A New Licensing Tool For Compiled COMSOL® Applications

D. Ericsson[1]

[1]Deflexional AB, Sweden

Abstract

As the global use of COMSOL® apps continues to grow, many developers are seeking practical and secure methods to protect their applications and control their usage. While traditional paid software often relies on license keys, there has been a lack of simple and flexible tools tailored specifically for COMSOL apps. The Deflexional License Manager addresses this need by offering a developer-friendly licensing system designed for COMSOL applications compiled using COMSOL Compiler™.

Integrating the Deflexional License Manager into a COMSOL app is straightforward. Developers only need to include a single JAR file using the External Java Library functionality in COMSOL Multiphysics®. This JAR file provides all the necessary methods for decrypting and validating license keys. Once added, the developer gains access to the LicenseManager class, which includes a comprehensive set of functions for verifying license status, managing trial periods, and enforcing usage constraints.
License keys and cryptographic secrets are securely generated and managed through a dedicated web interface at https://deflexional.com/license-manager/. The system supports various licensing models, including trial licenses, time-limited licenses, and licenses locked to specific MAC addresses for enhanced control. In addition to license key handling, it includes utility methods for encrypting and decrypting text and files—useful for protecting sensitive data such as proprietary material properties.

To help new users get started, several working examples are provided. Developers with basic programming skills can quickly adapt their existing apps to use the Deflexional License Manager's features. Importantly, developers maintain full control over how license checks are integrated into their workflows.

Despite being a relatively new product, the Deflexional License Manager has already been successfully deployed in commercial projects. Notably, it has been used by Heidelberg Materials in two projects: HETT22 [1], which has thousands of users, and BI Dry [2]. The BI Dry app, which simulates concrete drying behavior, is freely available for download and includes a built-in 7-day trial period managed via the Deflexional License Manager. After the trial expires, users are prompted to enter a valid purchased license key to continue. Another example is Model One, a company that distributes its commercial COMSOL apps using this licensing system.

By offering an easy-to-implement, secure, and flexible licensing solution tailored for COMSOL developers, the Deflexional License Manager enables app creators to monetize and protect their intellectual property while maintaining full compatibility with the COMSOL ecosystem.

Reference

1. D. Ericsson et al., COMSOL Conference 2023 Munich.
2. D. Ericsson et al., COMSOL Conference 2024 Florence.

Figures used in the abstract

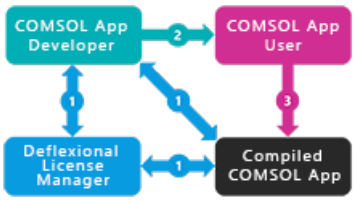

Figure 1 : The COMSOL® app developer integrates the Deflexional License Manager and creates license keys for users via a web-based interface.

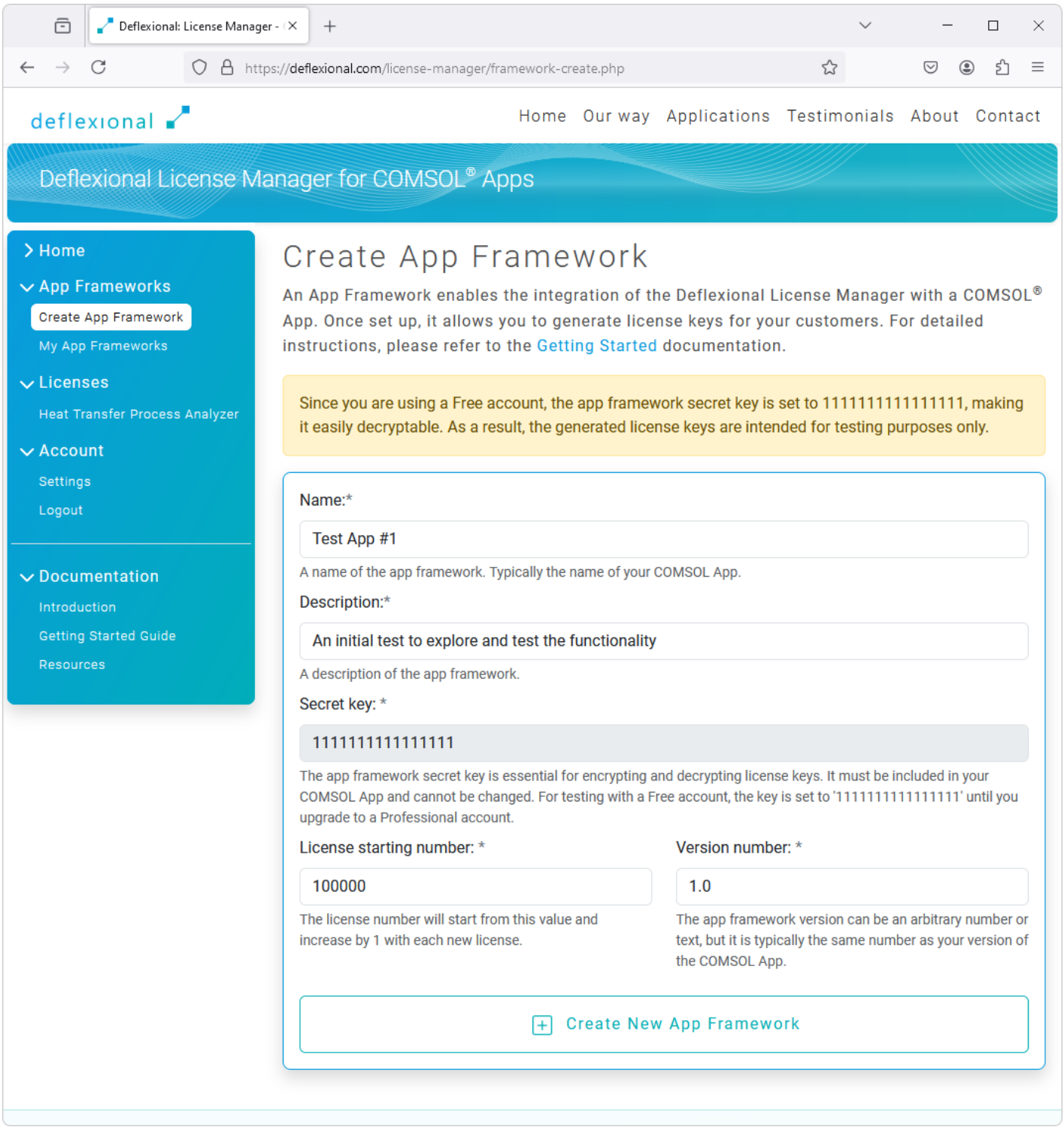

Figure 2 : All projects begin by creating a new App Framework for each COMSOL® app. In the Free version, the secret key cannot be modified.

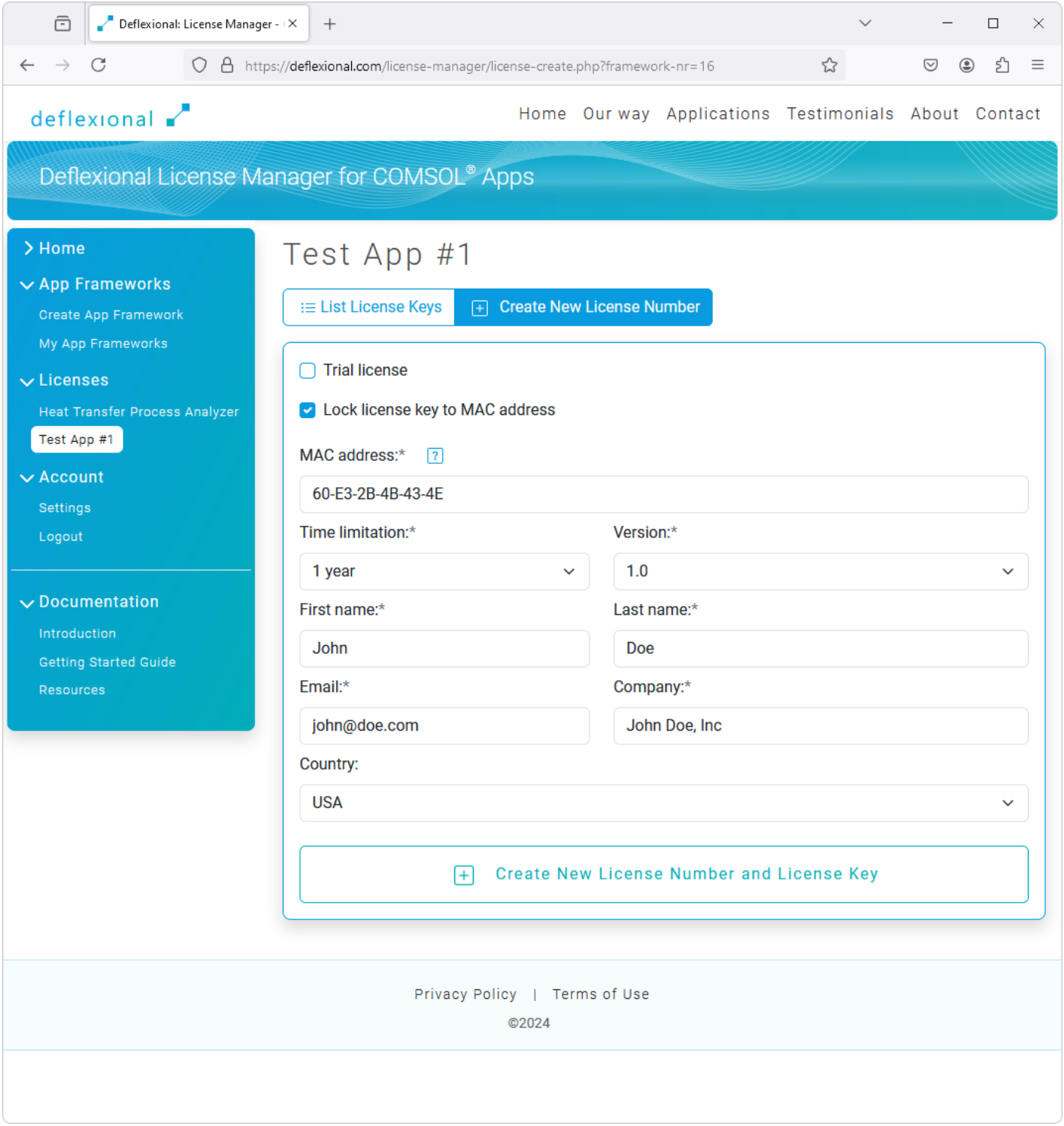

Figure 3 : It's straightforward to generate license keys that can be node-locked, time-limited, or trial-based with feature restrictions.

```
public class licenseIsValid extends ApplicationMethod {

    public void execute() {

        /**
        * Checks if the given license key is valid and if it can be decrypted.
        * This does not include checks for matching MAC address and license expiration.
        *
        * The global string licenseKey is the license key entered by the user.
        * The global string keys.appFrameworkSecretKey is the secret key for the specific app.
        *
        * Sets the boolean isLicenseKeyValid to true if the license key is correct.
        * If the license key is not validated, and warning message is sent to the user.
        */

        boolean isLicenseKeyValid = LicenseManager.isValid(licenseKey, keys.appFrameworkSecretKey);
        if (!isLicenseKeyValid) {
            alert("The license key cannot be validated. Please enter a new license key.", "Warning");
        }
    }
}
```

Figure 4 : This code snippet shows how to use the LicenseManager class in the Application Builder to validate a user-entered license key.