selection of the options for the creation of the installer. The options include all minor details which a software developer takes care while putting forth his design of the installer for eg. making changes to the registry keys, creation of shortcuts, selection of files needed by the application, the default destination for the extracted files on end users machine, etc. Also the developer can specify whether he needs a .msi or .exe file as setup file. After selection of the entire details, just on a click of mouse the user can get a setup file ready for installation on the end users machine in a matter of few minutes.

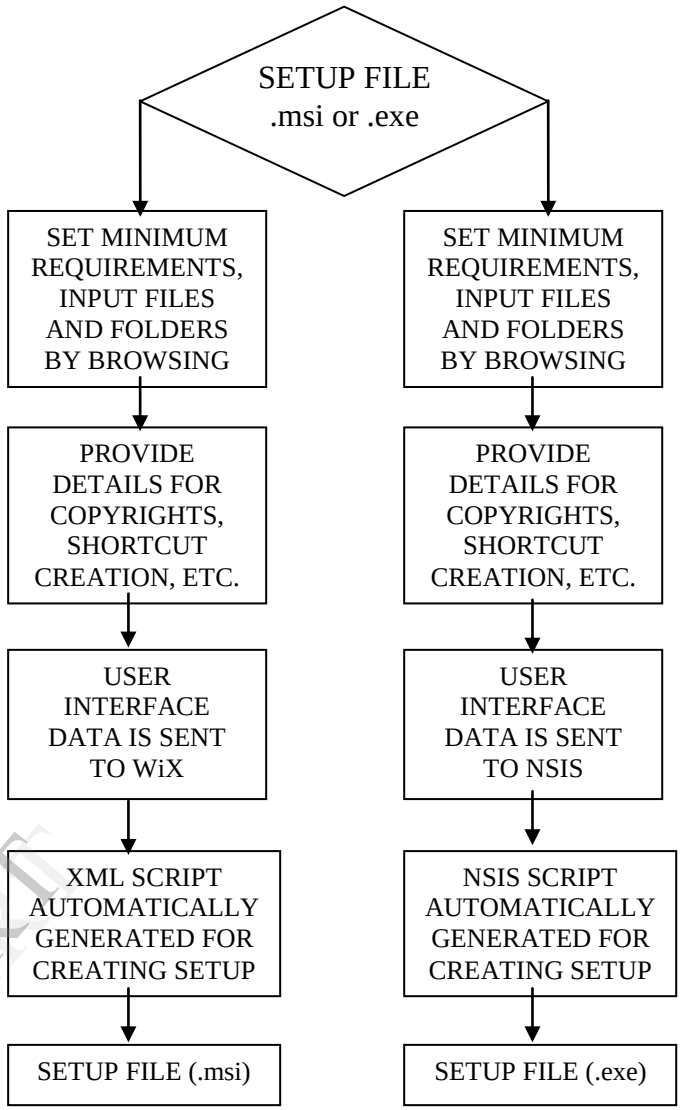

Fig no. 1

The visual Basic framework would be used to create the user friendly interface. Depending upon what options the application developer selects for the installer of his application a runtime script will be generated and then on the click of finish that script will be compiled and executed in order to create the final product i.e. setup file. It will also report errors during the creation in case of corrupted or missing some necessary files. Thus, the created installer would have a standard User interface as well and would contain all the features, files required for the proper execution of the setup as we see in fig. no 1.

The creation of uninstaller would also be automated i.e. as script would be generated at runtime for installer consequently an uninstaller would be created at the same time. Thus the need of extra coding for un-installer is not required.

IJERT

CONCLUSION

In this project we develop an installer that would provide an automated interface for the creation of installation wizard to the software developer. He will not be required to code for the installation process (mapping all files, registries, keys, configuration, shortcuts, etc.) but just has to select files that he wants to be there in the installation process. It would make the installation process simple for the user as well as for the software developer and by providing GUI it makes a user friendly interface.

It would provide an interface for the software developer that would help him to create the setup file(.exe) and to copy and/or update files, write registry keys, write configuration, create shortcuts, etc. that are necessary for installation. All of this would not require any manual intervention in terms of coding.

Future Scope For Implementation:

- Web based installer.
- Installer for other operating systems like MAC, LINUX.
- Non Repetitive installations – using database.

ADVANTAGES AND SHORTCOMINGS

ADVANTAGES

i) As the Software would be open source it would be available to use for all without the requirement of a license copy.

ii) Easy to use interface makes it easier on the developers end to create the setup file of his application.

iii) Lots of coding for development of the setup would be replaced with an automated procedure of setup file creation.

SHORTCOMINGS

i) Embedding one msi to other is not possible in WiX but possible through NSIS, though it would create only .exe formats.

ii) As the software is open source, modifications in the software procedure can harm other applications currently active in the system, this disadvantage can be overcome if the coded installer is encrypted and locked.

REFERENCES

[1] Darwin Sanoy And Jeremy Moskowitz, "The definition Guide To Windows Installer Technology For System Administrators"-realtimepublishers.com

[2] Robert Dickau, "Best Practices for Building Multi-Platform Installers" (White paper).

[3] Nick Ramirez, "WiX 3.6: A Developer's Guide to Windows Installer XML"

[4] MSIcode Scripting Technology for Windows Installer An InstallAware Whitepaper August 2007.

[5] [Online].Available: http://www.nsis.sourceforge.net

[6] [Online].Available:http://technet.microsoft.com/enus/library/cc978328.aspx

IJERT