

AxProtector.

Zaawansowane aspekty licencjonowania. Część 2

W pierwszej części artykułu omówiliśmy podstawy działania systemu licencjonowania z wykorzystaniem AxProtector, w tym kluczowe pojęcia takie jak Firm Code, Product Code. Opisaliśmy różnice między podejściem do projektowania licencjonowania opartym na funkcjonalnościach a pakietowym. Mechanizmy te pozwalają elastycznie definiować modele sprzedaży i kontrolować dostęp do poszczególnych funkcji aplikacji.

Druga część koncentruje się na praktycznych aspektach wdrażania licencjonowania – tj. konfiguracji AxProtector, przypisywaniu kodu aplikacji do funkcji i obsłudze licencji przy użyciu WUPI-API.

Jeśli implementujesz ochronę aplikacji i chcesz różnicować zabezpieczenia na poziomie modułów lub funkcji, dowiesz się z niej, jak to zrobić.

AxProtector GUI – Konfiguracja wpisów licencji i list licencji

AxProtector GUI to aplikacja desktopowa, dystrybuowana jak część pakietu CodeMeter SDK. Po jej uruchomieniu przechodzimy do trybu *Expert Configuration*. Po wybraniu pliku aplikacji, który chcemy licencjonować (np. dla .Net core będzie to plik dll), AxProtector analizuje kod i identyfikuje metody wykorzystywane w aplikacji.

W kolejnym kroku wybieramy typ ochrony. Pierwsza opcja definiuje użycie listy licencyjnej 0.

Lista licencji to zestaw licencji, który można przypisać do zabezpieczanych elementów aplikacji (np. klas, metod). Stosowana do modularnego przypisywania różnych warunków licencji do fragmentów kodu – można zdefiniować wiele list (ID: 0, 1, 2...) i przypisywać je do poszczególnych klas/metod.

Możemy wybrać pomiędzy wariantami metod ochrony:

- » Szyfrowanie z licencją
- » Szyfrowanie bez licencji (tryb Unbound)
- » Brak szyfrowania

Wybieramy tryb szyfrowania z licencją. Kolejna opcja określa, czy chcemy szyfrować funkcje (moduły) osobno.

W następnym kroku wprowadzamy kod produktu dla pierwszego wpisu licencji na liście licencji 0. Lista licencji i wpis licencji są tworzone automatycznie.

Dla lepszego zobrazowania posłużymy się dalej prostą aplikacją edytora, która posiada m.in. funkcjonalność zmiany czcionki (*Change-Font*) oraz zapisania dokumentu na dysku (*FileSave*).

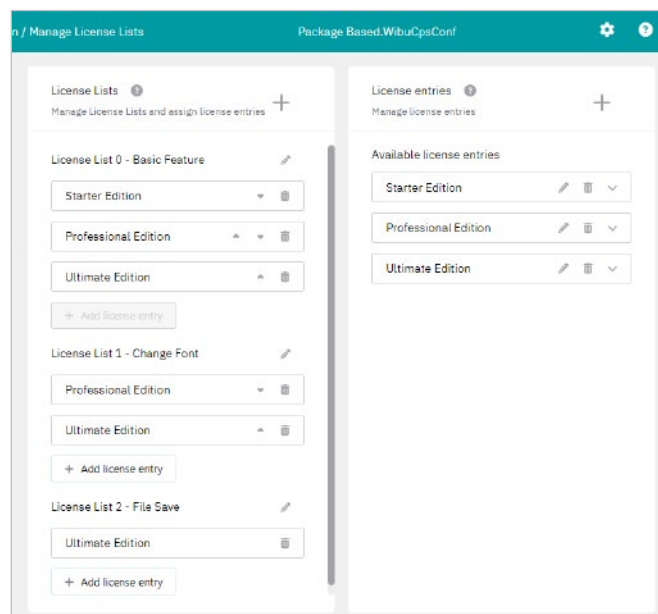
W przypadku wcześniej omawianego podejścia opartego na **pakietach** tworzymy najpierw wpisy dla planowanych rodzajów licencji: *License Entries* (np. Starter, Professional, Ultimate).

Pakiety licencji (*license packages*) to logiczne grupy funkcjonalności oprogramowania, które są sprzedawane jako całość w ramach jednej licencji. Służą one uproszczeniu procesu licencjonowania i oferowania produktu końcowego w różnych wariantach – np. w wersjach podstawowej, rozszerzonej lub specjalistycznej.

Następnie, dla każdej grupy funkcjonalności, tworzymy osobną listę licencji. W naszym przykładzie będą to listy: Basic Feature, Change Font, File Save.

Drugi krok to przypisanie do danej grupy funkcjonalności tych typów licencji, w których powinna ona być aktywowana.

Pakiety są definiowane na potrzeby konkretnego modelu sprzedaży – zawierają zestaw funkcji, które razem tworzą produkt dostosowany do konkretnej grupy odbiorców. Wszystkie funkcje wchodzące w skład pakietu są chronione tym samym kodem produktu. Dzięki temu klient otrzymuje jedną licencję odblokowującą całą zawartość pakietu, a producent ma ułatwione zarządzanie licencjami.



Rysunek 1. Tworzenie wpisów licencyjnych i list licencji w podejściu opartym na pakietach

W przypadku podejścia opartego na **funkcjonalnościach** każdej z nich przypisujemy osobny typ licencji. Tworzymy zatem wpisy licencji: Base License, Change Font License i File Save License oraz odpowiadające im listy licencji: Basic Feature, Change Font i File Save.

Podejście oparte na funkcjonalnościach (*feature-based licensing*) polega na tym, że każda funkcjonalność (np. moduł, metoda lub grupa metod) w aplikacji otrzymuje osobny kod produktu i osobną licencję. To podejście daje dużą elastyczność w zarządzaniu dostępem do poszczególnych elementów oprogramowania.

Następnie mapujemy licencje do list w relacji 1:1.

To podejście pozwala na indywidualne włączanie/wyłączanie konkretnych funkcji aplikacji w zależności od dostępnych licencji.

Możliwe jest również połączenie obu podejść. Przykładem jest licencja na potrzeby demonstracji, pokazów, testów (nie na sprzedaż). Model oparty na funkcjonalnościach rozszerzamy o wpis licencyjny NFR, który następnie dodajemy do wszystkich list licencji wykorzystywanych wewnętrznie.

Przypisywanie list licencji do kodu aplikacji

Skuteczne licencjonowanie to takie, które jest odporne na próby ingerencji w kod aplikacji. Z pomocą przychodzi kryptografia. AxProtector umożliwia **szyfrowanie** osobno każdej z funkcji /metody, za pomocą różnych kluczy kryptograficznych. Odbywa się to poprzez przypisanie list licencji do metod aplikacji.

W oknie ze strukturą kodu w AxProtector GUI przypisujemy metodom/funkcjom tę listę licencji (pakiet), który ją aktywuje. Określamy w ten sposób, który kod należy do której funkcji/modułu/funkcjonalności.

Funkcje w aplikacji można grupować w przestrzeniach nazw lub klasach. Wtedy cała taka grupa (np. przestrzeń nazw lub klasa) przypisywana jest do konkretnej listy licencyjnej. Jeśli jest taka potrzeba, pojedyncze metody lub funkcje można również przypisać indywidualnie.

Dla wygody programista może oznaczyć te przypisania bezpośrednio w kodzie, używając adnotacji. Jeśli jednak dokona zmian ręcznie w interfejsie AxProtectora, to one będą miały pierwszeństwo i nadpiszą adnotacje.

Kod, którego nie da się jednoznacznie przypisać do konkretnej funkcji, powinien trafić do domyślnej listy licencji 0. Natomiast metody, które mają działać bez licencji, można albo przypisać do specjalnej licencji w trybie niezwiązanym (unbound), albo całkowicie pominąć przy szyfrowaniu.

Obsługa licencji z użyciem WUPI-API

Kluczową kwestią w kontekście licencjonowania jest zachowanie aplikacji w sytuacji, gdy licencja dla określonej funkcji nie jest dostępna.

WUPI-API to interfejs programistyczny, który pozwala na sprawdzanie dostępności licencji dla funkcji w aplikacji. Działa na podstawie list licencji zdefiniowanych w AxProtector. W kodzie umieszczane są placeholdery, które po zaszyfrowaniu aplikacji AxProtectorem są zastępowane rzeczywistymi sprawdzeniami licencji (na podstawie Product Code zdefiniowanym w licencji).

Jeśli AxProtector nie zostanie użyty, placeholdery w kodzie pozostają niezmienione i domyślnie zawsze zwracają „true”, czyli pozwalają na działanie funkcji – bez faktycznego sprawdzenia licencji.

Dlatego ważne jest, aby sprawdzenie dostępności licencji dla danej funkcji było wykonywane, zanim dana funkcja zostanie użyta, najlepiej już na etapie startu aplikacji.

Ochrona i przypisywanie licencji do aplikacji mogą być również wykonywane w trybie wsadowym. AxProtectorGUI tworzy plik konfiguracyjny i uruchamia tryb wsadowy w tle. Tryb wsadowy jest używany głównie w potokach CI/CD.

Plik konfiguracyjny jest przechowywany jako plik *.WibuCp-sConf w formacie YAML.

Dalsza eksploracja

Możliwości AxProtector mogą być poszerzone o następujące opcje:

- » **File Encryption:** proste licencjonowanie i ochrona dowolnych plików, takich jak pliki konfiguracyjne lub modele AI.
- » **CodeMoving:** przesyłanie zaszyfrowanego kodu do klucza CmDongle lub CmCloudContainer i wykonywanie go w bezpiecznym środowisku.
- » **Unbound Mode:** używany do części kodu aplikacji, które powinny być chronione przed inżynierią wsteczną, ale uruchamiane bez żadnej licencji.



JANUSZ HRYSZKIEWICZ

janusz.hryszkiewicz@wibu.com

Związany z IT od 25 lat, głównie w zakresie rozwiązań dedykowanych (Microsoft, Java, Python, SAP). Autor kilku rozwiązań dla biznesu, które stały się produktami. Obecnie zaangażowany w zwiększenie cyberodporności oprogramowania, AI oraz systemów wbudowanych.

WIBU
SYSTEMS

CodeMeter – Od kodu do sukcesu

Generuj realne przychody swoim oprogramowaniem dzięki CodeMeter.

- **Wszechstronna monetyzacja:** Licencjonowanie dostosowane do wymagań rynku.
- **Skuteczna ochrona IP:** Innowacyjne szyfrowanie i ochrona integralności.
- **Pełna kompatybilność:** Bezproblemowa integracja z wieloma platformami.
- **Przyszłościowe rozwiązania:** Zaprojektowane, by rosły wraz z Twoimi wymaganiami.

CodeMeter to solidne podstawy i nowe możliwości dla Twojego oprogramowania.

sales@wibu.com
www.wibu.com



Rozpocznij teraz i zamów
CodeMeter SDK
wibu.com/pl/sdk

