SHIN&KIM
법무법인(유) 세종

AI-Assisted Reverse Engineering and Clean-Room Development: Copyright Risks Under Korean Law

2026.11.24

1. Introduction

AI-assisted reverse engineering and clean-room software development are drawing increasing industry attention. Recent examples include REA (Reverse Engineer Anything), an open-source tool that connects AI coding agents with reverse engineering tools, and ArtCraft, a project whose developer claims to have recreated Adobe and Microsoft applications using a clean-room approach.[1]

This newsletter examines the issues each example may raise under Korean copyright law, including the key distinction between reproducing a program's functionality and copying its protected expression, and offers practical considerations for businesses developing software with AI.

2. Background

REA is an open-source tool that enables users to analyze programs with plain-language instructions. For example, a user can ask how a particular feature of a program works, and the AI will use specialized reverse engineering tools to examine the executable file and explain the results.

REA has been described as making sophisticated reverse engineering tools, such as Ghidra and IDA,[2] accessible to developers without specialized expertise. It attracted considerable attention in October 2026, reportedly receiving more than 100,000 GitHub stars in a single week.[3] The developer states in the project's official documentation that REA is not designed to recover original source code or automatically clone applications. Users are advised to use REA only for research or interoperability purposes and only on software they are authorized to analyze. Nevertheless, PC ports of console-exclusive games developed using REA continue to be released, raising copyright concerns.

ArtCraft, meanwhile, is a collection of free alternatives to seven Adobe applications, including Photoshop, and three Microsoft Office applications, developed by a U.S.-based developer using Claude. The developer describes the project as

a “clean-room reimplementation,” claiming that the software was built solely from publicly available documentation and observed behavior without looking at the original source code. The developer also states that neither REA nor Ghidra was used.

Although REA and ArtCraft serve different purposes, both tools are essentially used to reproduce the functions of existing programs in the same or a similar way. The copyright issues that arise depend on the specific activities involved in analyzing and recreating the existing programs. The following sections examine several hypothetical scenarios under Korean copyright law.

3. Potential Copyright Issues

(1) Clean-Room Reimplementation: The ArtCraft Example

(i) Reimplementing software based on publicly available documentation and observations of its behavior without examining the original source code

Computer programs are protected as copyrighted works under Article 4(1)(9) of the Korean Copyright Act (the “Copyright Act”). However, copyright protection does not extend to programming languages, protocols, or algorithms (Article 101-2 of the Copyright Act). Because a program’s functionality and algorithms are not themselves protected by copyright, independently recreating those elements is unlikely to constitute infringement of copyright in the original computer program.

Nevertheless, a claim that a program was developed on a clean-room basis does not, by itself, establish that the development process was independent. It is therefore necessary to maintain development records demonstrating that the developers did not have access to the original source code or reverse engineering results.

(ii) Replicating the original user interface

A program’s user interface (“UI”) must be assessed separately from its underlying code. If the icons, graphics, or screen design contain sufficient originality, their reproduction may constitute copyright infringement.

However, copyright protection may be narrower for elements dictated by functional requirements, such as menu structures or tool arrangements. Each element should therefore be assessed to determine whether it constitutes protectable creative expression.

Potential trademark infringement involving product names or logos, as well as unauthorized use of another party’s business achievements under Article 2(1)(m) of the Korean Unfair Competition Prevention and Trade Secret Protection Act, should also be considered separately.

(2) Reverse Engineering Using REA

(i) Analyzing lawfully purchased software using REA to understand its underlying principles

Under the Copyright Act, reproduction includes temporary storage (Article 2(22)). Accordingly, when an executable file is loaded into a reverse engineering tool and decompiled into human-readable code, storing the original program in memory may constitute reproduction, while converting it into another form of code may constitute the creation of a derivative work (Articles 5 and 22).[4]

However, the Copyright Act provides exceptions that permit certain reverse engineering activities.

First, a person authorized to use a computer program may reproduce it to investigate, study, or test its functions in order to identify the ideas and principles underlying the program (Article 101-3(1)(6)).

Second, a person authorized to use a computer program, or someone acting with that person's permission, may reverse engineer the program code without authorization from the program's copyright holder, but only to the extent necessary to achieve interoperability, where the information required for interoperability is not readily available and obtaining it through reverse engineering is indispensable (Articles 2(34) and 101-4(1)).

Accordingly, analyzing lawfully purchased software to understand how it works is likely to fall within the first exception. However, that exception does not apply where the reproduction unreasonably prejudices the copyright holder's interests (proviso to Article 101-3(1)). In addition, if the software license prohibits reverse engineering, such analysis may constitute a breach of contract, independently of any copyright infringement.

(ii) Developing and distributing replacement software based on reverse engineering results

Developing and distributing replacement software based on reverse engineering results would fall outside the purpose of achieving interoperability, making the exception under Article 101-4 of the Copyright Act unlikely to apply. In this case reproduction during the analysis stage itself could also be found to prejudice the copyright holder's interests.

Moreover, code developed with reference to reverse engineering results is likely to incorporate protected expression from the original program, including its literal code, structure, or arrangement. This could give rise to infringement of the reproduction right, the right to create derivative works, or the right of transmission (i.e., the right to making available to the public).

(iii) Distributing PC ports of reverse-engineered console games

Where a console game is converted for use on a PC while retaining protected expression from the original program, the conversion is likely to infringe the reproduction right, the right to create derivative works, or both. Making the resulting PC port available for download on GitHub or similar platforms could also constitute infringement of the right of transmission.

4. Implications

Rapid advances in AI have made it possible for a single developer working with AI to analyze source code and recreate software in a matter of days, tasks that previously required teams of specialists working for months.

Businesses using AI agents to analyze third-party software should therefore establish internal guidelines defining the programs that may be analyzed and the permissible scope of such analysis. They should also review applicable software licenses for restrictions on reverse engineering before beginning any analysis.

When developing replacement software with AI assistance, businesses should maintain records of the development process, including prompts, work logs showing what materials were provided to AI, and instructions specifying what the AI was prohibited from doing. These records should be retained so that businesses can demonstrate, if necessary, that the original source code was not accessed during development.

[1] Clean-room development is an approach in which developers write original code from scratch based solely on publicly available technical documentation and observable functionality, without viewing the original program's source code.
[2] Ghidra is a free, open-source reverse engineering tool released by the U.S. National Security Agency (NSA) in 2019. IDA is a commercial reverse engineering tool developed by Belgium-based Hex-Rays and widely used as an industry standard in cybersecurity. Both tools offer decompilation capabilities that convert machine code in executable files into human-readable code.
[3] Rather than performing decompilation itself, REA reportedly enables AI agents to use existing reverse engineering tools, such as Ghidra.
[4] Creating a new program using protected creative expression from the original program may infringe the right to create derivative works, even if the programming language is changed or the code is rewritten.

[Korean version] AI 리버스 엔지니어링(REA)과 ArtCraft 사례로 살펴보는 저작권법적 쟁점

Key Contacts

Woo-Gyun Kim
Partner

+82-2-316-4083
wgkim@shinkim.com

Heung-Sik Kim
Advisor

+82-2-316-1985
hsikim@shinkim.com

Jong-Hoon Song
Partner

+82-2-316-2590
johsong@shinkim.com

Seongho Park
Associate

+82-2-316-1980
shopark@shinkim.com

Da Yoon Shin

Senior Foreign Attorney

+82-2-316-1749
dyshin@shinkim.com

Copyright SHIN & KIM LLC. All rights reserved.