

№ 2.2

Скрипт представляет собой инструмент для автоматической установки и обновления указанного ряда программного обеспечения. Его основная задача – максимально упростить и унифицировать процесс настройки новой рабочей станции или обновления существующей, избавляя пользователя от рутинной ручной установки каждого приложения. Для достижения этой цели скрипт использует два менеджера пакетов: Chocolatey (основной) и Winget (резервный для некоторых позиций).

Перед выполнением скрипта стоит убедиться, что система имеет установленный Winget, который является обязательным условием. Сам скрипт начинает свою работу с этой проверки.

Команда

```
@echo off
```

Отключает эхо – вывод на экран самих команд скрипта. Без этого терминал был бы перегружен служебной информацией от каждого действия, что сделало бы информацию нечитаемой. Это стандартная и необходимая практика для подобного сложного BAT-файла.

Первым делом выполняется проверка на наличие Winget:

```
where winget >nul 2>&1
if errorlevel 1 (
    goto error_winget
)
```

Команда where ищет исполняемый файл winget.exe в системных путях. Вывод команды перенаправляется в никуда >nul, а сообщения об ошибках 2>&1 также подавляются, чтобы не засорять консоль. Если программа не найдена (код ошибки errorlevel 1), выполнение переходит на метку :error_winget, где скрипт сообщает о проблеме и завершает работу. Winget является критически важным элементом, так как он используется для установки Chocolatey, и его отсутствие – это стоп-фактор для всего процесса.

Далее скрипт проверяет, не установлен ли уже менеджер пакетов Chocolatey:

```
where choco >nul 2>&1
if not errorlevel 1 (
    echo Chocolatey уже установлен
```

```
goto install_apps
```

```
)
```

Здесь присутствует та же логика с where: если choco найден и доступен (код ошибки не равен 1), скрипт просто информирует пользователя и сразу переходит к установке приложений, пропуская этап инсталляции самого Chocolatey.

Однако, Chocolatey может быть установлен, но его путь не добавлен в переменную окружения PATH текущего сеанса командной строки. Для этого предусмотрена дополнительная проверка:

```
if exist "C:\ProgramData\chocolatey\bin\choco.exe" (  
    set "PATH=%PATH%;C:\ProgramData\chocolatey\bin"  
    goto install_apps`  
)
```

В этом блоке скрипт ищет исполняемый файл по нужному пути. Если находит, то вручную добавляет этот путь в переменную PATH текущей сессии командой set . Это решение позволяет скрипту работать корректно, даже если Chocolatey был установлен ранее, но не был должным образом настроен для данного терминала.

Если все предыдущие проверки не прошли, скрипт приступает к установке Chocolatey. Это делается с помощью PowerShell, который есть в любой современной системе Windows:

```
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe"  
-NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex  
((New-Object System.Net.WebClient).DownloadString('https://  
community.chocolatey.org/install.ps1'))
```

-ExecutionPolicy Bypass – временно отключает политику безопасности PowerShell, запрещающую выполнение скриптов.

После установки скрипт делает паузу, чтобы дать системе время завершить все фоновые процессы установки и регистрации, прежде чем скрипт попытается использовать новую команду choco.

```
timeout /t 10 /nobreak >nul
```

Далее, из соображений надежности, путь к Chocolatey принудительно добавляется в PATH текущей сессии, после чего выполняется финальная проверка его доступности. Если и после этого choco недоступен, скрипт выдаст ошибку и предложит перезапустить себя, так как для обновления глобальной переменной PATH часто требуется перезагрузка сессии.

Основной блок установки приложений (:install_apps) начинается с определения переменной CHOCO.

```
if exist "C:\ProgramData\chocolatey\bin\choco.exe" (  
    set "CHOCO=C:\ProgramData\chocolatey\bin\choco.exe"  
) else (  
    set "CHOCO=choco"  
)
```

Скрипт пытается использовать абсолютный путь к исполняемому файлу (C:\...\choco.exe), что гарантирует работу даже если с переменной PATH что-то не так. Если по указанному пути файла нет, он отступает на просто choco, надеясь, что система его найдет. Таким образом, скрипт сначала пытается использовать гарантированный путь, а если это невозможно - полагается на стандартный механизм поиска программ в операционной системе.

Далее следует последовательность команд установки. Для большинства приложений используется стандартная команда:

```
"%CHOCO%" upgrade [название_пакета] -y
```

Ключ upgrade используется вместо install, так как он является более универсальным: если пакет уже установлен, он будет обновлен до последней версии, если нет — установлен. Ключ -y автоматически отвечает «да» на все запросы во время установки, делая процесс полностью автоматическим.

Для некоторых приложений, которых нет в репозиториях Chocolatey (например, KNIME, Yandex.Telemost, Sber Jazz, Arc) в скрипте используются другие команды. Например:

```
choco install knime -y 2>nul || winget install -e --id  
KNIMEAG.KNIMEAnalyticsPlatform --accept-package-agreements
```

С помощью таких команд скрипт сначала пробует установить приложение через choco (как более рациональный и быстрый способ), но в случае неудачи прибегает к установке через winget. Такой способ более надежен для полностью автоматической установки. Если Chocolatey по какой-то причине не смог установить пакет (например, пакет устарел или отсутствует в репозитории), скрипт автоматически переключается на использование Winget с явным указанием ID пакета. Флаг -e гарантирует, что будет установлен пакет с точным совпадением ID, что исключает ошибки.

Для установки Yandex.Telemost реализована схема прямого скачивания установочного файла с официального сервера. Команда

`powershell -Command «Invoke-WebRequest...`"

использует PowerShell для загрузки программы по URL <https://telemost.yandex.ru/download-desktop>. Особенностью является использование параметра `-UserAgent 'Mozilla/5.0'`,

который имитирует запрос от веб-браузера, что необходимо для корректной работы с сервером. Файл сохраняется в папку загрузок пользователя (`%USERPROFILE%\Downloads`). Установка выполняется запуском полученного исполняемого файла с параметром `/verysilent`, обеспечивающим полностью автоматический режим без показа окон инсталлятора.

Процесс установки Sber Jazz также использует прямое скачивание, но с предварительным определением пути через переменную `DOWNLOADS_PATH`. Это повышает читаемость кода и упрощает возможное изменение пути в будущем. Программа загружается с <http://dl.salutejazz.ru/desktop/latest/jazz.exe> с выводом статусных сообщений `echo` для информирования пользователя, а благодаря параметру `/S` установка происходит тихо. Оба подхода исключают зависимость от репозиторий пакетных менеджеров, гарантируя получение актуальных версий приложений напрямую от разработчиков.

WSL 2 также устанавливается отдельной командой

```
wsl --install,
```

т.к. эта команда автоматически включает необходимый компонент Windows, скачивает и устанавливает дистрибутив Ubuntu по умолчанию, что кардинально упрощает процесс по сравнению с ручной настройкой.

Скрипт завершается сообщением об успехе и командой `pause`, которая позволяет увидеть результат, после чего окно терминала можно закрыть.

