

# ОСОБЛИВОСТІ ПРОГРАМНОГО МОДУЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ МЕДІА-КОНТЕНТОМ ДЛЯ ADOBE AFTER EFFECTS

Вінницький національний технічний університет

## **Анотація**

У роботі представлено результати проектування та дослідження ефективності програмного модуля «Workflow Suite Pro» для автоматизації процесів сортування проектних активів та адаптивного кадрювання відеоконтенту в середовищі Adobe After Effects. Обґрунтовано доцільність переходу від застарілого однопотокового середовища ExtendScript до гібридної архітектури Adobe CEP з інтеграцією Chromium Embedded Framework та асинхронного середовища Node.js. Описано принципи роботи двостадійного алгоритму Snapshot-сортування та математичної моделі Fill-scale для неdestructивного ресайзу медіа-файлів. Експериментально доведено лінійну обчислювальну складність алгоритму та високу часову ефективність впровадженого модуля.

*Ключові слова:* комп'ютерна графіка, автоматизація робочого простору, Adobe CEP, Node.js, Snapshot-сортування, Fill-scale, асинхронний аудит.

## **Abstract**

*This paper presents the design and efficiency analysis of the "Workflow Suite Pro" software module for automating project asset management and adaptive video content reframing within Adobe After Effects. The expediency of shifting from the legacy single-threaded ExtendScript environment to the hybrid Adobe CEP architecture with integrated Chromium Embedded Framework and asynchronous Node.js runtime is substantiated. The operational principles of the two-stage Snapshot-sorting algorithm and the Fill-scale mathematical model for non-destructive media resizing are described. The linear computational complexity of the algorithm and the high time efficiency of the implemented module are experimentally proven.*

*Keywords:* computer graphics, workflow automation, Adobe CEP, Node.js, Snapshot-sorting, Fill-scale, asynchronous auditing.

## **Вступ**

Сучасна індустрія цифрового медіа-виробництва характеризується стрімкою фрагментацією форматів відображення відеоконтенту, що вимагає від моушен-дизайнерів адаптації графічних матеріалів під стандарти 1:1, 9:16 та 4:5. У середовищі хост-додатка Adobe After Effects рутинні процеси мануального структурування імпортованих файлів та геометричного перерахунку шарів композицій займають значну частину робочого часу фахівця.

Стандартний інструментарій автоматизації екосистеми Adobe обмежений застарілою специфікацією ExtendScript (ECMAScript 3), яка виконується в єдиному потоці та блокує графічний інтерфейс програми під час обробки важких проєктів. Це зумовлює актуальність розробки високооптимізованого програмного модуля на базі гібридної платформи Adobe CEP (Common Extensibility Platform), що дозволяє використовувати обчислювальний потенціал Node.js та Chromium для фонового керування медіа-активами.

## **Результати дослідження**

Архітектура розробленого модуля автоматизації базується на трикомпонентній моделі взаємодії: клієнтська веб-панель (фронтенд), асинхронний міст Node.js (системний рівень) та ізольоване графічне ядро ExtendScript. Для структурування файлової ієрархії всередині панелі Project було спроектовано та програмно реалізовано двостадійний алгоритм «Snapshot-сортування». На першій стадії алгоритм створює статичний злімок (масив покажчиків) усіх наявних елементів `app.project.items`, фіксуючи їх початковий стан. На другій стадії виконується безпосередній розподіл об'єктів за типами (FOOTAGE, COMPS, AUDIO, IMAGES) у відповідні цільові каталоги. Такий підхід повністю усуває проблему динамічного зсуву індексів елементів під час їх переміщення, що раніше викликало критичні збої та пропуски обробки даних у програмах-аналогах. Для автоматизації процесу ресайзу інтегровано математичну модель Fill-scale. Алгоритм зчитує лінійні габарити вихідної композиції, обчислює максимальний відносний коефіцієнт масштабування під новий цільовий формат

та автоматично генерує керуючий шар-контролер типу Null Object (Null 1). Зв'язування вихідних шарів із нуль-об'єктом реалізує принцип недеструктивного кадрювання, відокремлюючи автоматичні трансформації масштабу від подальшого творчого позиціонування елементів художником. Паралельно в архітектуру інтегровано підсистему асинхронного фонового аудиту подій. За допомогою низькорівневого модуля fs (Node.js) та абсолютної адресації шляхів через системну змінну `__dirname` плагін виконує потоковий запис технічних метаданих у журнал `ae_optimizer_log.txt` без витоків пам'яті. Логіку взаємодії компонентів та виконання операцій автоматизації було верифіковано серією навантажувальних стрес-тестів на реальних медіа-проектах із обсягом внутрішньої структури активів до 2000 елементів. Тестування підтвердило лінійну обчислювальну складність розробленого алгоритмічного забезпечення  $O(n)$ . Часовий хронометраж показав, що повний цикл організації проєкту та кадрювання контенту виконується системою за 18 секунд, порівняно з 14.3 хвилинами мануальної роботи спеціаліста, що забезпечує інтегральний коефіцієнт економії часового ресурсу на рівні 97.9%.

## Висновки

Впровадження розробленого гібридного модуля автоматизації дозволяє кардинально оптимізувати конвеєр підготовки медіа-даних у середовищі Adobe After Effects. Інтеграція асинхронних інструментів Node.js дозволила легітимно вийти за межі браузерної «пісочниці» та реалізувати паралельний аудит і безпомилкове сортування масивів даних без втрати продуктивності графічного додатка. Створене програмне рішення повністю усуває технічні ризики, пов'язані з людським фактором, та є ефективним інструментом для сучасних студій моушен-дизайну та відеовиробництва.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flanagan D. JavaScript: The Definitive Guide: Master the World's Most Used Programming Language. 7th ed. O'Reilly Media, 2020. 704 p.
2. Adobe CEP (Common Extensibility Platform) Resources. Adobe Developer Connection. URL: <https://github.com/Adobe-CEP/CEP-Resources>
3. Node.js File System Module API. Node.js Foundation. 2025. URL: <https://nodejs.org/api/fs.html>

**Романенко Адам Володимирович** – студент групи 5KN-22б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [oknenamor7777@gmail.com](mailto:oknenamor7777@gmail.com)

**Сілагін Єгор Олексійович** – асистент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [yehor.silahin@vntu.edu.ua](mailto:yehor.silahin@vntu.edu.ua)

**Romanenko Adam V.** – student of group 5KN-22b, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [oknenamor7777@gmail.com](mailto:oknenamor7777@gmail.com)

**Silagin Egor O.** – assistant of the Department of Computer Sciences, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [yehor.silahin@vntu.edu.ua](mailto:yehor.silahin@vntu.edu.ua)