Руководство по использованию WindowsOptimizer

Основные сценарии использования

Сценарий 1: Базовый запуск для очистки системы

Это самый простой способ использования скрипта для регулярной очистки системы.

```
# Откройте PowerShell от имени администратора cd C:\Tools\WindowsOptimizer .\WindowsOptimizer.ps1
```

Что произойдет:

- 1. Скрипт выведет начальные диагностические данные
- 2. Очистит временные файлы, корзину и prefetch
- 3. Запустит службы Windows Update и Windows Defender
- 4. Выполнит проверку целостности системных файлов (SFC)
- 5. Оптимизирует диски
- 6. Покажет финальный отчет

Время выполнения: 4-7 минут (зависит от объема данных)

Сценарий 2: Запуск перед обновлением системы

Рекомендуется запускать перед крупными обновлениями Windows для освобождения места.

```
# 1. Создайте точку восстановления
Checkpoint-Computer -Description "Перед очисткой и обновлением" -RestorePointType "MOD
IFY_SETTINGS"

# 2. Запустите оптимизацию
.\WindowsOptimizer.ps1

# 3. Проверьте освобожденное место
Get-PSDrive C | Select-Object Used,Free

# 4. Запустите обновление Windows
Start-Process ms-settings:windowsupdate
```

Сценарий 3: Еженедельная автоматическая очистка

Настройте автоматический запуск каждую неделю.

```
# Создание задачи планировщика
$action = New-ScheduledTaskAction -Execute "PowerShell.exe" `
    -Argument "-ExecutionPolicy Bypass -NoProfile -File C:\Tools\WindowsOptimizer.ps1"
$trigger = New-ScheduledTaskTrigger -Weekly -DaysOfWeek Saturday -At 3:00AM
$principal = New-ScheduledTaskPrincipal -UserId "SYSTEM" `
    -LogonType ServiceAccount -RunLevel Highest
$settings = New-ScheduledTaskSettingsSet `
    -AllowStartIfOnBatteries
    -DontStopIfGoingOnBatteries `
    -StartWhenAvailable
    -RunOnlyIfNetworkAvailable
Register-ScheduledTask `
   -TaskName "WindowsOptimizer Weekly Cleanup" `
    -Action $action `
    -Trigger $trigger `
    -Principal $principal `
    -Settings $settings
    -Description "Автоматическая еженедельная очистка и оптимизация системы"
```

Сценарий 4: Очистка после установки игр или ПО

После установки больших приложений или игр часто остается много временных файлов.

```
# 1. Запустите оптимизацию
.\WindowsOptimizer.ps1

# 2. Дополнительно очистите кэш установщиков
Remove-Item -Path "$env:TEMP\*" -Recurse -Force -ErrorAction SilentlyContinue
Remove-Item -Path "C:\Windows\Temp\*" -Recurse -Force -ErrorAction SilentlyContinue

# 3. Очистите папку загрузок (осторожно!)

# Remove-Item -Path "$env:USERPROFILE\Downloads\*" -Recurse -Force -Confirm
```

Сценарий 5: Глубокая очистка для освобождения максимума места

Для максимальной очистки диска выполните расширенную последовательность команд.

```
# 1. Запустите WindowsOptimizer
.\WindowsOptimizer.ps1

# 2. Очистите старые точки восстановления (оставить последнюю)
vssadmin delete shadows /for=c: /oldest /quiet

# 3. Очистите Windows Update кэш
Stop-Service -Name wuauserv -Force
Remove-Item -Path "C:\Windows\SoftwareDistribution\Download\*" -Recurse -Force -Error-
Action SilentlyContinue
Start-Service -Name wuauserv

# 4. Очистите файлы обновлений Windows (осторожно!)
Dism.exe /online /Cleanup-Image /StartComponentCleanup /ResetBase

# 5. Очистите логи Windows старше 30 дней
Get-ChildItem -Path "C:\Windows\Logs" -Recurse -File |
Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-30) } |
Remove-Item -Force -ErrorAction SilentlyContinue
```

▲ Внимание: Эта глубокая очистка может занять 15-30 минут!

Сценарий 6: Проверка производительности до и после

Измерьте эффект от оптимизации.

```
# Создайте функцию для измерения
function Get-SystemPerformance {
    $disk = Get-PSDrive C
    $mem = Get-CimInstance Win32 OperatingSystem
    [PSCustomObject]@{
        Timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
        DiskFreeGB = [math]::Round($disk.Free / 1GB, 2)
        DiskUsedGB = [math]::Round($disk.Used / 1GB, 2)
        MemoryFreeGB = [math]::Round($mem.FreePhysicalMemory / 1MB, 2)
        TotalMemoryGB = [math]::Round($mem.TotalVisibleMemorySize / 1MB, 2)
   }
}
# Измерение ДО
$before = Get-SystemPerformance
Write-Host "`n=== ДО ОПТИМИЗАЦИИ ===" -ForegroundColor Cyan
$before
# Запуск оптимизации
.\WindowsOptimizer.ps1
# Измерение ПОСЛЕ
$after = Get-SystemPerformance
Write-Host "`n=== ПОСЛЕ ОПТИМИЗАЦИИ ===" -ForegroundColor Green
$after
# Сравнение
Write-Host "`n=== РЕЗУЛЬТАТЫ ===" -ForegroundColor Yellow
Write-Host "Освобождено места: $(($after.DiskFreeGB - $before.DiskFreeGB)) GB"
Write-Host "Освобождено памяти: $(($after.MemoryFreeGB - $before.MemoryFreeGB)) GB"
```

Интерпретация результатов

Пример успешного отчета

```
______
WINDOWS 11 AUTO OPTIMIZATION REPORT
______
Start Time: 2025-10-21 15:30:45
______
STEP 1 : Pre-Optimization Diagnostics
System : OS Version: 2009, RAM: 31.9 GB
Disk : Drive C: Free 120.45 GB, Used 350.22 GB
STEP 2 : Cleaning Temporary Files
[OK] User Temp cleaned: 1.2 GB freed
[OK] Windows Temp cleaned: 0.8 GB freed
[OK] Recycle Bin emptied: 0.5 GB freed
STEP 3 : Prefetch Cleanup
[OK] Prefetch folder cleaned: 50 MB freed
STEP 4 : Starting Windows Services
[OK] Windows Update Service started
[OK] Windows Defender Service started
STEP 5 : System File Check
[OK] System integrity verified, no issues found
STEP 6 : Disk Optimization
[OK] Drive C: optimized successfully
STEP 7 : Final Disk Space Check
Disk : Drive C: Free 123.00 GB, Used 347.67 GB
OPTIMIZATION SUMMARY
Total Duration: 4.35 minutes
Space Freed: 2.55 GB 🔽
Optimizations Applied: 7
Status: SUCCESS 🗸
```

Индикаторы успешной оптимизации

Хорошие показатели:

- Space Freed: 1-10 GB нормально для регулярной очистки
- Total Duration: 3-6 minutes оптимальное время выполнения
- Status: SUCCESS все операции завершены успешно
- Нет ошибок [ERROR] в логе

↑ Требует внимания:

- Space Freed: 0 MB или отрицательное значение возможна проблема в скрипте
- Total Duration: >10 minutes медленная система или проблемы с диском
- Наличие [WARNING] некритические проблемы
- Наличие [ERROR] ошибки выполнения, требуется проверка

Анализ логов

Логи сохраняются в C:\Windows\Temp\WindowsOptimizer.log

Просмотр логов

```
# Просмотр последних 50 строк лога
Get-Content "C:\Windows\Temp\WindowsOptimizer.log" -Tail 50

# Поиск ошибок в логе
Select-String -Path "C:\Windows\Temp\WindowsOptimizer.log" -Pattern "ERROR|FAIL"

# Экспорт лога на рабочий стол
Copy-Item "C:\Windows\Temp\WindowsOptimizer.log" "$env:USERPROFILE\Desktop\WindowsOptimizer_$(Get-Date -Format 'yyyy-MM-dd_HHmmss').log"
```

Архивирование логов

```
# Создайте папку для архива логов
$logArchive = "C:\Tools\WindowsOptimizer\Logs"
New-Item -Path $logArchive -ItemType Directory -Force

# Скопируйте лог с датой
$timestamp = Get-Date -Format "yyyy-MM-dd_HHmmss"
Copy-Item "C:\Windows\Temp\WindowsOptimizer.log" "$logAr-chive\WindowsOptimizer_$timestamp.log"

# Удалите старые логи (старше 30 дней)
Get-ChildItem -Path $logArchive -Filter "*.log" |
Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-30) } |
Remove-Item -Force
```

Расширенные параметры (для опытных пользователей)

Модификация скрипта для выборочной очистки

Если вы хотите отключить некоторые функции, откройте WindowsOptimizer.ps1 в текстовом редакторе и закомментируйте ненужные строки:

```
# Например, отключить SFC проверку (долгая операция):
# Найдите строку:
# sfc /scannow

# Закомментируйте ее:
# # sfc /scannow
```

Запуск только определенных функций

Вы можете вручную выполнить отдельные шаги:

```
# Только очистка временных файлов
Remove-Item -Path "$env:TEMP\*" -Recurse -Force -ErrorAction SilentlyContinue
Remove-Item -Path "C:\Windows\Temp\*" -Recurse -Force -ErrorAction SilentlyContinue

# Только очистка корзины
Clear-RecycleBin -Force -ErrorAction SilentlyContinue

# Только запуск служб
Start-Service -Name wuauserv -ErrorAction SilentlyContinue
Start-Service -Name WinDefend -ErrorAction SilentlyContinue

# Только оптимизация диска C:
Optimize-Volume -DriveLetter C -Defrag -Verbose
```

Интеграция с другими инструментами

Использование с CCleaner

```
# 1. Запустите WindowsOptimizer
.\WindowsOptimizer.ps1
# 2. Запустите CCleaner для дополнительной очистки
& "C:\Program Files\CCleaner\CCleaner64.exe" /AUTO
# 3. Проверьте освобожденное место
Get-PSDrive C | Select-Object Used,Free
```

Использование с Windows Disk Cleanup

```
# 1. Запустите WindowsOptimizer
.\WindowsOptimizer.ps1

# 2. Запустите встроенную утилиту очистки диска
Start-Process cleanmgr.exe -ArgumentList "/sagerun:1" -Wait

# 3. Проверьте результат
Get-PSDrive C
```

Рекомендуемая периодичность использования

Интенсивность использования ПК	Рекомендуемая частота
Легкое (офисная работа, браузер)	Раз в 2 недели
Среднее (мультимедиа, игры)	Раз в неделю
Интенсивное (разработка, монтаж видео)	2-3 раза в неделю
Сервер или рабочая станция	Не рекомендуется (используйте специализированные инструменты)

Советы по оптимизации

- 1. Регулярность: Лучше запускать часто, чем ждать критического заполнения диска
- 2. Резервное копирование: Всегда делайте точку восстановления перед запуском
- 3. Закрытие приложений: Закройте все программы перед запуском для лучшего эффекта
- 4. **Проверка SSD:** Для SSD дисков проверяйте здоровье диска регулярно: powershell
 - Get-PhysicalDisk | Get-StorageReliabilityCounter | Select-Object DeviceId, Wear, Temperature
- 5. **Мониторинг:** Следите за освобожденным местом, если оно постоянно отрицательное проверьте систему на вирусы

Дополнительные ресурсы

- Часто задаваемые вопросы (FAQ) (FAQ.md)
- Решение проблем (Troubleshooting) (Troubleshooting.md)
- Установка (Installation.md)

Вопросы? Посетите $FAQ \rightarrow (FAQ.md)$