Batch2Exe: Convert Batch File Into Windows Executable

BSCS 2C - Group 3

Leader:
Baldivicio, Jonathan Eldy I.

Members:
Dela Pena, Jefford
Evaristo, Argie
Pago, Jhondel
Reyes, Damiel O.

Abstract

Batch2Exe is a program for converting batch files to executable files on Windows. It has a user-friendly interface for inputting information and uses advanced compression and obfuscation techniques for security. This paper describes its technical aspects and features, making it a valuable tool for system administrators and developers.

Introduction

Batch files are a type of scripting language used primarily in Windows operating systems to automate repetitive tasks or perform complex operations. However, distributing batch files to other users can be problematic, as they require specific dependencies and may be vulnerable to unauthorized modifications.

To address these issues, this paper will discuss the process of how this program works in order to convert batch files into executable files, which can be distributed as standalone programs without the need for dependencies. This allows users to easily run the script on their system, while also offering a layer of protection against unauthorized modifications which can be a serious security risk, especially if the script is performing sensitive operations or accessing confidential data. This paper will also explore the benefits and limitations of this approach, as well as real-world use cases for batch file to executable file conversion.

Overall, batch file to executable file conversion offers a simple and effective way to distribute and protect batch scripts. Whether for a developer looking to distribute script or a snippet to users or an IT professional seeking a streamlined way to automate tasks, this paper will provide valuable insights into this powerful tool.

Statement of the Problem

Batch files are a popular tool for automating tasks and performing complex operations on Windows systems. However, when it comes to distributing batch files to other users, there are a few challenges.

- Batch files require specific dependencies to be installed on the system in order to run. If the user doesn't have these dependencies installed, the script may fail to run or produce unexpected results. This can create frustration for users and make it difficult to distribute batch files to a wider audience.
- Batch files are often stored in plain text format, which makes them vulnerable to unauthorized modifications. This can be a serious security risk, especially if the script is performing sensitive operations or accessing confidential data.
- Converting a batch file into an executable file may limit its flexibility, making it more difficult to modify or adapt in the future.

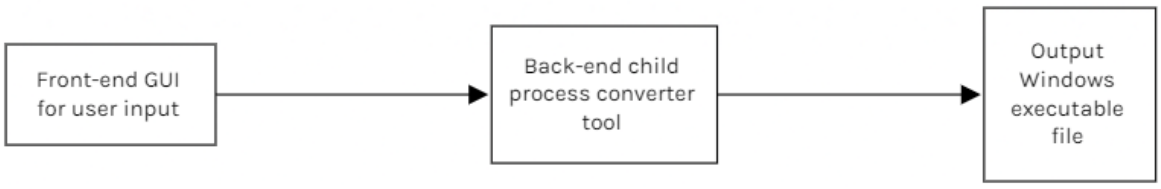
- There may be compatibility issues when distributing executable files to different versions of Windows or other operating systems. This can create additional challenges when trying to distribute your script to a wider audience.

To solve the issues surrounding the distribution of batch files, this system will be divided into two (2) category to convert batch files into executable files - which is (1) the back-end which is the converter itself and (2) the GUI program to make it a user-friendly yet modern system.

Nonetheless, the problem this paper aims to address is the need for a streamlined and secure way to distribute batch files, without the need for dependencies or the risk of unauthorized modifications. Hence, the paper will explore the benefits and limitations of batch file to executable file conversion, and provide best practices throughout the development to leverage this powerful tool.

Input-Process-Output

The Batch2Exe program boasts an intuitive graphical user interface (GUI) that prompts the user to provide various inputs, such as the location of the input batch file, the desired icon for the output executable file, the file path of the output executable file, and the name of the author. Once the user provides this information, the program activates a sophisticated back-end converter tool through a child process. This tool executes a series of complex steps to convert the batch file into a stand-alone executable file.



These steps include compressing the batch file, embedding necessary resources and dependencies, and obfuscating the executable file to protect it from unauthorized access and modifications. Once the conversion process is complete, the output executable file is saved to the specified location on the user's system, providing a secure and self-contained solution for distributing batch files.

Definitions and Terms

This paper aims to provide readers with a clear understanding of key terms and concepts related to batch files, executable files, and the conversion process. Additionally, the section will define specialized terms such as back-end converter, compression, and obfuscated wrapper, which are essential to understanding the technical aspects of the conversion process.

1. *Batch files*: Batch files are plain text files containing a series of commands that are executed by the Windows command prompt. They are used to automate routine tasks and system configurations, and are often used by system administrators and software developers.
2. *Windows executable file*: A Windows executable file is a file that can be run on a Windows operating system. It contains code that can be executed directly by the operating system, and is typically used to distribute software applications, drivers, and other programs.
3. *Front-end and back-end*: Front-end and back-end refer to the two parts of a software application or system. The front-end is the user-facing part of the application, including the graphical user interface (GUI) and input/output features. The back-end is the part of the application that handles data processing, storage, and manipulation, often through a command-line interface (CLI) or an API.
4. *Obfuscation*: Obfuscation is the act of intentionally making code or data more difficult to understand or read, often to protect intellectual property or to prevent unauthorized access or modification. Obfuscation techniques can include code obfuscation, string encryption, and variable renaming.

By defining these terms and providing context for their usage within this paper, readers can gain a deeper understanding of the underlying concepts and technical details of batch file conversion, even if they are not familiar with the subject matter.