

# Яндекс Музыка (для UWP)

Документация, содержащая описание технических средств хранения исходного текста и объектного кода программного обеспечения, а также технических средств компиляции исходного текста в объектный код программного обеспечения

6.06.2024

**Яндекс**

Яндекс Музыка (UWP, приложение для Windows). Документация, содержащая описание технических средств хранения исходного текста и объектного кода программного обеспечения, а также технических средств компиляции исходного текста в объектный код программного обеспечения. Версия 1

Дата подготовки документа: 6.06.2024

Этот документ является составной частью технической документации Яндекса.

© 2008—2024 ООО «Яндекс Музыка». Все права защищены.

## **Предупреждение об исключительных правах и конфиденциальной информации**

Исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации юридических лиц, товаров, работ, услуг и предприятий, которым предоставляется правовая охрана (интеллектуальную собственность), используемые при разработке, поддержке и эксплуатации службы Яндекс Музыка (для UWP), включая, но не ограничиваясь, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования, принадлежат ООО «Яндекс Музыка» либо его лицензиарам.

Использование результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации в целях, не связанных с разработкой, поддержкой и эксплуатацией службы Яндекс Музыка (для UWP), не допускается без получения предварительного согласия правообладателя. Настоящий документ содержит конфиденциальную информацию ООО « Яндекс Музыка ». Использование конфиденциальной информации в целях, не связанных с разработкой, поддержкой и эксплуатацией службы Яндекс Музыка (для UWP), а равно как и разглашение таковой, не допускается. При этом под разглашением понимается любое действие или бездействие, в результате которых конфиденциальная информация в любой возможной форме (устной, письменной, иной форме, в том числе с использованием технических средств) становится известной третьим лицам без согласия обладателя такой информации либо вопреки трудовому или гражданско-правовому договору.

Отношения ООО «Яндекс Музыка» с лицами, привлекаемыми для разработки, поддержки и эксплуатации службы Яндекс Музыка (для UWP), регулируются законодательством Российской Федерации и заключаемыми в соответствии с ним трудовыми и/или гражданско-правовыми договорами (соглашениями). Нарушение требований об охране результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации, а равно как и конфиденциальной информации, влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с законодательством Российской Федерации.

## **Контактная информация**

ООО «Яндекс Музыка» <https://www.yandex.ru>

Тел.: +7 495 739 7000

Email: [pr@yandex-team.ru](mailto:pr@yandex-team.ru)

Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

# Содержание

Сведения о хранении и компиляции. ....	4
Хранение исходного кода.....	5
Компиляция исходного кода. ....	5
Структура сервиса в Deploy .....	5
Адрес нахождения технических средств хранения исходного текста и объектного кода программного обеспечения, а также технические средства компиляции исходного текста в объектный код программного обеспечения .....	6

# Сведения о хранении и компиляции

## Хранение исходного кода

Исходные коды программного обеспечения разработаны и контролируются российской организацией ООО «Яндекс Музыка» и находятся в принадлежащих ООО «Яндекс Хостинг» дата-центрах на территории Российской Федерации.

Технические средства хранения исходного кода программного обеспечения контролируются российской организацией ООО «Яндекс Музыка» и находятся в принадлежащих ООО «Яндекс Хостинг» дата-центрах на территории Российской Федерации.

Услуги распределенного хостинга предоставляются ООО «Яндекс Хостинг» по договору.

Для работы над проектом используется система контроля версий Arcadia (Arc). Команда разработчиков взаимодействует с консольным или браузерным инструментом для выгрузки кода на сервер и изменения структуры.

**Arc** — легковесная система контроля версий для монорепозитория, хранящая данные в облаке и использующая виртуализацию рабочей копии вместо скачивания всех данных репозитория. Это позволяет занимать место на диске только для хранения локальных изменений и для оптимизации скорости работы (дисковый кэш).

Для работы с исходными кодами единого репозитория из браузера создан веб-интерфейс Arcanum.

### Особенности единого репозитория

- Исходные коды всех проектов хранятся в одном месте.** Для каждого крупного проекта заводится отдельный каталог, где расположены его исходные коды. Есть специальный каталог для хранения исходных кодов общих внутренних библиотек (library), внешних библиотек (contrib) и экспериментов (junk).
- Trunk-based development.** Весь актуальный код хранится в главной «ветке», которая называется «транк» (англ. trunk - ствол). В определенных случаях от транка могут отводиться дочерние ветки (branches). Все проекты должны отправлять свои коммиты в транк.
- Зеленый транк (green trunk).** Любые изменения перед добавлением в транк проверяются набором тестов. Изменения, для которых тесты не проходят, не добавляются в транк. В Arc добавление изменений организовано через пулл-реквесты (pull-request).
- Герметичность.** Любой из проектов, хранящихся в едином репозитории, собирается только с использованием исходных кодов из этого репозитория. Исходные коды внешних библиотек и описание процесса сборки хранятся в этом же репозитории.

### Уровни использования единого репозитория

В едином репозитории предусмотрено два уровня использования **tier** (англ. tier - уровень):

- Tier 0** — проекты, полностью интегрированные со всеми рекомендуемыми инструментами и технологиями разработки в едином репозитории. Сюда относится большинство проектов на C++, Java, Python и Go.
- Tier 1** — проекты, исходные коды которых хранятся в едином репозитории, но при этом используют собственные инструменты тестирования и сборки. В первую очередь сюда относятся проекты мобильной и фронтенд-разработки.

ПО сервиса Яндекс Музыка (UWP, приложение для Windows) относится к tier 1.

## Компиляция исходного кода

Сервис использует объектный код и средства компиляции.

Исходный текст и объектный код программного обеспечения, а также технические средства компиляции исходного текста в объектный код программного обеспечения контролируются российской организацией ООО "Яндекс Музыка" и находятся в принадлежащих ООО "Яндекс"/ООО "Яндекс Хостинг" дата-центрах на территории Российской Федерации. Услуги распределенного хостинга предоставляются ООО «Яндекс.Хостинг» по договору.

Для компиляции сервиса из исходного кода используются внутренние инструменты Ya.make, Yandex

Яндекс Музыка (для UWP). Документация, содержащая описание

технических средств хранения исходного текста и объектного кода программного обеспечения, а также технических средств компиляции исходного текста в объектный код

Deploy, Runtime Cloud. Ya.make собирает проект из исходного кода, Yandex Deploy управляет релизом проекта, запуская его в контейнерах инфраструктурного контейнерного облака Runtime Cloud.

### Ya.make

Система сборки Ya.make позволяет собирать и тестировать код на четырех основных языках: C++, Python, Java и go. Она имеет развитые средства работы с protobuf. Сборка может исполняться локально с локальным кэшированием, локально с удалённым кэшированием, а также на кластере распределённой сборки.

Система сборки Ya.make:

- **Полностью статическая.** Все зависимости анализируются заранее и изменения фиксируются в графе команд. На основе анализа каждая команда получает уникальный идентификатор UID, который фиксирует её результат на данном состоянии входных данных и зависимостей. Неизменность UID говорит о неизменности её результата и потому служит ключом в кэше результатов, а также используется при анализе изменений для исключения команды из исполнения.
- **Универсальная и высокочувственная.** Описание системы сборки делается на уровне модулей, макросов и зависимостей между модулями.
- **Декларативная.** В описании сборки большая часть конструкций фиксирует свойства модулей и команд и связи между ними. При этом часть конструкций выполняется последовательно: установка и вычисление локальных переменных, условные конструкции — порядок написан в ya.make-файле.

Все инструменты, используемые в сборке зафиксированы в бинарном виде или строятся из исходного кода в рамках сборки. Единственными допустимыми источниками входных данных являются репозиторий Аркадия и хранилище Sandbox, ссылки в которое по ID фиксируются в репозитории. Сам Sandbox гарантирует неизменность данных для ID.

### Deploy и Runtime Cloud

Runtime Cloud — инфраструктурное (внутреннее) контейнерное облако из более чем сотни тысяч серверов расположенных в нескольких геолокациях и обслуживающее десятки тысяч пользовательских сервисов.

## Структура сервиса в Deploy

Самый нижний уровень представляет **Workload** — это приложение, бинарный файл, микросервис, скрипт.

В Workload указываются параметры запуска, проверки готовности и живости процесса, логи, unistat url для сбора метрик, сбор корок и локальные переменные окружения.

Workload расположен внутри **Box** — контейнера, файловой системы, docker образа, porto слоёв. В настройках Box указываются ограничения по ресурсам, общие для Workload переменные окружения, базовые слои вашего приложения (docker/porto), статические и динамические ресурсы. В Box определяются файлы, запуск которых настраивается в Workload<sup>^</sup>.

**Pod (Под)** — это основная и минимальная единица управления деплоя, агрегирующая сущность для Box, Workload. Для Pod доступны такие настройки как FQDN, net/mount/pid ns. Именно Pod<sup>'bi</sup> создаются и запускаются в dataцентрах, настраивается их количество для распределения нагрузки, Pod<sup>'bi</sup> отображаются в UI Deploy и мониторингах, имеют свою ревизию и т.д.

Уровнем выше Box находится **Deploy Unit**, который управляет группами одинаковых Pod в соответствии с выбранной политикой деплоя, сколько Pod<sup>^</sup> с какими ресурсами в каких data центрах запустить в какой последовательности. N одинаковых Pod<sup>^</sup> = Deploy Unit.

**Stage** — группа deploy unit, выкладка которых оркестрируется между собой. Служит, например, для разделения окружений (testing, pre-stable, stable) или объединения разных, но связанных логически Deploy unit (backend + frontend).

**Проект** — это сущность, предназначенная для:

---

### Сведения о хранении и компиляции

- логического разделения сервисов между собой;
- группировки Stage;

- выдачи ролей на все стейджи проекта.

## **Адрес нахождения технических средств хранения исходного текста и объектного кода программного обеспечения, а также технические средства компиляции исходного текста в объектный код программного обеспечения**

Технические средства хранения исходного текста и объектного кода программного обеспечения, а также технические средства компиляции исходного текста в объектный код программного обеспечения контролируются российской организацией ООО «Яндекс Музыка» и находятся в принадлежащих ООО «Яндекс Хостинг» дата-центрах на территории Российской Федерации.

Услуги распределенного хостинга предоставляются ООО «Яндекс Хостинг» по договору.

**ДЦ Владимир Россия, Владимирская обл., г. Владимир, Микрорайон Энергетик, Поисковая улица, дом 1, корпус 2**