

ЛОКАЛІЗАЦІЯ МОБІЛЬНИХ ЗАСТОСУНКІВ НА IOS ЗА ДОПОМОГОЮ SWIFTUI ТА НОВОГО ФОРМАТУ LOCALIZABLE.XCSTRINGS

Вінницький національний технічний університет

Анотація

У роботі розглянуто сучасний підхід до локалізації мобільних застосунків, розроблених за допомогою SwiftUI, з використанням нового формату Localizable.xcstrings. Порівняно з традиційним Localizable.strings, новий формат є більш структурованим, надає вбудовану підтримку pluralization та значно полегшує командну роботу над локалізацією. Для прикладу було використано мобільний застосунок для бронювання квест-кімнат, де реалізовано мультимовний інтерфейс, адаптований до української та англійської мов.

Ключові слова: SwiftUI, локалізація, Localizable.xcstrings, iOS, інтерфейс користувача.

Abstract

This work explores a modern approach to localizing mobile applications developed using SwiftUI through the new Localizable.xcstrings format. Compared to the traditional Localizable.strings method, the new format offers a more structured and centralized system with built-in pluralization support and enhanced collaboration capabilities. A case study is presented based on a quest room booking app with multilingual support in Ukrainian and English.

Keywords: SwiftUI, localization, Localizable.xcstrings, iOS, user interface.

Вступ

Локалізація iOS-застосунків є ключовою для досягнення глобальної аудиторії. Традиційно для рядків інтерфейсу використовувався файл Localizable.strings (по одному на мову), інколи разом з .stringsdict для підтримки множинних форм. Такий підхід працював, але мав обмеження: відсутність гнучких механізмів для плюралізації (потрібен окремий .stringsdict), можливі помилки форматування (пропущені лапки чи крапка з комою ламали збірку), відсутність відслідковування статусу перекладу тощо. Apple усвідомила ці проблеми і впровадила сучасний формат Localizable.xcstrings – так званий String Catalog – починаючи з Xcode 15 (частково доступний ще з Xcode 13). Цей новий формат став рекомендованим підходом для локалізації на платформах Apple. Він являє собою структурований каталог рядків, що об'єднує всі переклади в одному файлі та тісно інтегрується з Xcode. У доповіді розглянуто відмінності між Localizable.xcstrings та класичним Localizable.strings, показано використання SwiftUI з новим підходом, приклади структури .xcstrings (формат JSON), відображення локалізованих рядків у інтерфейсі SwiftUI. [1]

Основна частина

String Catalog (.xcstrings) – це єдиний каталог локалізованих рядків, який замінює і .strings, і .stringsdict файли. Фактично .xcstrings зберігається у форматі JSON (розробнику не потрібно редагувати JSON вручну – натомість Xcode надає візуальний редактор). На відміну від простого списку ключів і рядків у Localizable.strings, формат .xcstrings є набагато більш структурованим: він містить метадані про стан перекладу (новий, потребує перевірки, застарілий, перекладений), коментарі для перекладачів, варіації для різних форм (множини, різні пристрої тощо) та можливість підстановок змінних. Усі підтримувані мови зберігаються в одному файлі замість розділених .lproj-папок, що дозволяє Xcode виконувати перевірки цілісності: відображати прогрес перекладу для кожної мови, автоматично додавати нові ключі в усі мови, попереджати про відсутні чи застарілі переклади. Це значно спрощує командну роботу – розробники і локалізатори завжди бачать актуальний статус, а ризик конфліктів чи помилок форматування мінімізовано. [2]

String Catalogs забезпечують вбудовану підтримку різних варіацій тексту без зміни коду. Зокрема, передбачено механізм Vary by Plural для відмінювання рядків за числом (one, few, many, other тощо) і Vary by Device для варіацій залежно від типу пристрою (iPhone, iPad, Mac, Apple Watch тощо). На

практиці це означає, що одна локалізаційна сутність може містити кілька версій перекладу: наприклад, окремо для “1 елемент” і “2 елементи”, або різні фрази для кнопки на Mac (“Click...”) vs iPhone (“Tap...”). Раніше досягти такого можна було лише через додаткові ключі чи складні правила .stringsdict. Новий підхід суттєво полегшує підтримку множини: достатньо додати варіації One/Other у редакторі Xcode – для більшості мов цього достатньо без зайвої складності. [4]

Статуси перекладів: Xcode автоматично відслідковує стан кожного рядка, позначаючи їх спеціальними мітками в інтерфейсі. Існує чотири основні стани: NEW (новий) – щойно доданий ключ, для якого в поточній мові ще немає перекладу; NEEDS REVIEW (потребує перевірки) – переклад потребує перегляду (наприклад, текст базової мови змінився чи додано нову варіацію і потрібно упевнитись, що старий переклад актуальний); STALE (застарілий) – рядок більше не використовується в коді застосунку; TRANSLATED (перекладено) – переклад актуальний і не потребує уваги. Розробник або перекладач у Xcode бачить ці статуси у вигляді кольорових бейджів поруч із кожним рядком і відсоткових індикаторів завершення для мов. Таким чином, команда одразу може оцінити прогрес локалізації і зосередитися на рядках, які потребують роботи.

Інтеграція з Xcode і командна робота: структура .xcstrings розроблена спеціально для глибокої інтеграції з процесом розробки. Xcode 15 отримав вбудований візуальний редактор локалізацій, де всі рядки зібрані в одному місці (аналогічно до каталогів ресурсів для зображень чи кольорів). При додаванні до проекту String Catalog Xcode генерує файл Localizable.xcstrings. Далі можна увімкнути опцію “Use Compiler to Extract Swift Strings” у налаштуваннях збірки, щоб компілятор автоматично сканував код і інтерфейс на наявність локалізованих рядків [3]. SwiftUI-структури, такі як Text чи Button, коли їм передано літеральний рядок, неявно трактуються як локалізований ключ (SwiftUI автоматично використовує LocalizedStringKey для літералів). Завдяки цьому всі такі рядки збираються в каталог при компіляції. Розробникам більше не потрібно вручну складати списки ключів – Xcode автоматично додає нові знайдені рядки (з базовою мовою) до каталогу. Для командної роботи це величезний плюс: розробник пише UI-код зі SwiftUI, а перекладач або інша частина команди може паралельно працювати над перекладами в тому ж Xcode-проекті або отримати єдиний файл (через систему контролю версій чи експортування). JSON-формат .xcstrings, хоча не призначений для ручного редагування, відкриває можливості для сторонніх інструментів і сервісів: його легше парсити програмно, тому системи на кшталт CI/CD, хмарні платформи перекладу (Lokalise, Phrase тощо) тепер можуть надавати автоматичні перевірки, синхронізацію та звіти про статус локалізації. Загалом, новий підхід робить процес локалізації більш керованим, наочним і надійним, що позитивно впливає на спільну роботу розробників та перекладачів.



```
1 import SwiftUI
2
3 struct ContentView: View {
4     let itemCount = 5
5     var body: some View {
6         VStack {
7             Text("welcome_message") // статичний ключ
8             Text("cart.items.count \(itemCount)") // динамічний рядок з числом
9             Button("action.checkout") { /* ... */ }
10        }
11    }
12 }
```

Рис.1. Приклад використання в SwiftUI

На рис. 1 Text("welcome_message") та "action.checkout" в Button інтерпретуються як локалізаційні ключі. Перший – статичний текст, другий – також ключ для кнопки. Рядок Text("cart.items.count \(itemCount)") містить інтерполяцію \(itemCount) – SwiftUI розпізнає це як намір локалізувати фразу з підстановкою числа. Під час компіляції Xcode з увімкненим вилученням рядків додасть ці ключі до

Localizable.xcstrings. Для інтерпольованого рядка буде сформовано ключ з формат-специфікатором, напр. "%lld items" (тип %lld відповідає цілому числу на 64-бітних системах). Розробник може відкрити .xcstrings у редакторі і побачити згенеровані записи.

Структура .xcstrings (JSON): кожен ключ у файлі має запис зі всіма перекладами. Наприклад, після додавання перекладів для ключа welcome_message файл Localizable.xcstrings може містити наступне (рис. 2):



```
1 {
2   "sourceLanguage": "en",
3   "strings": {
4     "welcome_message": {
5       "comment": "Welcome text on home screen",
6       "localizations": {
7         "en": { "stringUnit": { "state": "translated", "value": "Welcome!" } },
8         "uk": { "stringUnit": { "state": "translated", "value": "Ласкаво просимо!" } },
9         "de": { "stringUnit": { "state": "translated", "value": "Willkommen!" } }
10      }
11    },
12    "cart.items.count %lld items": {
13      "comment": "Cart summary label",
14      "localizations": {
15        "en": {
16          "variations": {
17            "plural": {
18              "one": { "stringUnit": { "state": "translated", "value": "%lld item" } },
19              "other": { "stringUnit": { "state": "translated", "value": "%lld items" } }
20            }
21          }
22        },
23        "uk": {
24          "variations": {
25            "plural": {
26              "one": { "stringUnit": { "state": "translated", "value": "%lld елемент" } },
27              "few": { "stringUnit": { "state": "translated", "value": "%lld елементи" } },
28              "other": { "stringUnit": { "state": "translated", "value": "%lld елементів" } }
29            }
30          }
31        }
32      }
33    }
34  },
35  "version": "1.0"
36 }
```

Рис.2. Приклад файлу Localizable.xcstrings

У наведеному JSON видно, що всі мови зібрані разом: для welcome_message є переклади англійською (en), українською (uk), німецькою (de). Для ключа з кількістю товарів cart.items.count %lld items показано вкладену структуру plural (one/other для англійської та one/few/other для української, враховуючи особливості мов). Кожен переклад має свій стан (translated) і може містити свій текст. За потреби можна додати й коментар – у прикладі коментар пояснює призначення рядка для перекладача. Поле "version": "1.0" вказує версію формату файлу. Весь цей вміст автоматично підтримується Xcode – розробник здебільшого працює з візуальним інтерфейсом замість редагування JSON напряму.

Відображення в інтерфейсі SwiftUI: Після того як переклади додано до .xcstrings, ніяких додаткових кроків у коді для їх використання не потрібно – SwiftUI автоматично підставить локалізований текст. Якщо запустити застосунок на пристрої з мовою Українська, Text("welcome_message") відобразить “Ласкаво

просимо!”), на пристрої з мовою English – “Welcome!” тощо. Це відбувається завдяки тому, що SwiftUI через NSLocalizedStringKey звертається до відповідного рядка в ресурсах локалізації, аналогічно до UIKit-методу NSLocalizedString. Таким чином, розробнику достатньо забезпечити наявність перекладених значень у Localizable.xcstrings, а решту бере на себе фреймворк. За бажання, у SwiftUI Preview можна навіть переглянути інтерфейс різними мовами, встановивши для прев'ю `.environment(\.locale, Locale(identifier: "uk"))`), щоб впевнитися, що перекладені рядки правильно відображаються і вміщуються у дизайн.

Висновки

Запровадження String Catalogs (.xcstrings) знаменує суттєвий крок вперед у локалізації iOS-застосунків. Новий формат поєднує гнучкість і багаті можливості (підтримка плюралізації, варіацій під різні контексти) з інтуїтивною інтеграцією в Xcode. На відміну від застарілого підходу з розрізненими Localizable.strings, де легко було припуститися помилки або втратити несинхронізований переклад, .xcstrings забезпечує єдине джерело правди для всіх мов. Розробники отримують автоматизацію – рядки вилучаються із SwiftUI та коду автоматично, статуси актуальності відслідковуються, а перекладачі можуть працювати в наочному редакторі чи експортувати дані у стандартних форматах (XLIFF) для перекладацьких сервісів. Командна розробка виграє від прозорості: кожен бачить, що перекладено, а що потребує уваги, прогрес локалізації вимірюється і контролюється. Як наслідок, якість локалізованого інтерфейсу зростає, а час на підтримку перекладів зменшується.

На сьогодні формат Localizable.xcstrings в Xcode 15 став стандартом де-факто для проєктів, що цінують багатомовність. Він повністю сумісний з SwiftUI і UIKit, а Apple продовжує удосконалювати інструменти локалізації (наприклад, можливе розширення перевірок узгодженості форматів, інші поліпшення у майбутніх версіях). Рекомендується новим проєктам одразу використовувати String Catalog, а існуючим – здійснити міграцію (Xcode надає зручний майстер міграції з .strings до .xcstrings). Підсумовуючи, сучасний підхід із .xcstrings робить локалізацію в iOS більш доступною і ефективною, що сприяє створенню інклюзивних, зрозумілих для різних користувачів застосунків

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Apple. Localizing with String Catalogs. [Електронний ресурс] – Режим доступу: <https://developer.apple.com/documentation/Xcode/localizing-with-string-catalogs> (дата звернення: 01.06.2025).
2. Apple. Discover String Catalogs. WWDC 2023. [Електронний ресурс] – Режим доступу: <https://developer.apple.com/videos/play/wwdc2023/10152> (дата звернення: 01.06.2025).
3. Apple. Localize your SwiftUI app. WWDC 2021. [Електронний ресурс] – Режим доступу: <https://developer.apple.com/videos/play/wwdc2021/10025> (дата звернення: 01.06.2025).
4. Swiftypace. How to translate and localize iOS app with string catalogs in Xcode 15. [Електронний ресурс] – Режим доступу: <https://swiftyplace.com/blog/localization-xcstrings-xcode15> (дата звернення: 01.06.2025).

Чумак Олексій Віталійович – студент групи ІІСТ-21б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: oleksiychumak2004@gmail.com.

Богач Ілона Віталіївна – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com.

Chumak Oleksii Vitaliyovich – student of IIIST-21b group, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksiychumak2004@gmail.com.

Bogach Ilona Vitaliivna – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com.