



АО "ИТЦ ИТ РОСА"

**СИСТЕМА АВТОМАТИЗАЦИИ СБОРКИ И ПОДДЕРЖКИ ПАКЕТОВ,
РЕПОЗИТОРИЕВ И ДИСТРИБУТИВОВ ПРОДУКТОВ АВР**

Руководство по эксплуатации

Листов 24

2025

Инв. № подл.	Подпись и дата
Взам.инв.№.	Инв. № дубл.
Подпись и дата	Подпись и дата

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ.....	4
2 НАЧАЛО РАБОТЫ С СИСТЕМОЙ.....	6
1.1 Авторизация в системе через веб-интерфейс.....	6
1.2 Главное меню и пользовательский интерфейс.....	6
1.3 Просмотр пользовательских проектов.....	7
1.4 Публикация собранных пакетов.....	9
3 РАБОТА В ВЕБ-ИНТЕРФЕЙСЕ АВФ.....	10
3.1 Запуск новой сборки.....	10
3.2 Этапы сборки проекта.....	11
3.2.1 Настройка доступа к репозиторию.....	11
3.2.2 Клонирование репозитория проекта.....	11
3.2.3 Подготовка файлов для сборки.....	12
3.2.4 Добавление сторонних источников в репозиторий через службу File Store... 12	
3.2.5 Коммиты и отправка изменений на сервер.....	14
3.2.6 Запуск сборки пакета.....	15
3.2.7 Проверка тестов после сборки.....	16
3.2.8 Просмотр сборочных листов.....	17
3.2.9 Завершение сборки.....	18
4 РАБОТА С КОНСОЛЬНЫМ КЛИЕНТОМ АВФ.....	20
4.1 Аутентификация и настройка клиента.....	20
4.2 Получение информации о проекте.....	20
4.3 Управление задачами сборки.....	21
4.4 Загрузка исходников.....	21
4.5 Управление релизами и публикация.....	21
4.6 Дополнительные возможности.....	21

ВВЕДЕНИЕ

Настоящий документ является руководством по эксплуатации программного обеспечения "Система автоматизации сборки и поддержки пакетов, репозитория и дистрибутивов продуктов ABF" (Automatic Build Farm, далее в тексте ABF, Система).

ABF — это распределённая среда непрерывной разработки и сборки программного обеспечения, предназначенная для автоматизации процессов создания, тестирования и распространения дистрибутивов Linux и прикладных программ. Система обеспечивает полный цикл разработки, начиная от управления исходным кодом и заканчивая формированием готовых ISO-образов.

В данном документе представлены сведения о порядке работы пользователя с программным комплексом ABF через веб-интерфейс. Руководство содержит описание основных функций пользовательского интерфейса, включая процедуры авторизации, навигации по главному меню и работе с пользовательскими проектами.

Настоящий документ подготовлен в соответствии с технологической инструкцией: "РОСА. Регламент формирования документации к программным продуктам" РСЮК.11001-01 90 01.

1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Система автоматизированной сборки ABF предоставляет широкий набор инструментов для различных категорий пользователей — разработчиков, мейнтейнеров, администраторов и участников платформ. Ниже представлены ключевые возможности, реализованные в системе.

Общие функции системы:

- интуитивно понятный и функциональный веб-интерфейс;
- возможность участия в групповых проектах;
- гибкая система уведомлений по электронной почте;
- лента активности с актуальными событиями по проектам;
- публичный профиль пользователя с отображением контактной информации и списка публичных проектов.

Для разработчиков:

- интеграция с системой контроля версий git;
- наличие легковесного проектного трекера задач;
- встроенная проектная вики;
- возможность редактирования исходных файлов непосредственно через веб-интерфейс;
- гибкое управление правами доступа, в том числе создание пользовательских групп;
- поддержка как публичных, так и приватных проектов;
- комментирование коммитов;
- просмотр истории изменений (log), сравнений (diff) и аннотаций (blame) через веб-интерфейс;
- возможность клонирования (форка) публичных проектов в собственное пространство.

Для мейнтейнеров:

- импорт исходных кодов из .src.rpm через веб-интерфейс;
- сборка пакетов в индивидуальный частный репозиторий, доступный другим пользователям ABF;

- мониторинг состояния сборочных заданий с возможностью фильтрации по владельцу или статусу;
- подробные журналы (логи) выполнения сборок;
- установка собранного пакета для тестирования до его публикации в общий репозиторий;
- одновременная сборка под различные архитектуры и платформы;
- настройка подключаемых репозиториях платформ на уровне отдельной сборки;
- возможность отмены запущенной сборки;
- автоматическая генерация команд подключения к частному репозиторию;
- безопасная сборка на изолированных контейнерах в распределённой инфраструктуре.

Для администраторов:

- управление пользователями системы, включая их блокировку;
- ведение журнала действий и проведение полного аудита операций в системе.

Для владельцев и участников платформ:

- поддержка собственных дистрибутивов в рамках системы ABF;
- управление составом репозиториях (включение/исключение проектов);
- сборка продуктов (например, ISO-образов).

В экспериментальном режиме реализована поддержка сборки пакетов с учётом их обратных зависимостей. Эта возможность позволяет при обновлении одного пакета автоматически инициировать повторную сборку всех зависимых, что снижает риск нарушения согласованности и повышает надёжность финального дистрибутива.

2 НАЧАЛО РАБОТЫ С СИСТЕМОЙ

2.1 Авторизация в системе через веб-интерфейс

Для начала работы с ABF пользователь должен авторизоваться в системе. Для этого необходимо:

1. Открыть веб-браузер и перейдите по ссылке <https://abf.io/> для входа в систему ROSA ABF;
2. Ввести имя пользователя и пароль;
3. Нажмите кнопку **Войти**.

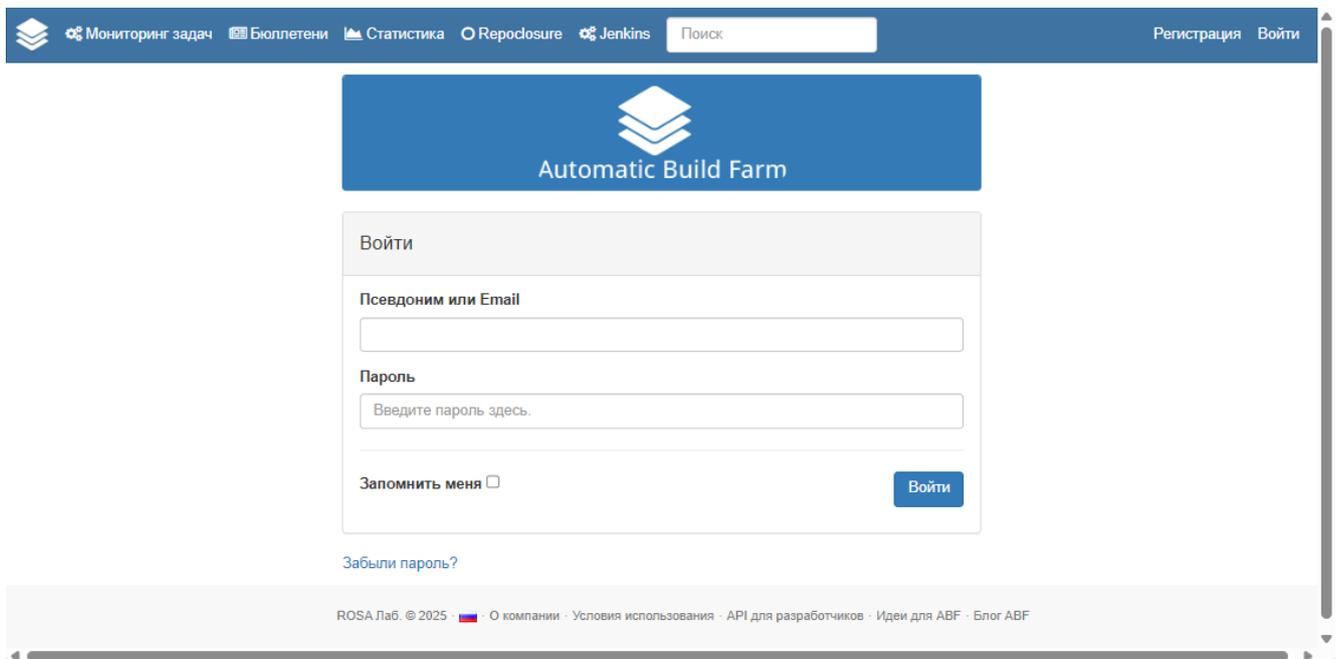


Рисунок 1: Форма авторизации пользователей в веб-интерфейсе ABF

2.2 Главное меню и пользовательский интерфейс

После успешной авторизации пользователь попадает в главное окно системы. Основные элементы интерфейса:

- верхнее навигационное меню с основными разделами системы, панелью пользователя и кнопкой выхода из системы;

— центральная рабочая область для отображения активности пользователя в выбранном разделе системы.

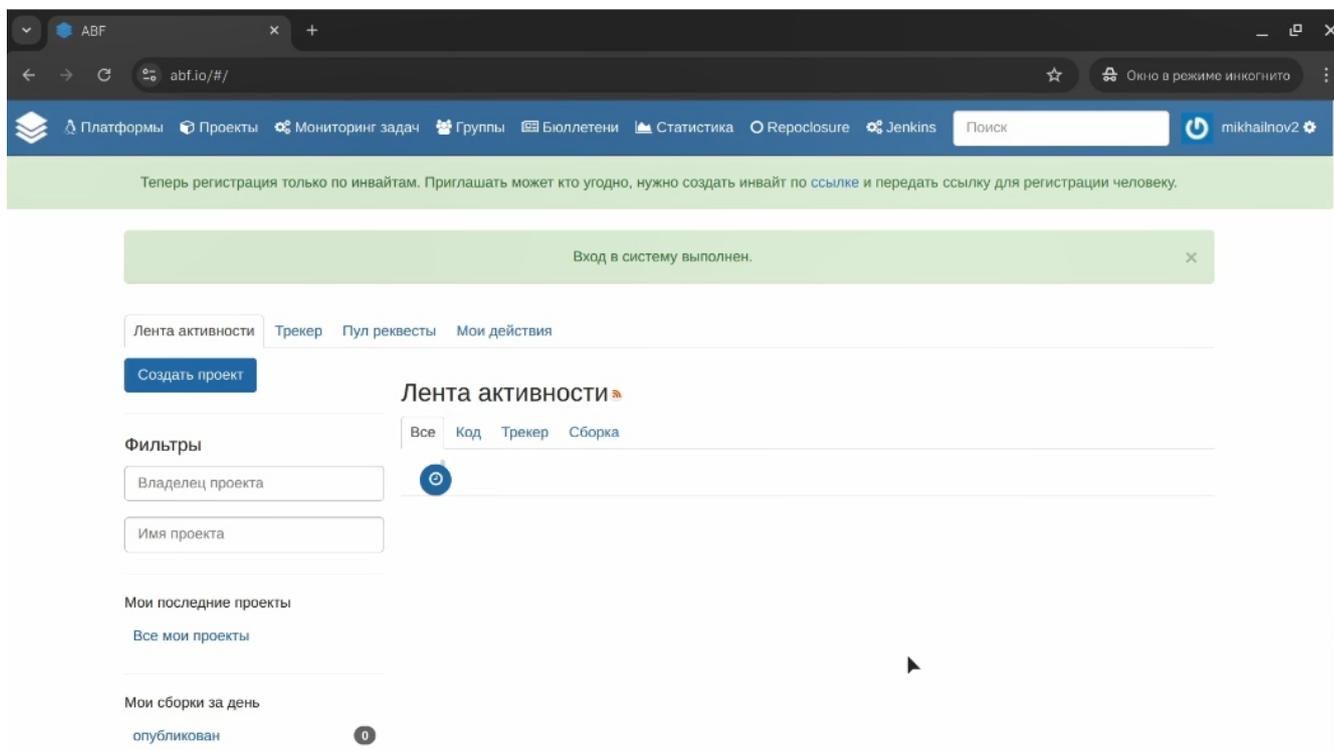


Рисунок 2: Главное окно системы после входа, основное меню и активные разделы

2.3 Просмотр пользовательских проектов

В разделе **Платформы** (на верхней панели меню) представлен список пользовательских репозиториев.

Для просмотра списка всех пользовательских репозиториев:

1. Откройте раздел **Платформы** в верхнем навигационном меню;
2. На экране отобразится список всех доступных проектов с указанием типа дистрибутива;
3. Для перехода в конкретный проект — нажмите на его имя.

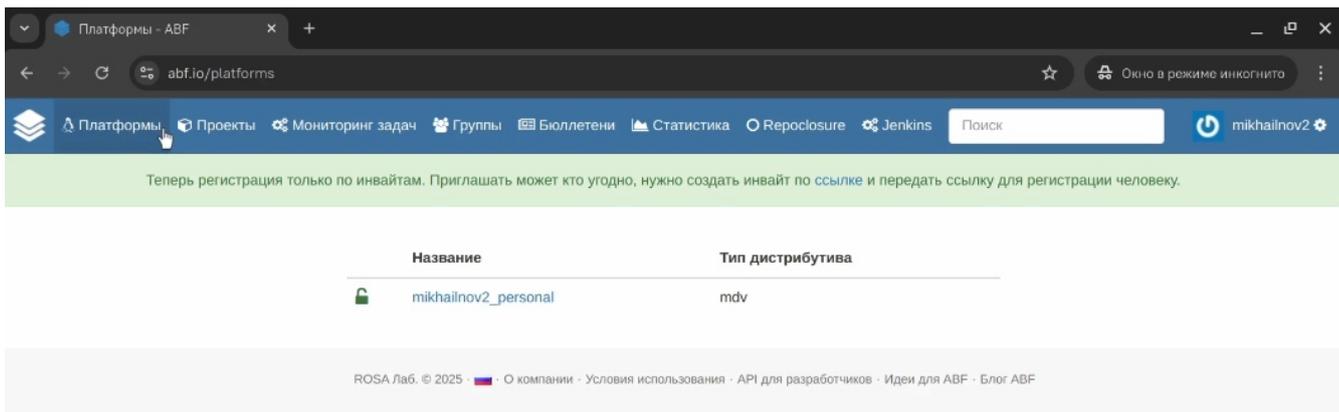


Рисунок 3: Просмотр проектов пользователя

При регистрации личного пользователя по умолчанию создается платформа, которая имеет название в формате **логин пользователя_personal**.

Для просмотра подробной информации о содержимом репозитория и его настроек перейдите в конкретную платформу по нажатию на имя платформы. Для каждого репозитория видны ветки, история изменений и теги.

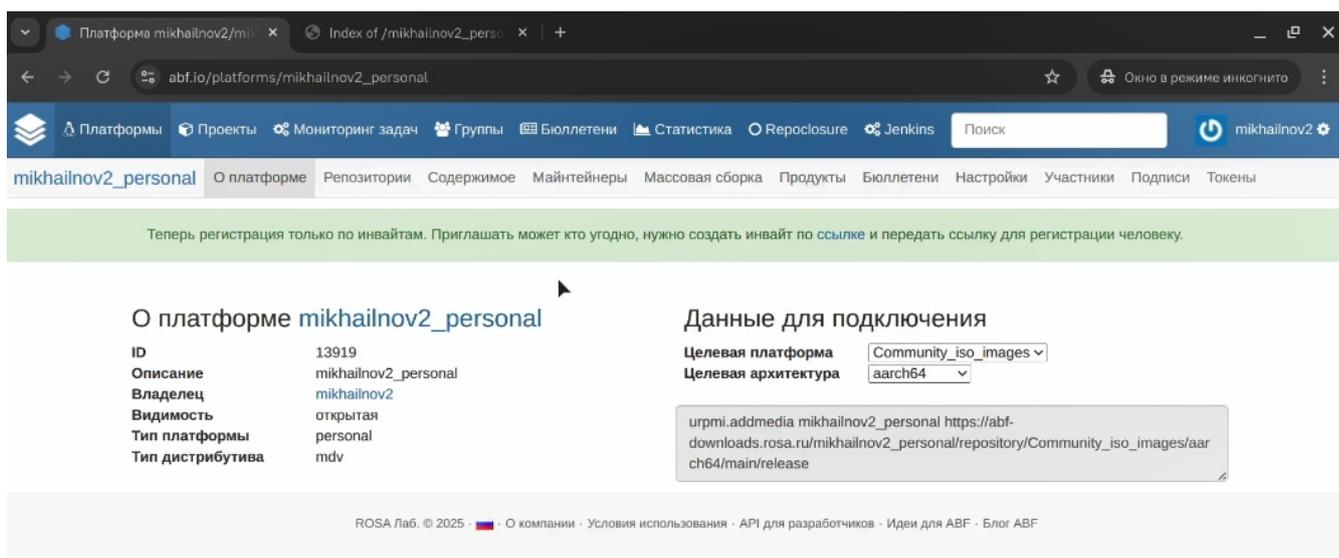


Рисунок 4: Просмотр подробной информации о платформе

Для просмотра всех пакетов, которые будут опубликованы на соответствующих ресурсах, перейдите во вкладку **Содержимое**.

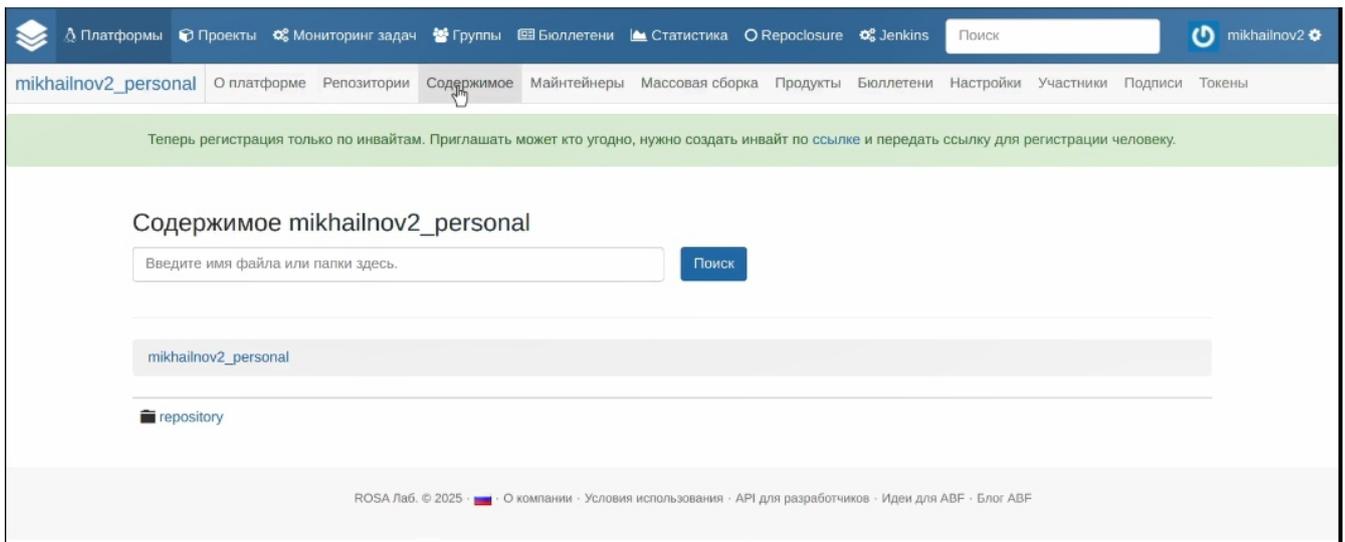


Рисунок 5: Просмотр содержимого текущего репозитория

2.4 Публикация собранных пакетов

Адрес ресурса, на который происходит публикация репозитория, то есть тех пакетов, которые уже собраны, имеет следующий формат:

abf-downloads.rosa.ru/(или иной адрес данной копии abf)/название_платформы



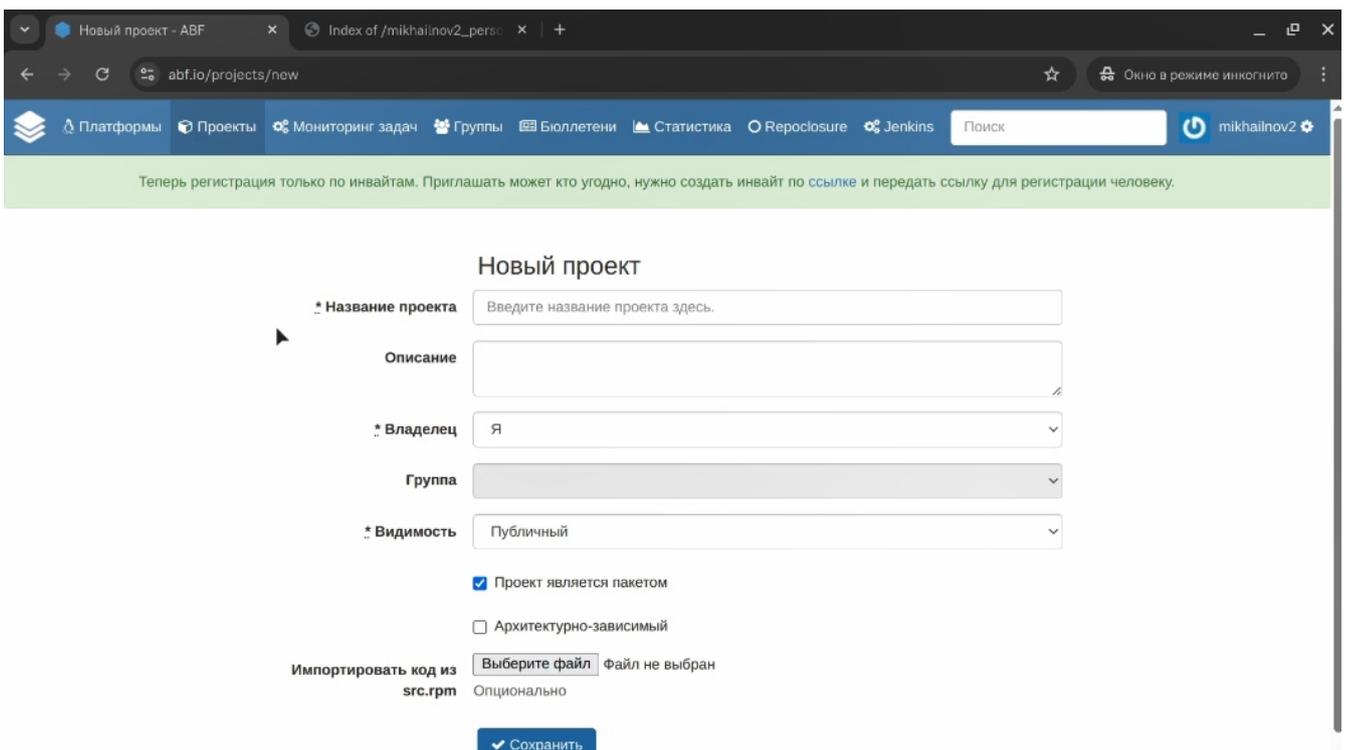
Рисунок 6: Публикация собранных пакетов

3 РАБОТА В ВЕБ-ИНТЕРФЕЙСЕ ABF

3.1 Запуск новой сборки

Для начала работы по созданию репозитория необходимо создать новый проект и настроить git-репозиторий. Для этого:

1. Перейдите на вкладку **Проекты** → **Новый проект**;
2. Заполните форму создания сборки. Обязательные для заполнения поля отмечены символом звездочки (*);
3. Нажмите на кнопку **Сохранить**, для того, чтобы продолжить процесс создания проекта.



The screenshot shows a web browser window with the URL `abf.io/projects/new`. The page title is "Новый проект". The form contains the following elements:

- Название проекта** (Project Name): A text input field with a placeholder "Введите название проекта здесь." and a red asterisk indicating it is required.
- Описание** (Description): A text area for entering project details.
- Владелец** (Owner): A dropdown menu with "Я" (Me) selected.
- Группа** (Group): A dropdown menu.
- Видимость** (Visibility): A dropdown menu with "Публичный" (Public) selected.
- Project is a package**: A checked checkbox.
- Architecture-dependent**: An unchecked checkbox.
- Import code from file**: A section with a "Выберите файл" (Select file) button and the text "Файл не выбран" (File not selected). Below it, "srs.rpm" is listed as an optional file.
- Сохранить** (Save): A blue button with a checkmark icon.

Рисунок 7: Создание нового проекта

После создания будет сгенерирован пустой git-репозиторий, который необходимо заполнить:

- добавить `.sres`-файл (описание пакета);
- добавить файл `.abf.yml` (описание сторонних источников в репозитории, если они есть).

3.2 Этапы сборки проекта

Для примера рассмотрим создание проекта с именем **test1**:

3.2.1 Настройка доступа к репозиторию

Взаимодействие с git-репозиторием может осуществляться через протоколы HTTP и SSH.

Настройка доступ через SSH протокол выполняется следующим образом:

- Создайте ключевую пару с помощью команды:

```
ssh-keygen
```

- Добавьте открытый ключ в разделе **Настройки** → **SSH-ключи** в меню платформы ABF;

- После добавления ключа возможна работа с репозиторием через команду `git clone ssh://<адрес_репозитория>`

3.2.2 Клонирование репозитория проекта

Перед внесением изменений в проект и запуском сборки необходимо клонировать соответствующий репозиторий. Клонирование — это обязательный этап подготовки, позволяющий получить локальную копию проекта для редактирования его содержимого.

Система ABF использует git-репозитории для хранения и управления исходными файлами пакетов. Все изменения, необходимые для сборки (модификация .spec-файлов, обновление патчей, добавление новых исходников и др.), должны быть зафиксированы в Git-коммитах и отправлены на сервер (push).

Для этого репозиторий проекта клонируется локально с использованием git-клиента:

```
git clone https://abf.rosalinux.ru/<пользователь или группа>/<имя-проекта>.git
```

После клонирования разработчик может вносить изменения в проект, создавать коммиты и отправлять их в репозиторий. Только после этого становится возможен запуск сборки с использованием актуального состояния проекта.

Клонирование не требуется только в случае повторной сборки проекта без внесения изменений. Однако при любой модификации файлов — клонирование обязательно.

Пример клонирования публичного проекта:

```
git clone https://abf.rosalinux.ru/rosa/hello.git
```

После выполнения команды локально создается директория с копией проекта, в которой можно производить все необходимые изменения перед сборкой.

3.2.3 Подготовка файлов для сборки

Создайте в локальном репозитории файл с расширением `.spec` (например, `test1.spec`). Название файла, как правило, совпадает с названием проекта, но это не обязательное требование.

Дополнительно создайте файл `.abf.yml` с описанием исходных файлов, необходимых для сборки. Пример строки в `.abf.yml`:

```
sources:
```

```
- имя_файла.tar.gz sha1:1234567890abcdef...
```

Исходные файлы, которые необходимо использовать для сборки, прописываются в параметре `source0`. Обратите внимание на то, что все указанные исходные файлы при сборке должны быть заранее загружены в службу File Store, дополнительно загружаться со сторонних ресурсов в Интернете они не будут.

Файл `.abf.yml` содержит информацию о бинарных файлах, используемых в проекте. Рекомендуется периодически проверять и очищать секцию `removed_sources` от ненужных файлов, чтобы ускорить сборку проектов и снизить нагрузку на файловое хранилище.

3.2.4 Добавление сторонних источников в репозиторий через службу File Store

Служба File Store является неотъемлемой частью платформы ABF, она используется для хранения сторонних исходных файлов, которые не включаются в git-репозиторий. Загрузка файлов осуществляется:

- через веб-интерфейс службы по адресу <https://file-store.rosa.ru> по кнопке **Add file** (авторизация по логину и паролю от ABF);
- с помощью командной утилиты `abf-put`.

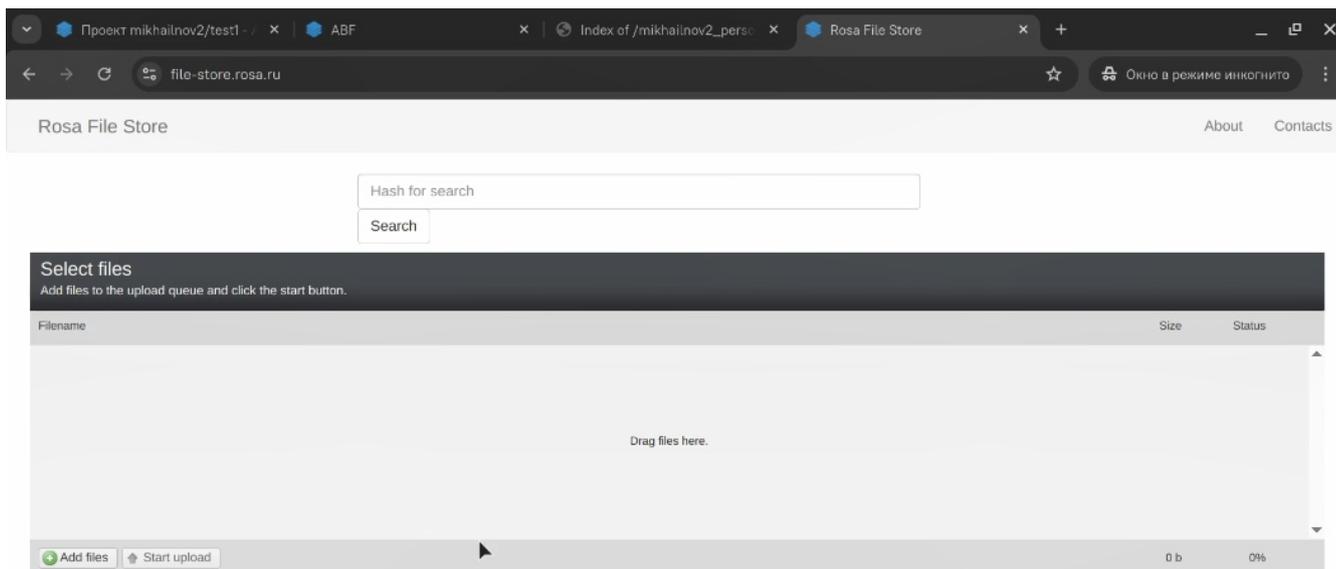


Рисунок 8: Веб-интерфейс службы File Store

Утилита `abf-put`

Команда `abf put` предназначена для загрузки изменений, внесённых в локальном репозитории, на сервер ABF. Она выполняет следующие действия:

- загружает бинарные файлы из текущей директории на файловый сервер ABF и добавляет их идентификаторы в файл `.abf.yml`;
- определяет файлы, указанные в `.abf.yml`, которые больше не нужны для сборки, и перемещает их в секцию `removed_sources` файла `.abf.yml`;
- применяет все изменения в проекте, сделанные локально, и отправляет их в удалённый Git-репозиторий.

Пример использования:

```
abf put -m "Обновление версии пакета"
```

Команда `abf put` поддерживает следующие опции:

- `-m, --message <сообщение>`: сообщение коммита, передаваемое команде `git commit`.
- `-s, --minimal-file-size <size>`: минимальный размер бинарного файла, при превышении которого файл загружается на файловый сервер. По умолчанию – `0`.
- `-n, --do-not-remove-files`: не удалять бинарные файлы из текущей директории после их загрузки.

– `-a, --upload-all`: загружать все бинарные файлы из текущей директории, независимо от их использования в `спес-файле`.

Для получения ссылок на исходники, указанные в `.спес-файле`, используется команда:

```
spectool -g *.спес
```

Эта команда подставляет значения макросов `%name` и `%version` и скачивает соответствующий архив.

Пример команды загрузки:

```
abf-put имя_файла.tar.gz
```

В результате в `.abf.yml` автоматически добавляется строка с именем файла и его хэш-суммой SHA1.

Для поиска и выгрузки исходно файла из службы File Store необходимо скопировать хэш-сумму файла SHA1 и вставить в соответствующую форму **Search** службы File Store. После нажатия на кнопку **Search** будет сформирована ссылка на скачивание файла.

3.2.5 Коммиты и отправка изменений на сервер

Добавьте файлы в индекс git:

```
git add
```

Зафиксируйте изменения:

```
git commit -m "Initial commit"
```

Отправьте изменения на сервер:

```
git push
```

После отправки изменений все добавленные в проект файлы будут отображаться в пользовательском веб-интерфейсе ABF в разделе работы с проектом.

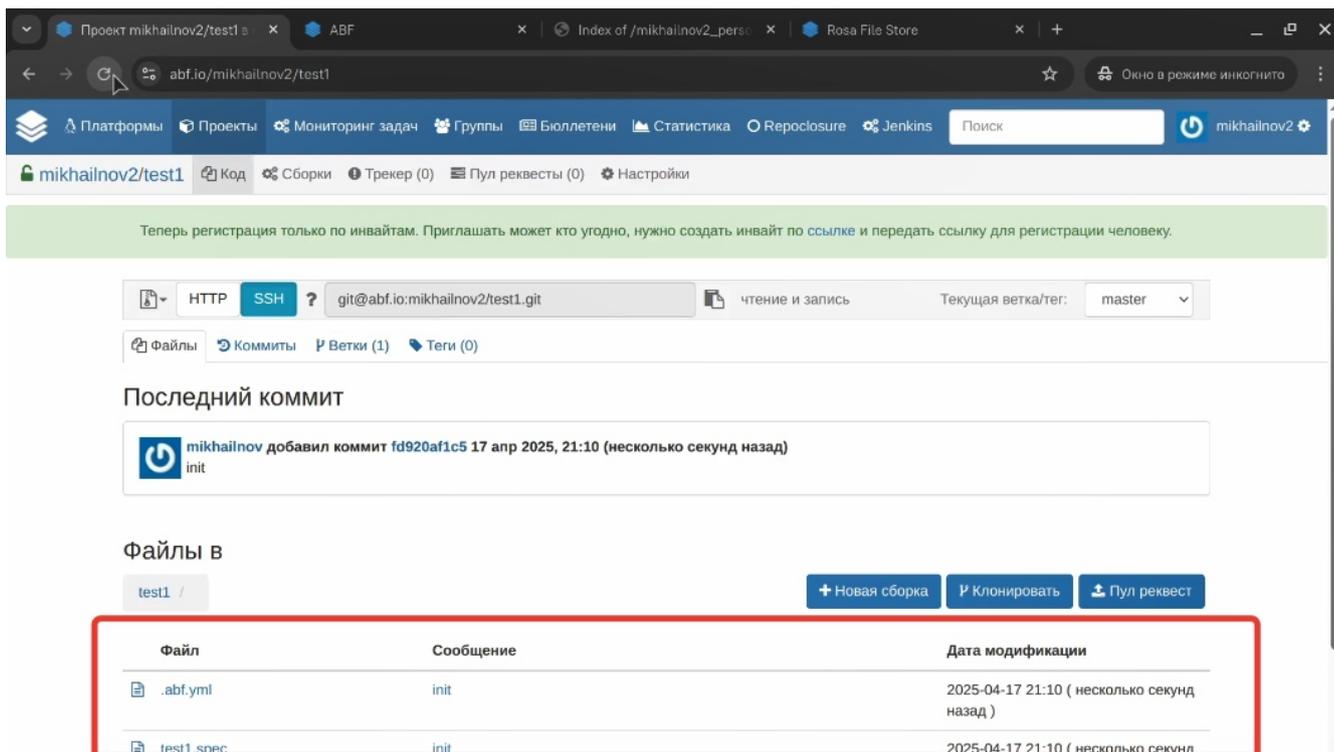


Рисунок 9: Список файлов, загруженных в пользовательский проект

3.2.6 Запуск сборки пакета

После подготовки файлов необходимо инициировать сборку:

- Нажмите кнопку **Новая сборка** в рабочем интерфейсе проекта ABF, после чего откроется меню сборки, где необходимо выбрать необходимые параметры;
- В выпадающем списке **Сохранить в репозиторий** выберете проект, в который будет осуществляться сборка пакетов;
- В поле **Версия** выберете ветку для сборки пакетов;
- В блоке **Архитектура** отметьте аппаратную архитектуру, под которую будет осуществляться сборка;

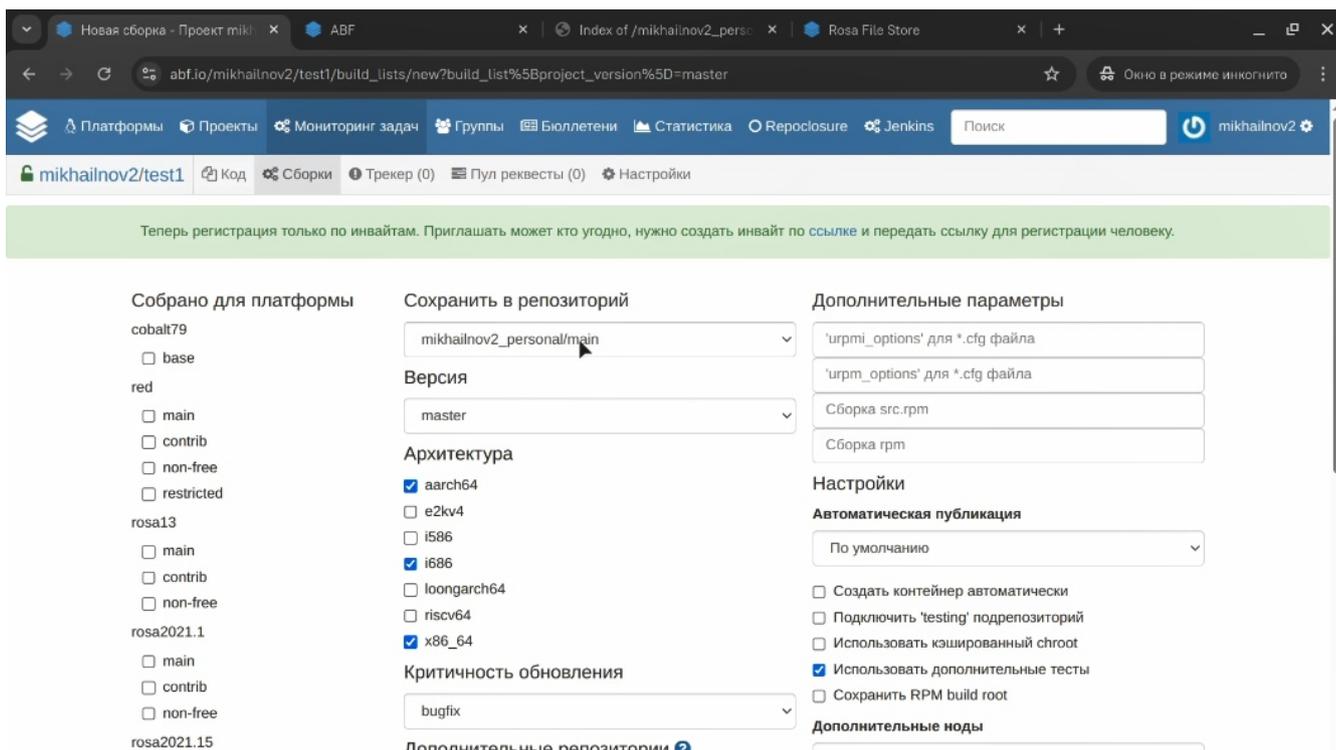


Рисунок 10: Меню сборки пакетов

- При первой сборке пакетов необходимо отключить какие-либо дополнительные репозитории в соответствующем блоке;
- В блоке **Собрано для платформы** выбираются платформа под которые будет осуществляться сборка;
- При необходимости в блоке **Настройки** установите флажок **Создать контейнер автоматически** – это создаст мини-репозиторий только с результатом данной сборки. Также данная опция будет дополнительно предложена после окончания сборки пакетов.

3.2.7 Проверка тестов после сборки

Для того, чтобы провести дополнительное тестирование собранных пакетов в блоке **Настройки** при сборке пакетов установите флажок **Использовать дополнительные тесты**.

Дополнительные тесты выполняются путем установки всех собранных пакетов, после чего система проверяет успешно ли прошла установка. Если какие-либо зависимости отсутствуют, тесты завершатся с ошибкой. В этом случае появится кнопка **Перезапустить тесты**.

Для инициации процесса сборки пакетов нажмите на кнопку **Начать сборку**.

3.2.8 Просмотр сборочных листов

Обновите рабочей интерфейс ABF, после чего в рабочем поле будут отображены все сборочные листы проекта.

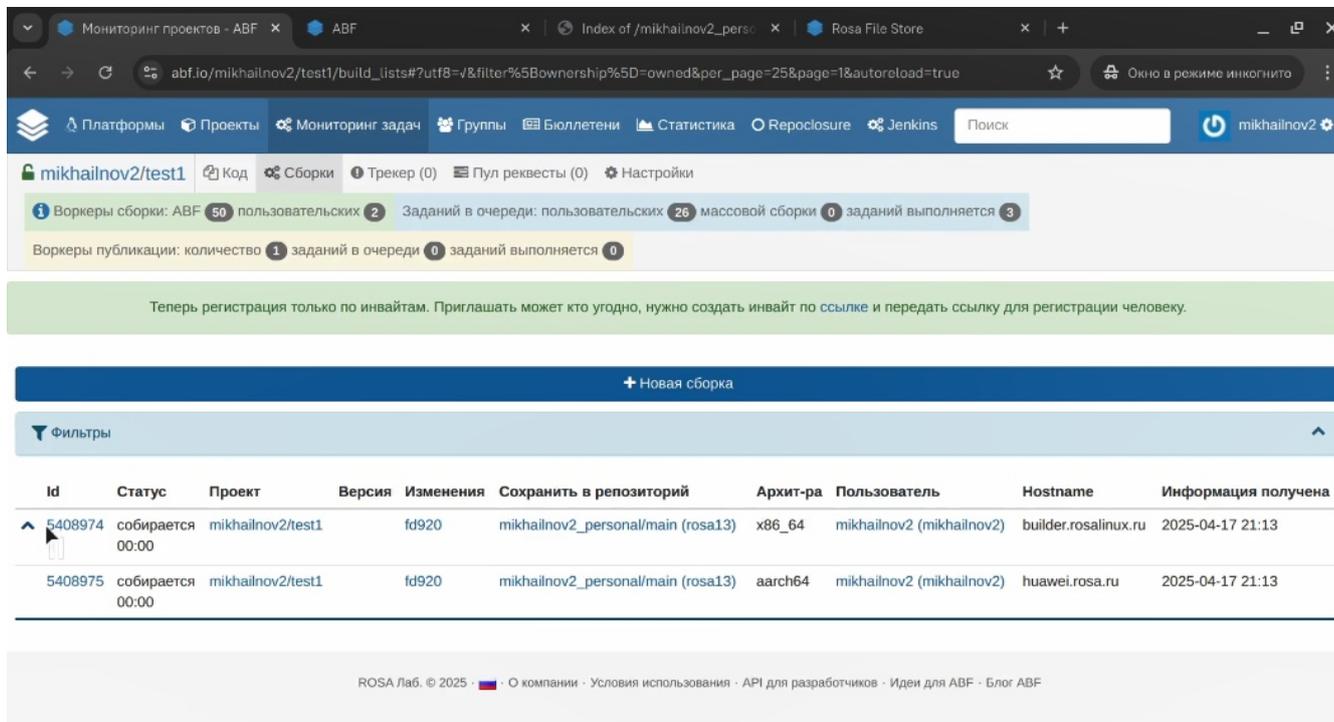


Рисунок 11: Список сборочных листов проекта

Перейдите на сборочную страницу по нажатию на id пакета. Адрес страницы сборочного листа имеет формат: **abf.io/buildlists/номер сборочного листа**

На открывшейся странице отображаются основные данные и другая информация по собираемому пакету, указывается ожидаемое время сборки и текущее время сборки.

На момент, пока сборка еще не завершена, на сборочной странице можно **отменить сборку** или **пересоздать сборку** нажатием на соответствующую кнопку:

— Кнопка **Пересоздать сборку** не отменяет текущую сборку, она просто открывает страницу создания новой сборки, в которой настройки выставлены аналогично сборке, которую вы пересоздали.

— Кнопка **Отменить сборку** досрочно завершает сборку но логи сборки все равно появляются в сборочном листе.

Также на данной странице в режиме реального времени можно просматривать лог сборки.

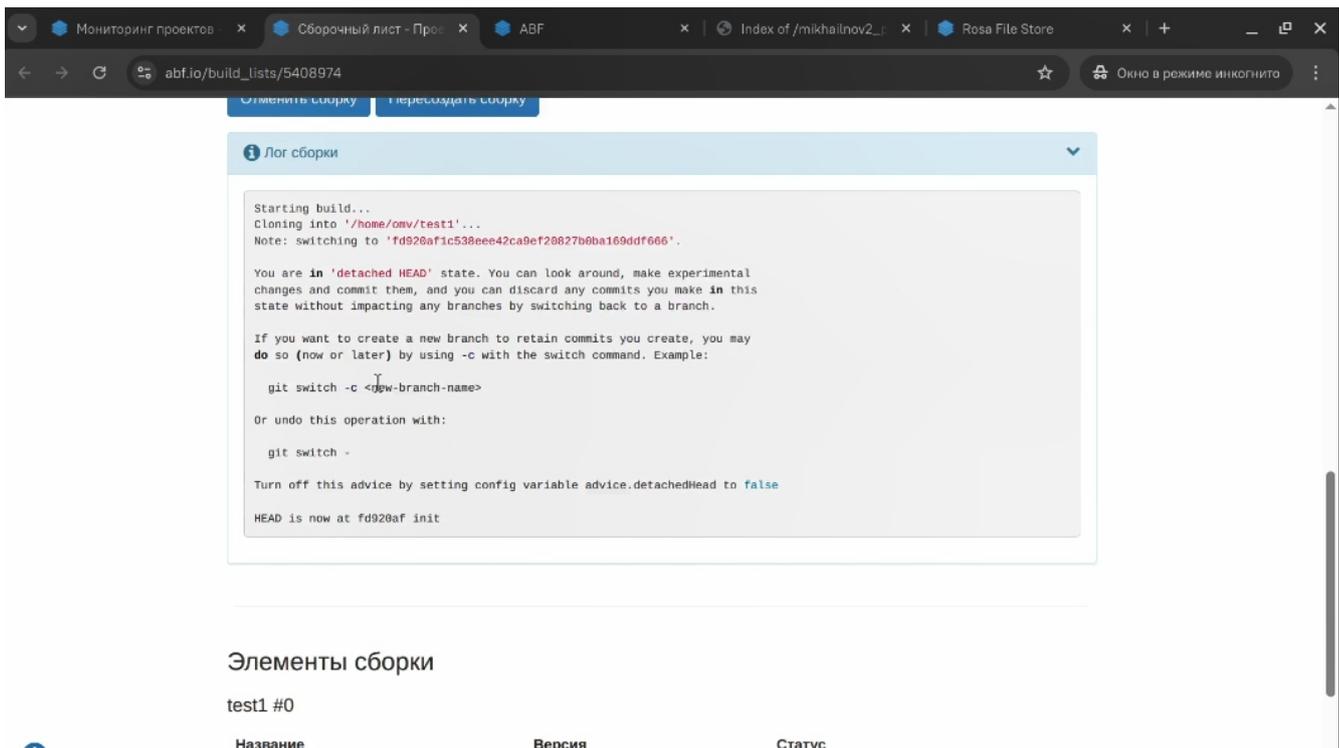


Рисунок 12: Просмотр лога сборки пакетов в реальном времени

По завершению сборки пакета, полный лог сборки будет доступен в виде текстового файла.

3.2.9 Завершение сборки

Все собираемые пакеты имеют общую очередь, где публикация происходит последовательно.

Когда все пакеты успешно собраны и опубликованы в сборочном листе, в блоке **Пакеты** отображается список всех собранных пакетов, в блоке **Логи** — все логи сборки.

Пакеты

Полное имя	Название	Размер	Эпоха	Версия	Релиз
dos2unix-7.5.2-3-rosa13.x86_64.rpm	dos2unix	428.0KiB		7.5.2	3
dos2unix-7.5.2-3.src.rpm	dos2unix	976.65KiB		7.5.2	3
dos2unix-debuginfo-7.5.2-3-rosa13.x86_64.rpm	dos2unix-debuginfo	70.05KiB		7.5.2	3
dos2unix-debugsource-7.5.2-3-rosa13.x86_64.rpm	dos2unix-debugsource	34.02KiB		7.5.2	3

Логи

Имя файла	SHA1	Размер (МБ)	Создан
build.log.gz	32b5d4773b08497baaf516274171d98cf346a900	0.00	2025-04-17 18:14:45 UTC
installed_pkgs.log.gz	860cf50d9efb38f6100546f0bc27459badbc9151	0.01	2025-04-17 18:14:45 UTC
root.log.gz	bde56f1f35426aa3d31a7e6a00b83d7c260115cb	0.02	2025-04-17 18:14:45 UTC
script_output.log	e13266cd13d6f6a6dceebc22fb572547de3109f8	0.11	2025-04-17 18:14:45 UTC
state.log.gz	a963f3dfaed12075104adcb2983b3833999004a3	0.00	2025-04-17 18:14:45 UTC
test.04-17-2025-18-13-16.log	76b8ba2653d27e48b85ca8a7f743539a69001bb8	0.01	2025-04-17 18:14:45 UTC

Рисунок 13: Собранные пакеты и логи по текущему проекту пользователя

Файл `script_output.log` является основным логом сборки.

Все собранные пакеты будут опубликованы по адресу, который указывался ранее в разделе 2.4 : **abf-downloads.rosa.ru/(или иной адрес данной копии abf)/название_платформы**. После успешной сборки данный репозиторий может быть подключен как репозиторий к менеджеру пакетов DNF.

4 РАБОТА С КОНСОЛЬНЫМ КЛИЕНТОМ ABF

Система ABF также предоставляет консольный клиент `abf`, предназначенный для выполнения ряда задач через командную строку без необходимости использования веб-интерфейса. Данный инструмент позволяет разрабатывать, собирать и управлять проектами в системе ABF в автоматизированном или скриптовом режиме.

Консольный клиент доступен в репозиториях дистрибутива и устанавливается через систему управления пакетами:

```
sudo dnf install abf-console-client
```

После установки необходимо выполнить базовую настройку.

4.1 Аутентификация и настройка клиента

Перед началом работы необходимо пройти аутентификацию с использованием учетной записи ABF. Выполните следующую команду:

```
abf login
```

При этом потребуется ввести логин и пароль. После успешной авторизации клиент сохранит конфигурационные данные в пользовательском каталоге.

Проверить текущую конфигурацию можно с помощью команды:

```
abf whoami
```

4.2 Получение информации о проекте

Консольный клиент позволяет получать различную информацию о проектах:

– Просмотр списка собственных проектов:

```
abf projects list
```

– Просмотр информации о конкретном проекте:

```
abf project show <имя_проекта>
```

4.3 Управление задачами сборки

Запуск сборки пакета можно выполнить из локального каталога проекта:

```
abf build --arch x86_64 --platform <имя_платформы>
```

Параметры сборки могут быть уточнены в командной строке, включая архитектуру, тип сборки и используемые репозитории. Также возможно указание тестовой сборки, без публикации артефактов в основной репозиторий.

4.4 Загрузка исходников

Для загрузки новых исходных файлов (например, .tar.gz-архивов), используйте команду:

```
abf put <путь_к_файлу>
```

Загруженные файлы автоматически размещаются в службе File Store и становятся доступными в проекте.

4.5 Управление релизами и публикация

Публикация пакетов и управление релизами также может быть выполнена через консольный клиент. Для этого используется команда:

```
abf publish <опции>
```

Подробности публикации (включая размещение в определённом репозитории, выбор архитектур и др.) можно задать через параметры CLI.

4.6 Дополнительные возможности

Консольный клиент предоставляет дополнительные возможности при работе с утилитой. Список основных команд консольного клиента ABF представлен в таблице 1.

Таблица 1 – Основные команды консольного клиента ABF

Команда	Назначение
abf login	Аутентификация пользователя в системе ABF.
abf whoami	Отображение текущего авторизованного пользователя.
abf projects list	Просмотр списка проектов пользователя.
abf project show <имя_проекта>	Просмотр информации о конкретном проекте.
abf build --arch <архитектура> --platform	Запуск сборки пакета с указанием архитектуры и платформы.

<платформа>	
abf builds list	Список всех сборочных заданий пользователя.
abf build cancel <ID_задания>	Отмена задания на сборку.
abf build log <ID_задания>	Получение логов по сборочному заданию.
abf put <путь_к_файлу>	Загрузка сторонних файлов в File Store.
abf publish <опции>	Публикация собранного пакета в репозиторий.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ПО	– Программное обеспечение
ABF	– Automated Build Farm (автоматизированная сборочная ферма)
API	– Application Programming Interface (программный интерфейс приложения)
DNF	– DNF (Dandified Yum) – менеджер пакетов для RPM-дистрибутивов Linux
HTTPS	– HyperText Transfer Protocol Secure (защищённый протокол передачи гипертекста)
ISO	– Файловый образ диска в формате ISO
RPM	– RPM Package Manager (система управления пакетами RPM)
VCS	– Version Control System (Система контроля версий)

