Hardware / Driver Guard

Этот раздел перенёс в DevOps Tools логику двух старых утилит:

- `Audion Windows Driver Update Blocker`;
- `Audion NVIDIA HDMI DP Audio Killer`.

Старые внешние папки больше не нужны для работы GUI/CLI. Рабочие скрипты теперь лежат внутри проекта:

```
system_core\windows_driver_guard
system_core\nvidia_audio
```

Окно вызывает их через общий service layer, PowerShell программы (`App\AudionPs.exe`) и журнал операций DevOps Tools. Ручной CLI-доступ вынесен в `cli\launcher_hardware.cmd`; главный `launcher_project.cmd` и `launcher_project_ru.cmd` просто открывают этот модульный launcher.

CLI-слой намеренно короткий: это не полное воспроизведение старых TUI, а основные команды из прототипов на случай, если GUI не нужен или недоступен.

Driver Update Blocker

Главный сценарий - запретить Windows Update silently подменять рабочие драйверы.

Рекомендуемый порядок:

1. `Проверить защиту драйверов`.
2. `Блокировать драйверы Windows Update`.
3. Reboot.
4. Снова `Проверить защиту драйверов`.

Команда block делает backup policy-ключей и включает:

- `HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\ExcludeWUDriversInQualityUpdate = 1`;
- policy/current settings для отключения driver wizard search через Windows Update;
- policy для запрета автоматической загрузки device metadata.

Официальная основа: Microsoft документирует `ExcludeWUDriversInQualityUpdate` как Update Policy CSP, который исключает драйверы из Windows quality updates.

NVIDIA Driver Install Restrictions

Это более узкий слой: запретить Windows устанавливать/обновлять драйвер для текущих NVIDIA PCI устройств через Device Installation Restrictions.

Обычный порядок:

1. Поставить вручную нужный NVIDIA-драйвер.
2. `Проверить защиту драйверов`.
3. Оставить `Запрет и для уже установленных` выключенным.
4. `Запретить установку драйверов NVIDIA`.
5. Reboot.

`Запрет и для уже установленных` опасен: Windows может применить запрет к уже установленным matching devices. Обычно он не нужен, если текущий драйвер уже рабочий.

`Include compatible IDs` расширяет matching; включать только если Hardware IDs недостаточно.

Официальная основа: Device Installation Restrictions - документированный policy-механизм Windows. `DenyDeviceIDs` запрещает установку/обновление устройств, чьи hardware ID или compatible ID есть в списке.

Для быстрого применения NVIDIA lock без GUI в корне проекта есть wrapper:

```
cli\Lock-NVIDIA-Driver-Installs.cmd
```

Он вызывает актуальный `Block-NVIDIA-Driver-Updates.ps1`, сам ищет present NVIDIA PCI devices и по умолчанию пишет только NVIDIA Hardware IDs без retroactive removal. Для

автозагрузки или Task Scheduler можно использовать `cli\Lock-NVIDIA-Driver-Installs.cmd --no-pause`. Дополнительные expert-флаги: `--include-compatible` и `--retroactive`.

Custom HWID Install Restrictions

Этот слой делает то же, что NVIDIA restrictions, но без привязки к vendor profile. В поле `Hardware IDs` можно ввести один или несколько HWID:

```
PCI\VEN_8086&DEV_46A8&SUBSYS_22E717AA
PCI\VEN_8086&DEV_46A8&SUBSYS_22E717AA&REV_0C
```

Для текущей Intel Iris Xe GUI показывает один понятный `HWID для защиты`:

```
PCI\VEN_8086&DEV_46A8&SUBSYS_22E717AA
```

`...&REV_0C` не нужно держать в видимом поле обычного blocker. Он остаётся внутренним alias: status/unblock найдут и старую `REV`-запись за счёт prefix match, если она уже была записана старым прототипом.

В GUI это разделено так:

- обычная `HWID-защита драйвера устройства` показывает только информационный badge `Общий` и одно поле `HWID для защиты`;
- аварийный `rank repair` дополнительно показывает warning badge `OEM REV alias ...&REV_0C`, потому что это признак уже пойманного OEM/REV-хвоста, который нужно учитывать при ремонте/cleanup.

Эти Intel Iris Xe значения из исходного прототипа оставлены как встроенный protected cache: GUI показывает кнопку `Intel Iris Xe`, поле `HWID для защиты` заполнено recommended ID по умолчанию, а PowerShell-скрипты используют этот cache как fallback, если HWID не передан. Это не пользовательская история и не runtime cache, который можно случайно очистить через cleanup.

Скрипт пишет только выбранные значения в `DenyDeviceIDs`, сохраняет уже существующие чужие entries и при unblock удаляет только совпадающие HWID. Это безопаснее старого точечного прототипа, который мог снести весь subtree `DenyDeviceIDs`.

Обычный порядок для осознанного обновления драйвера:

1. `Проверить текущий замок`.

2. `1. Разблокировать перед обновлением`.
3. Поставить нужный manual/generic driver.
4. Проверить, что нужная версия стала active/best-ranked.
5. `2. Заблокировать после установки`.
6. Reboot.

По умолчанию targeted block снимает global `ExcludeWUDriversInQualityUpdate`, чтобы не держать всю систему в режиме полного запрета драйверов Windows Update. Если нужен старый глобальный guard, включить advanced option `Оставить общий запрет драйверов WU`.

Для быстрого применения текущего Intel Iris Xe lock без GUI в корне проекта есть wrapper:

```
cli\Lock-Intel-IrisXe-HWID.cmd
```

Он не содержит собственной логики, а вызывает актуальный `Set-HardwareId-DriverInstallRestriction.ps1` с HWID `PCI\VEN_8086&DEV_46A8&SUBSYS_22E717AA`. Для автозагрузки или Task Scheduler можно использовать `cli\Lock-Intel-IrisXe-HWID.cmd --no-pause`, чтобы окно не ждало клавиши после выполнения.

Driver Rank Repair By HWID

Это аварийный сценарий для live-системы, где уже активировался плохой OEM package и перебивает нужный generic driver по rank/version.

Скрипт:

1. Находит present device по введённому HWID или использует вручную заданный `ID экземпляра устройства`.
2. Создаёт preflight backup bundle в `backup\driver_guard\rank_repair\RankRepair_...\preflight_backup`.
3. Проверяет текущую версию драйвера против `Плохая версия драйвера`.
4. Ищет target INF в Driver Store по `Нужная версия драйвера` и HWID; при необходимости можно задать точный `Путь к INF цели`.
5. Экспортирует старый INF package в `backup\driver_guard\rank_repair`.
6. Удаляет old INF package через `pnputil /delete-driver ... /uninstall /force`.
7. Ставит target INF через `pnputil /add-driver ... /install`.
8. Делает rescan/restart устройства и применяет targeted HWID block.

Preflight bundle сохраняет:

- policy REG backups: `DeviceInstall\Restrictions` и `WindowsUpdate`;
- device REG backup: `SYSTEM\CurrentControlSet\Enum\<DeviceInstanceId>`;
- class/driver/service REG backups, если Windows отдаёт `ClassGuid`, driver key и service;
- `device_snapshot.json` с PnP properties, signed-driver data, HWID, class/service markers;
- `before_device_drivers.txt` и `before_class_drivers.txt`;
- `RESTORE_NOTES.txt` рядом с repair log, когда repair доходит до выбора old/target INF.

Для `PCI + Storage/System/ACPI/Bridge`, USB controllers/root hubs и других красных зон этот backup обязателен для ручного анализа. Он не превращает registry import в безопасную кнопку отката, но оставляет точный снимок того, что Windows знала об устройстве до destructive repair.

Для Intel Iris Xe текущие defaults оставлены как удобный preset:

```
Visible HWID:
PCI\VEN_8086&DEV_46A8&SUBSYS_22E717AA

Internal cleanup alias:
PCI\VEN_8086&DEV_46A8&SUBSYS_22E717AA&REV_0C

Bad driver version:    32.0.101.7026
Target driver version: 32.0.101.7085
```

Rank repair - не обычный путь обновления. Для другого железа обязательно сначала выполнить `Проверить цель ранга`, убедиться в выбранном устройстве и версиях, затем при необходимости сузить `Шаблон INF цели` или указать `Путь к INF цели`.

По умолчанию repair теперь останавливается, если нашёл INF с нужной версией, но не нашёл внутри него указанный HWID. Expert fallback `Allow version-only INF fallback` нужен только после ручной проверки конкретного INF. Старый primary INF package должен успешно экспортироваться до `pnputil /delete-driver`; иначе destructive repair не продолжается.

Driver Store Backups

`Сохранить опись Driver Store` сохраняет отчеты без экспорта driver packages в `backup\driver_guard\manifests`:

- `pnputil /enum-drivers`;

- `systeminfo`;
- `Get-WindowsDriver -Online`.

`Экспортировать установленные драйверы` сохраняет third-party INF-based packages из Driver Store в `backup\driver_guard\driver_store` через `Export-WindowsDriver` или DISM fallback.

`Восстановить экспортированные драйверы` запускает:

```
pnputil /add-driver "<backup>\drivers\*.inf" /subdirs /install
```

В GUI restore не интерактивный: backup выбирается из dropdown. Пустой выбор означает самый свежий backup. Ручное поле `Папка копии вручную` можно использовать для папки с `drivers\` или INF-файлами.

Лучшее правило: восстанавливать драйверы с той же машины или очень близкого hardware profile.

NVIDIA HDMI/DP Audio

Этот блок решает отдельную раздражающую проблему: Windows/NVIDIA создают аудиоустройства мониторов и проекторов, которые потом лезут в список playback devices.

Мягкий режим:

1. `Статус звука NVIDIA`.
2. `Экспортировать ID звука NVIDIA`.
3. `Отключить звук NVIDIA HDMI/DP`.

Жесткий режим:

1. `Запретить звук NVIDIA HDMI/DP политикой`.
2. Reboot.

Для быстрого применения этого policy без GUI в корне проекта есть wrapper:

```
cli\Lock-NVIDIA-HDMI-DP-Audio.cmd
```

Он вызывает `Block-NvidiaHdmiDpAudioPolicy.ps1`, сохраняет backup текущей DeviceInstall policy, блокирует только NVIDIA HDAUDIO IDs вида `HDAUDIO\FUNC_01&VEN_10DE...` и отключает

уже установленные matching NVIDIA HDMI/DP audio devices. Для автозагрузки или Task Scheduler можно использовать `cli\Lock-NVIDIA-HDMI-DP-Audio.cmd --no-pause`.

Backup policy перед изменениями сохраняется в `backup\nvidia_audio`, экспорт ID - в `output\nvidia_audio`, state для отката - в `data\nvidia_audio`.

Скрипт специально блокирует только NVIDIA HDAUDIO codec IDs вида:

```
HDAUDIO\FUNC_01&VEN_10DE...
```

Он не блокирует GPU PCI IDs вида `PCI\VEN_10DE...` и не трогает обычные аудиоинтерфейсы вроде Audient, Realtek или Bluetooth Audio.

Что Не Делает Раздел

- Не удаляет WebView/Edge/Photos/Media Player.
- Не скачивает драйверы из интернета.
- Не заменяет OEM updater-приложения Lenovo/NVIDIA/Intel.
- Не гарантирует, что сторонняя утилита производителя не поставит драйвер сама.

Этот раздел ставит admin-policy предохранители вокруг Windows Update и Device Installation, а не превращает систему в immutable image.

Источники Microsoft

- https://learn.microsoft.com/en-us/windows/client-management/mdm/policy-csp-update#excludewudriversinqualityupdate
- https://learn.microsoft.com/en-us/windows/client-management/manage-device-installation-with-group-policy/
- https://learn.microsoft.com/en-us/windows/client-management/mdm/policy-csp-deviceinstallation