

Common Scripting Solutions

In this reading, you will learn about a variety of scripting languages, their uses, and their risks. As an IT Support professional, you may need to automate routine tasks. For example, you might want to automate a backup of company data that runs every night. You might also need to automate high volume tasks, like changing security access settings on thousands of files. Scripting is a common tool used for automation. This tool can help IT Support staff save time and resources in a busy enterprise work environment.

Scripting languages

There are many scripting languages available to use for a variety of tasks in different operating system environments. Most scripts are written in command line environments.

Scripting languages for Windows environments:

- **PowerShell (.ps1)** - Windows PowerShell is among the most common command line scripting tools used in Windows environments. PowerShell is built on the .NET platform and employs many of the same elements that programming languages do. PowerShell scripts are used for building, testing, and deploying solutions, in addition to automating system management.
- **Batch scripts (.bat)** - Batch scripts, also called batch files, have been around since the early days of MS DOS and OS/2. Batch files can execute simple tasks, like calling a set of programs to run when a computer boots up. This type of script could be useful in setting up employees' workspaces when they power on their computers.
- **Visual Basic Script (.vbs)** - Visual Basic Script is an older scripting language. It has reached its end of life for Microsoft support and has been replaced by PowerShell scripts. However, as an IT professional, you may encounter .vbs scripts on some legacy systems.

Scripting languages for Linux and Unix environments:

- **Shell script (.sh)** - Shell scripting languages, like Bash, are used in Unix or Linux environments. The scripts are often used to manipulate files, including changing file security settings, creating, copying, editing, renaming and deleting files. They can also be used to execute programs, print, navigate the operating system, and much more. The scripts run in command-line interpreter (CLI) shells, such as the Bourne shell, Bourne Again SHell (Bash), C shell, and Korn (KSH) shell.

Programming languages that can be used for scripting:

- **JavaScript (.js)** - JavaScript the most used programming language in the world. It is a lightweight language that is used for scripting in web development, mobile and web apps, games, and more. It can also be used to develop software and automate web server functions.
- **Python (.py)** - Python is a user-friendly programming language that can perform advanced tasks and import modules from libraries specially designed for automation scripts.

Scripting uses - finding the right tool for the job

- **Basic automation:** Python is an excellent script for automation. It's one of the most commonly used, with many available automation libraries.
-

Restarting machines: Many power users use PowerShell (.ps1) scripts to restart machines (Windows). For Linux machines, they can use .sh (shell) scripts.

- **Mapping network drives:** In the past, mapping network drives was accomplished with .bat or .vbs scripts. However, PowerShell scripts are most commonly used to map drives in Windows environments today. For Linux users, shell scripts can be used for this purpose.
- **Installing applications:** Batch files and shell scripts are often used for automated software installation.
- **Automated Backups:** Windows PowerShell and Linux/Unix shell scripts can automate backups.
- **Gathering of information and data:** Python is a popular choice for gathering data. Python has many available libraries to help with this task.
- **Initiating Updates:** Powershell and shell scripts can be used for initiating updates in Windows and Linux, respectively.

Security risks of using scripts

IT Support professionals need to be very careful when using scripts, especially with prewritten scripts copied or downloaded from the internet. Some of the security risks of using scripts could include:

- **Unintentionally introducing malware:** As an IT Support professional that is new to scripting, you may try to search the internet for assistance in writing scripts. In your search, you might find a script online for a task that you want to automate. It's tempting to save time and effort by downloading the script and deploying it in your network environment. However, this is dangerous because scripts authored by an unverified source could potentially contain malware. Malicious scripts could have the power to delete files, corrupt data and software, steal confidential information, disable systems, and even bring down an entire network. Malicious scripts can create security weaknesses for the purpose of creating entry points for cybercriminals to penetrate networks. Scripts could also introduce ransomware attacks, which often works by encrypting file systems and then selling the decryption keys for ransom.
- **Inadvertently changing system settings:** Scripts are powerful tools for changing system settings. Using the wrong script can cause the user to inadvertently configure harmful settings. For example, one minor typo in a shell script that sets file permission security in Linux could make confidential files accessible to the world.
- **Browser or system crashes due to mishandling of resources:** Mishandling resources can lead to program crashes in the browser or cause the entire computer to crash. For example, directing too much memory to the browser can overload the computer system.

Key takeaways

A basic knowledge of scripting is an important tool for IT professionals. You may need to improve workflow efficiency by automating basic functions with a scripting language. Some common scripting languages include:

- Windows environments: batch scripts (.bat), Powershell (.ps1), Visual Basic Script (.vbs)

- Linux/Unix environments: shell scripts (.sh)
- Most OS environments: javascript (.js), Python (.py)

Scripts have multiple helpful uses, such as:

- Basic Automation
- Restarting Machines
- Remapping Network Drives
- Installing Applications
- Automating Backups
- Gathering of information/ data
- Initiating Updates

There are risks in using scripts, including:

- Unintentionally introducing malware
- Inadvertently changing system settings
- Browser or system crashes due to mishandling of resources

Resources for more information

For more information about scripting languages, please visit:

- <https://library.naruzkurai.com/link/236#bkmrk-page-title>

Revision #2

Created 2023-05-29 13:18:38 UTC by naruzkurai

Updated 2023-05-29 14:13:00 UTC by naruzkurai