Инструкция по реализации средства контрольного суммирования сборки приложения по алгоритму SHA (публикации в App Store и Play Store)

Оглавление

[1. App Store. Процесс подписания кода приложения в iOS 2](#_Toc91754085)

[1.1. Обязательная подпись кода 2](#_Toc91754086)

[1.2. Процесс подписания приложений разработчиками 2](#_Toc91754087)

[2. Google Store. Как использовать функцию подписания приложений 3](#_Toc91754088)

[2.1. Описания ключей, объектов и инструментов 3](#_Toc91754089)

[2.3. Как создать ключ загрузки и обновить хранилища ключей 6](#_Toc91754090)

[2.4. Как обновить ключ подписи для новых установок приложения 7](#_Toc91754091)

[2.6. Рекомендации 8](#_Toc91754092)

[2.6.1. Что делать, если ключ загрузки утерян или взломан 9](#_Toc91754093)

[2.6.2. Схема подписи APK версии 4 9](#_Toc91754094)

# App Store. [Процесс подписания кода приложения в iOS](https://support.apple.com/ru-ru/guide/security/sec7c917bf14/web)

# Обязательная подпись кода

После запуска ядро iOS или iPadOS определяет, какие процессы пользователей и приложения могут быть запущены в системе. Чтобы помочь проверить, что все приложения получены из известного и утвержденного источника и не подделаны, iOS и iPadOS требуют, чтобы весь исполняемый код был подписан с помощью выпущенного компанией Apple сертификата. Установленные на устройстве приложения, такие как Почта и Safari, подписаны Apple. Приложения сторонних разработчиков также должны быть проверены и подписаны с помощью выпущенного компанией Apple сертификата. Обязательная подпись кода расширяет концепцию цепочки доверия с операционной системы на приложения. Она помогает принять меры, чтобы приложения сторонних разработчиков не загружали неподписанные фрагменты кода или не использовали самомодифицирующийся код.

# Процесс подписания приложений разработчиками

Разработчики могут подписывать свои приложения с помощью сертификатов (в рамках программы Apple Developer Program). Они также могут встраивать в свои приложения различные программные среды и использовать сертификаты, выданные Apple, для проверки кода (с помощью строки идентификатора команды).

* *Проверка сертификата.*Для разработки и установки приложений на устройствах iOS или iPadOS разработчики должны зарегистрироваться в Apple и присоединиться к программе Apple Developer Program. Перед выдачей сертификата компания Apple проверяет личность каждого разработчика, будь то частное лицо или компания, в реальном мире. Используя эти сертификаты, разработчики могут подписывать приложения и отправлять их в App Store для распространения. В результате все приложения в App Store отправляются идентифицированными людьми и организациями, что выступает в качестве сдерживающего фактора для создания вредоносных приложений. Кроме того, Apple проверяет все приложения, что помогает выявить явные ошибки или другие заметные проблемы и определить, соответствуют ли приложения своему описанию. Эта проверка дает пользователям дополнительную уверенность в качестве приложений, которые они покупают.
* *Проверка подписи кода.*Разработчики приложений для iOS и iPadOS могут встраивать в свои приложения различные программные среды, используемые самим приложением или встроенными в него расширениями. Чтобы защитить систему и другие приложения от загрузки стороннего кода в их адресное пространство, в момент загрузки система выполняет проверку подписи кода для всех динамических библиотек, на которые ссылается процесс. Эта проверка выполняется с помощью идентификатора команды (Team ID), который извлекается из выпущенного компанией Apple сертификата. Идентификатор команды представляет собой десятизначную буквенно-цифровую строку, например 1A2B3C4D5F. Приложение может ссылаться на любую библиотеку платформы, поставляемую вместе с системой, и любую библиотеку с таким же идентификатором команды в подписи кода, как у основного исполняемого модуля. Поскольку исполняемые модули, поставляемые с системой, не имеют идентификатора команды, они могут ссылаться только на библиотеки, которые также поставлялись с системой.
* *Проверка корпоративных приложений*. У компаний есть возможность разрабатывать корпоративные приложения для внутреннего использования и распространять их среди своих сотрудников. Предприятия и организации могут подать заявку на участие в программе Apple Developer Enterprise Program (ADEP), указав свой номер D-U-N-S. Apple проверяет личность заявителей, их соответствие условиям программы и утверждает заявки. Став членом ADEP, организация может зарегистрироваться для получения [профиля обеспечения](https://support.apple.com/ru-ru/guide/security/aside/secafdef9686/1/web/1), который разрешает запускать собственные приложения компании на указанных в нем устройствах. Для запуска корпоративных приложений у пользователей должен быть установлен профиль обеспечения. Эта мера направлена на то, чтобы только санкционированные организацией лица могли загружать приложения на устройства iOS и iPadOS. Приложения, установленные через систему [управления мобильными устройствами (MDM)](https://support.apple.com/ru-ru/guide/security/aside/sec1a32cff88/1/web/1), по умолчанию считаются доверенными, поскольку взаимосвязь между организацией и устройством уже установлена. В противном случае пользователям необходимо авторизовать профиль обеспечения приложения в Настройках. Организации могут запретить пользователям авторизовывать приложения неизвестных разработчиков. При первом запуске любого корпоративного приложения устройство должно получить подтверждение от Apple, что приложение разрешено запускать.

# Google Store. [Как использовать функцию подписания приложений](https://support.google.com/googleplay/android-developer/answer/9842756?hl=ru#zippy=%2C%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81-%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D1%8F-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9%2C%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%BA-%D0%BA%D0%BB%D1%8E%D1%87%D1%83-%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D0%B8-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F)

Функция подписания приложений в Google Play позволяет Google управлять ключом подписи вашего приложения, а также защищать этот ключ и использовать его для оптимизированных APK-файлов, которые предназначены для распространения и генерируются из наборов App Bundle. Ключ подписи хранится в надежной инфраструктуре Google, и его можно обновить для большей безопасности.

Чтобы использовать эту функцию, вы должны быть владельцем аккаунта или иметь права на [выпуск рабочей версии, исключение устройств и использование функции подписания приложений в Google Play](https://support.google.com/googleplay/android-developer/answer/9844686#release_production). Кроме того, необходимо принять [Условия использования](https://play.google.com/about/play-app-signing-updated-terms/).

**Как это работает?**

Когда у вас включена функция подписания приложений в Google Play, для хранения ваших ключей используется та же безопасная инфраструктура, что и для хранения ключей Google, а их защиту обеспечивает специальный сервис управления ключами. Подробную информацию об инфраструктуре Google можно найти в [документации по безопасности в Google Cloud](https://services.google.com/fh/files/misc/security_whitepapers_march2018.pdf).

Приложения для Android подписываются закрытым ключом. С каждым таким ключом связан открытый сертификат, с помощью которого устройства и сервисы могут проверять источник обновлений. На устройства устанавливаются только те обновления, подпись которых соответствует подписи установленного приложения. Если вы позволите Google управлять ключом подписи приложения, этот процесс станет безопаснее.

Примечание. Для приложений, созданных до августа 2021 года, по-прежнему можно загружать APK-файлы, а не [наборы Android App Bundle](https://developer.android.com/guide/app-bundle/) и управлять собственными ключами, вместо того чтобы использовать функцию подписания приложений в Google Play. Однако если вы потеряете доступ к хранилищу ключей или оно будет взломано, то вы не сможете обновить свое приложение и вам придется опубликовать его заново с другим названием пакета. В таких случаях мы рекомендуем использовать функцию подписания приложений в Google Play и наборы App Bundle.

## Описания ключей, объектов и инструментов

|  |  |
| --- | --- |
| **Термин** | **Описание** |
| **Ключ подписи приложения** | **Ключ, используемый в Google Play для подписания APK-файлов, доставляемых на устройство пользователя. С функцией подписания приложений в Google Play вы можете либо загрузить существующий ключ подписи, либо использовать ключ, сгенерированный Google.**  **Ключ подписи никому нельзя передавать, но при необходимости вы можете поделиться открытым сертификатом приложения.** |
| **Ключ загрузки** | **Ключ, которым вы подписываете набор App Bundle перед загрузкой в Google Play. Этот ключ никому нельзя передавать, но при необходимости вы можете поделиться открытым сертификатом приложения. В целях безопасности для подписания и загрузки приложения рекомендуется использовать разные ключи.**  **Сгенерировать ключ загрузки можно двумя способами:**   * **Используйте ключ подписи приложения. Если вы разрешили Google сгенерировать ключ подписи приложения, ключом загрузки станет ключ, использованный вами для подписания первого выпуска.** * **Используйте отдельный ключ загрузки. Если вы предоставили собственный ключ подписи приложения, то для безопасности можете сгенерировать новый ключ загрузки. Если вы не хотите этого делать, используйте ключ подписи приложения в качестве ключа загрузки для подписания выпусков.** |
| **Сертификат (DER или PEM)** | **Сертификат, который содержит открытый ключ и дополнительную информацию о его владельце. Сертификат открытого ключа позволяет кому угодно узнать, кто подписал набор App Bundle или APK-файл. Этим сертификатом можно делиться, так как он не включает закрытый ключ.**  **Чтобы зарегистрировать ключи у поставщиков API, вы можете скачать открытый сертификат для ключа подписи и ключа загрузки на странице**[**Функция подписания приложений в Google Play**](https://play.google.com/console/developers/app/keymanagement)**(Выпуск > Настройка > Целостность приложения) в Play Console. Сертификатом открытого ключа можно делиться со всеми, так как он не включает закрытый ключ.** |
| **Цифровой отпечаток сертификата** | **Короткий и уникальный идентификатор сертификата. Цифровой отпечаток вместе с названием пакета часто запрашивают поставщики API для предоставления доступа к их услугам.**  **Цифровые отпечатки MD5, SHA-1 и SHA-256 сертификатов загрузки и подписи приложения можно найти на странице**[**Функция подписания приложений в Google Play**](https://play.google.com/console/developers/app/keymanagement)**(Выпуск > Настройка > Целостность приложения) в Play Console. Вы также можете получить цифровой отпечаток другого типа. Для этого скачайте оригинальный сертификат в формате DER на той же странице.** |
| **Хранилище ключей Java (JKS или KEYSTORE)** | **Хранилище сертификатов безопасности и закрытых ключей.** |
| **Инструмент PEPK** | **Инструмент для экспорта закрытых ключей из хранилища Java и их шифрования для передачи в Google Play.**  **Когда вы предоставите Google ключ подписи приложения, выберите экспорт и загрузку собственного ключа (и при необходимости – его открытого сертификата), а затем, следуя инструкциям, скачайте инструмент и воспользуйтесь им. Вы также можете скачать, посмотреть и использовать открытый исходный код инструмента PEPK.** |

**Процесс подписания приложений**

Вот как это работает:

1. Вы подписываете набор App Bundle и загружаете его в [Play Console](https://play.google.com/console).
2. Google оптимизирует APK-файлы из вашего набора, а затем подписывает их с помощью ключа подписи приложения.
3. Google добавляет к манифесту приложения две метки (com.android.stamp.source и com.android.stamp.type), используя инструмент [apksigner](https://developer.android.com/studio/command-line/apksigner), а затем подписывает APK-файлы с помощью ключа подписи приложения. Метки, добавленные через apksigner, позволяют определить, кто подписал APK-файлы.
4. Google доставляет подписанные APK-файлы пользователям.
   1. Как настроить функцию подписания приложений в Google Play

**Инструкции для приложений, созданных после августа 2021 года**

#### Шаг 1. Создайте ключ загрузки

1. Создайте ключ загрузки, следуя [инструкциям](https://developer.android.com/studio/publish/app-signing).
2. Подпишите свой набор App Bundle с помощью ключа загрузки.

#### Шаг 2. Подготовьте выпуск

1. [Подготовьте и внедрите выпуск приложения](https://support.google.com/googleplay/android-developer/answer/9859348), следуя инструкциям.
2. После того как вы выберете тип версии, в разделе "Целостность приложения" появится статус функции подписания приложений в Google Play.
3. Чтобы продолжить с ключом подписи приложения, сгенерированным Google, загрузите набор App Bundle. Вы также можете нажать **Изменить ключ подписи приложения** для доступа к следующим вариантам:
   * **Использовать ключ подписи приложения, сгенерированный Google.** Более чем в 90 % новых приложений используются ключи подписи, созданные Google. Такие ключи нельзя скачать. Это обеспечивает защиту от потери или раскрытия. Если вы выберете этот вариант, то сможете скачать из [**App Bundle Explorer**](https://play.google.com/console/developers/app/bundle-explorer) APK-файлы распространения, которые подписаны с помощью ключа, сгенерированного Google, для других каналов распространения или использовать для них другой ключ.
   * **Использовать другой ключ подписи приложения.** Вы можете использовать ключ подписи другого приложения в вашем аккаунте разработчика (например, если оно предустановлено на некоторых устройствах) или создать локальную копию ключа. В случае взлома копии, хранящейся за пределами серверов Google, повышаются риски безопасности. Если вы хотите использовать другой ключ подписи, доступны следующие варианты:
     + Использовать ключ подписи другого приложения из этого аккаунта разработчика.
     + Экспортировать и загрузить ключ из Java Keystore.
     + Экспортировать и загрузить ключ (не использовать Java Keystore).
4. Выполните оставшиеся действия для [подготовки и внедрения выпуска](https://support.google.com/googleplay/android-developer/answer/9859348).

**Примечание.** Чтобы продолжить, нужно принять Условия использования и включить функцию подписания приложений.

#### Шаг 3. Зарегистрируйте ключ подписи приложения у поставщиков API

Как правило, если в приложении используются API, необходимо зарегистрировать ключ подписи у их поставщиков в целях аутентификации. Для этого нужен цифровой отпечаток сертификата. Вот как найти сертификат:

1. Откройте Play Console и перейдите на страницу [**Функция подписания приложений в Google Play**](https://play.google.com/console/developers/app/keymanagement) (**Выпуск > Настройка > Целостность приложения**).
2. Прокрутите страницу вниз до раздела "Сертификат для ключа подписи приложения" и скопируйте цифровые отпечатки (MD5, SHA-1 и SHA-256) сертификата подписания приложений.
   * Если поставщик API требует другой тип цифрового отпечатка, вы можете скачать оригинальный сертификат в формате DER и преобразовать его с помощью необходимых инструментов.

**Требования к ключу подписи приложения**

Если вы решите использовать ключ, сгенерированный Google, для вас будет автоматически создан криптостойкий 4096-битный ключ RSA. Если вы хотите загрузить собственный ключ подписи приложения, используйте ключ RSA размером не менее 1024 бит.

**Инструкции для приложений, созданных до августа 2021 года**

#### Шаг 1. Настройте функцию подписания приложений в Google Play

1. Откройте Play Console и перейдите на страницу [**Функция подписания приложений в Google Play**](https://play.google.com/console/developers/app/keymanagement) (**Выпуск > Настройка > Целостность приложения**).
2. Если вы ещё этого не сделали, ознакомьтесь с Условиями использования функции подписания приложений в Google Play и нажмите **Принять**.

#### Шаг 2. Отправьте в Google копию оригинального ключа и создайте ключ загрузки

1. Найдите оригинальный ключ подписи приложения.
2. Откройте Play Console и перейдите на страницу [**Функция подписания приложений в Google Play**](https://play.google.com/console/developers/app/keymanagement) (**Выпуск > Настройка > Целостность приложения**).
3. Экспортируйте и загрузите ключ подписи приложения тем способом, который больше всего подходит для вашего процесса выпуска.

#### Шаг 3. Создайте ключ загрузки (необязательно, но рекомендуется)

1. [Создайте ключ загрузки](https://support.google.com/googleplay/android-developer/answer/9842756#create) и загрузите сертификат в Google Play.
   * В качестве ключа загрузки можно также использовать ключ подписи приложения.
2. Скопируйте цифровые отпечатки (MD5, SHA-1 и SHA-256) сертификата подписания приложений.
   * Для проведения тестирования вам может потребоваться зарегистрировать у поставщика API сертификат ключа загрузки с помощью цифрового отпечатка сертификата и ключа подписи приложения.

#### Шаг 4. Подпишите следующее обновление своего приложения ключом загрузки

Выпускаемые обновления приложения нужно подписывать ключом загрузки.

* **Если вы не сгенерировали новый ключ загрузки**, продолжайте использовать для наборов App Bundle оригинальный ключ подписи приложения. Если вы потеряете этот ключ подписи, то сможете [сгенерировать новый ключ загрузки](https://support.google.com/googleplay/android-developer/answer/9842756#create) и зарегистрировать его в Google, чтобы и дальше выпускать обновления.
* **Если вы сгенерировали новый ключ загрузки**, используйте его для подписания наборов App Bundle. Google использует ключ загрузки для проверки вашей личности. Если вы потеряете ключ загрузки, [обратитесь в службу поддержки, чтобы его сбросить](https://support.google.com/googleplay/android-developer/answer/9842756?hl=ru#reset).

## Как создать ключ загрузки и обновить хранилища ключей

В целях безопасности рекомендуем подписывать приложение новым ключом загрузки, а не ключом подписи приложения. Вы можете создать ключ загрузки, когда будете включать функцию подписания приложений в Google Play, или сгенерировать его позже на странице [**Функция подписания приложений в Google Play**](https://play.google.com/console/developers/app/keymanagement) (**Выпуск > Настройка > Целостность приложения**).

Создание ключа загрузки:

1. Следуйте инструкциям на [сайте для разработчиков Android](https://developer.android.com/studio/publish/app-signing#generate-key). Храните ключ в безопасном месте.
2. Экспортируйте сертификат для ключа загрузки в формате PEM. В следующем примере команды вместо аргументов, выделенных подчеркиванием, укажите свои значения:
   * $ keytool -export -rfc -keystore upload-keystore.jks -alias upload -file upload\_certificate.pem
3. Когда в процессе выпуска появится запрос, загрузите сертификат, чтобы зарегистрировать его в Google.

#### **Если вы используете ключ загрузки:**

* Ключ загрузки регистрируется в Google только для аутентификации личности создателя приложения.
* Ваша подпись удаляется из всех загруженных APK-файлов, прежде чем они попадают к пользователям.

**Требования к ключу загрузки**

* Ключом загрузки должен быть ключ RSA размером не менее 2048 бит.

**Обновление хранилищ ключей**

Создав ключ загрузки, проверьте и при необходимости обновите следующие местоположения:

* локальная система;
* защищенный локальный сервер (с различными списками контроля доступа);
* облачная система (с различными списками контроля доступа);
* специальные сервисы по управлению ключами;
* репозитории Git.

## Как обновить ключ подписи для новых установок приложения

В некоторых случаях вы можете запросить обновление ключа подписи приложения. Новый ключ будет использоваться для подписи новых установок и обновлений приложения, а устаревший – для обновлений подписанных им версий, которые уже установлены пользователями. Для каждого приложения ключ подписи можно обновить только один раз. В маловероятном случае, когда вы используете один ключ подписи для нескольких приложений, чтобы запускать их в одном процессе, ключ нельзя обновить.

Запрашивать обновление ключа подписи приложения следует в таких случаях:

* Вам нужен более криптостойкий ключ.
* Ключ подписи приложения взломан.

**Примечание.** Запрос на обновление ключа подписи приложения в Play Console не связан с ротацией ключей в [схеме подписания APK-файлов версии 3](https://source.android.com/security/apksigning/v3) для Android P и более поздних версий. В настоящее время такая ротация ключей не поддерживается Google Play.

Важные примечания, касающиеся обновления ключей

Прежде чем запрашивать обновление ключа, важно понять, какие изменения это повлечет.

* Если вы применяете один ключ подписи для нескольких приложений, чтобы использовать в них одинаковые данные или код, обновите приложения. Тогда они смогут распознавать сертификаты как нового, так и устаревшего ключа.
* Если в приложении используются API, перед публикацией его обновления обязательно зарегистрируйте сертификаты для нового и устаревшего ключей подписи приложения у поставщиков API. Сертификаты доступны на странице [Функция подписания приложений в Google Play](https://play.google.com/console/developers/app/keymanagement) (Выпуск > Настройка > Целостность приложения) в Play Console.
* Пользователи вашего приложения, которые скачивают обновления через файлообменные сети, смогут устанавливать только обновления, подписанные тем же ключом, что и версия приложения на их устройствах. Если приложения нельзя обновить, поскольку установленная версия подписана другим ключом, пользователи могут удалить и переустановить его, чтобы получать обновления.
  1. Как запросить обновление ключа для новых установок

1. Откройте Play Console и перейдите на страницу [Функция подписания приложений в Google Play](https://play.google.com/console/developers/app/keymanagement) (Выпуск > Настройка > Целостность приложения).
2. В карте "Обновите ключ подписи для новых установок приложения" выберите Запросить обновление ключа.
3. Выберите, что сделать с устройством.
   * В зависимости от выбранного варианта, возможно, придется обратиться в службу поддержки, чтобы завершить запрос.
4. Разрешите Google создать новый ключ подписи приложения (рекомендуется) или загрузите его.
   * После обновления ключа подписи приложения, если он совпадал с ключом загрузки, вы можете и дальше использовать прежний ключ подписи приложения в качестве ключа загрузки или сгенерировать новый ключ загрузки.
5. При необходимости зарегистрируйте новый ключ подписи приложения у поставщиков API.

## Рекомендации

* Если вы также опубликовали приложение за пределами Google Play или собираетесь это сделать и при этом хотите использовать тот же ключ подписи, у вас есть два варианта:
  + Google сгенерирует ключ (рекомендуемый вариант), после чего вы сможете скачать из [**App Bundle Explorer**](https://play.google.com/console/developers/app/bundle-explorer) подписанный универсальный APK-файл и использовать его за пределами Google Play.
  + Вы сами сгенерируете ключ подписи приложения, который будете использовать во всех магазинах, и загрузите копию в Google при настройке функции подписания приложений в Google Play.
* Чтобы защитить свой аккаунт, [включите двухэтапную аутентификацию](https://www.google.com/landing/2step/) для всех аккаунтов, у которых есть доступ к Play Console.
* После публикации набора App Bundle в нужной версии вы можете открыть [**App Bundle Explorer**](https://play.google.com/console/developers/app/bundle-explorer) и получить доступ к устанавливаемым APK-файлам, которые Google генерирует из набора. Доступные действия:
  + Скопировать ссылку для [внутреннего доступа к приложению](https://support.google.com/googleplay/android-developer/answer/9844679) и поделиться ей. С помощью этой ссылки можно быстро проверить, какой контент будет установлен из вашего набора App Bundle на то или иное устройство.
  + Скачать универсальный APK-файл, подписанный ключом подписи приложения, который хранится в Google и может быть установлен на любое поддерживаемое вашим приложением устройство.
  + Скачать ZIP-архив со всеми APK-файлами для определенного устройства. Они подписаны ключом подписи приложения, который хранится в Google. Вы можете установить эти APK-файлы на устройство с помощью команды adb install-multiple \*.apk.
* Для большей безопасности сгенерируйте новый ключ загрузки, который отличается от ключа подписи приложения.
* Если вы используете какие-либо API Google, то можете зарегистрировать для своего приложения сертификаты ключа загрузки и ключа подписи в [Google Cloud Console](https://console.developers.google.com/).

## Что делать, если ключ загрузки утерян или взломан

Если вы потеряли доступ к закрытому ключу загрузки или он был взломан, [создайте новый](https://support.google.com/googleplay/android-developer/answer/9842756#create) и попросите владельца вашего аккаунта [обратиться в службу поддержки, чтобы сбросить ключ](https://support.google.com/googleplay/android-developer/contact/key). При обращении в службу поддержки владелец аккаунта должен прикрепить файл upload\_certificate.pem. Когда команда службы поддержки зарегистрирует новый ключ загрузки, вы получите электронное письмо, а затем сможете обновить хранилища ключей и зарегистрировать ключ у поставщиков API. **Важно!** Сброс ключа загрузки не затрагивает ключ подписи приложения, с помощью которого Google Play подписывает APK-файлы перед отправкой пользователям.

## Схема подписи APK версии 4

Устройства с Android 11 и более поздними версиями поддерживают новую [схему подписи APK версии 4](https://source.android.com/security/apksigning/v4). В рамках функции подписания приложений в Google Play будет выпущена подпись версии 4 для выбранных приложений. Благодаря этому в них можно будет использовать предстоящие функции повышения производительности, доступные на новых устройствах. Никаких действий со стороны разработчика не требуется, никакого влияния на пользователя не ожидается.