



АО "ИТЦ ИТ РОСА"

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

**Руководство пользователя**

РСЮК.10902-02 34 01

Листов 23

2026

## АННОТАЦИЯ

Настоящий документ является руководством по эксплуатации программного обеспечения "Система автоматизации сборки и поддержки пакетов, репозиторий и дистрибутивов продуктов ABF" (индекс – РСЮК.10902-02) (далее – ABF, Система).

Автоматизированная система сборки ABF, разработанная компанией АО "НТЦ ИТ РОСА", включена в Единый реестр российских программ для ЭВМ и баз данных (реестровая запись №28826 от 28.07.2025).

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 настоящем руководстве 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.

Для разработки документа использованы ссылки на следующие стандарты:

- ГОСТ Р 2.105-2019 "Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам";
- ГОСТ 2.601 "Единая система программной документации. Виды программных документов";
- ГОСТ 19.101-77 "Единая система программной документации. Виды программ и программных документов";
- ГОСТ 19.105-78 "Единая система программной документации. Общие требования к программным документам";
- ГОСТ 19.503-79 "Единая система программной документации. Руководство системного программиста".

Настоящий документ подготовлен в соответствии с технологической инструкцией "РОСА. Регламент формирования документации к программным продуктам" РСЮК.11001-02 90 01.

## СОДЕРЖАНИЕ

1	Функциональные возможности.....	4
2	Начало работы.....	6
2.1	Авторизация через веб-интерфейс.....	6
2.2	Главное меню и пользовательский интерфейс.....	6
2.3	Просмотр пользовательских проектов.....	7
2.4	Публикация собранных пакетов.....	9
3	Работа в веб-интерфейсе.....	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.....	12
3.2.3	Подготовка файлов для сборки.....	12
3.2.4	Добавление сторонних источников в репозиторий через службу File Store.....	13
3.2.5	Коммиты и отправка изменений на сервер.....	15
3.2.6	Запуск сборки пакета.....	15
3.2.7	Проверка тестов после сборки.....	17
3.2.8	Просмотр сборочных листов.....	17
3.2.9	Завершение сборки.....	19
4	Работа с консольным клиентом.....	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
	Перечень сокращений.....	23

## 1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Система автоматизированной сборки ABF предоставляет широкий набор инструментов для различных категорий пользователей — разработчиков, мейнтейнеров, администраторов и участников платформ. В этом разделе 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 частных 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 Системы;
- 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;
- 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

### 2.1 Авторизация через веб-интерфейс

Для начала работы пользователь должен авторизоваться в Системе. Для этого необходимо:

- а) открыть веб-браузер и перейти по ссылке <https://abf.io/> (рисунок 1);
- б) ввести имя пользователя и пароль;
- в) нажать кнопку **Войти**.

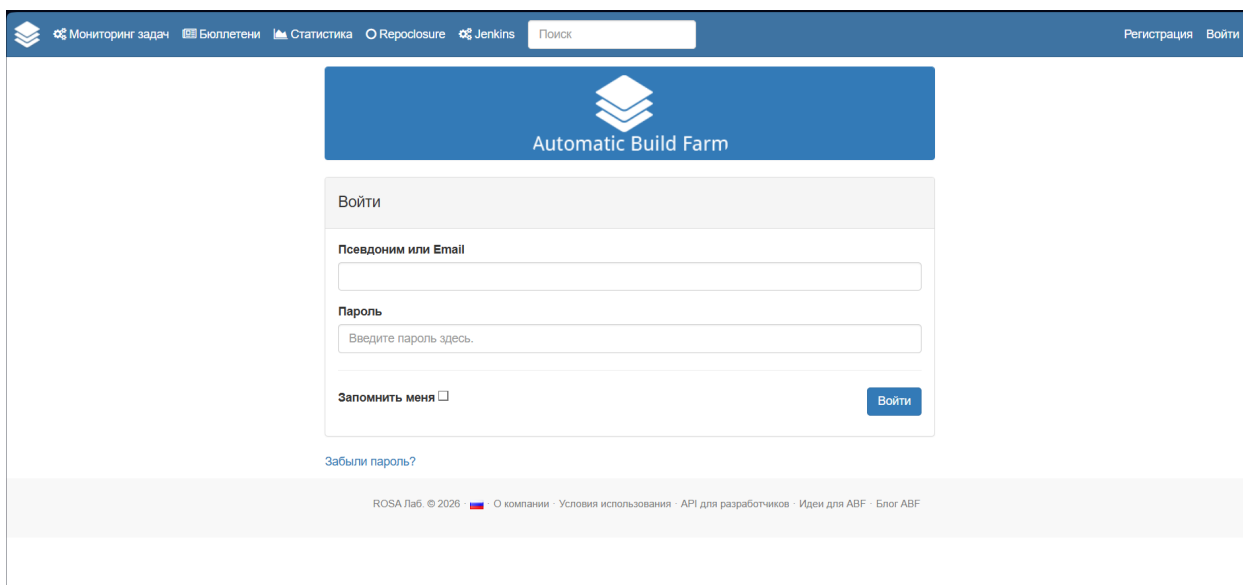


Рисунок 1 – Форма авторизации пользователей в веб-интерфейсе ABF

### 2.2 Главное меню и пользовательский интерфейс

После успешной авторизации пользователь попадает в главное окно Системы (рисунок 2). 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.

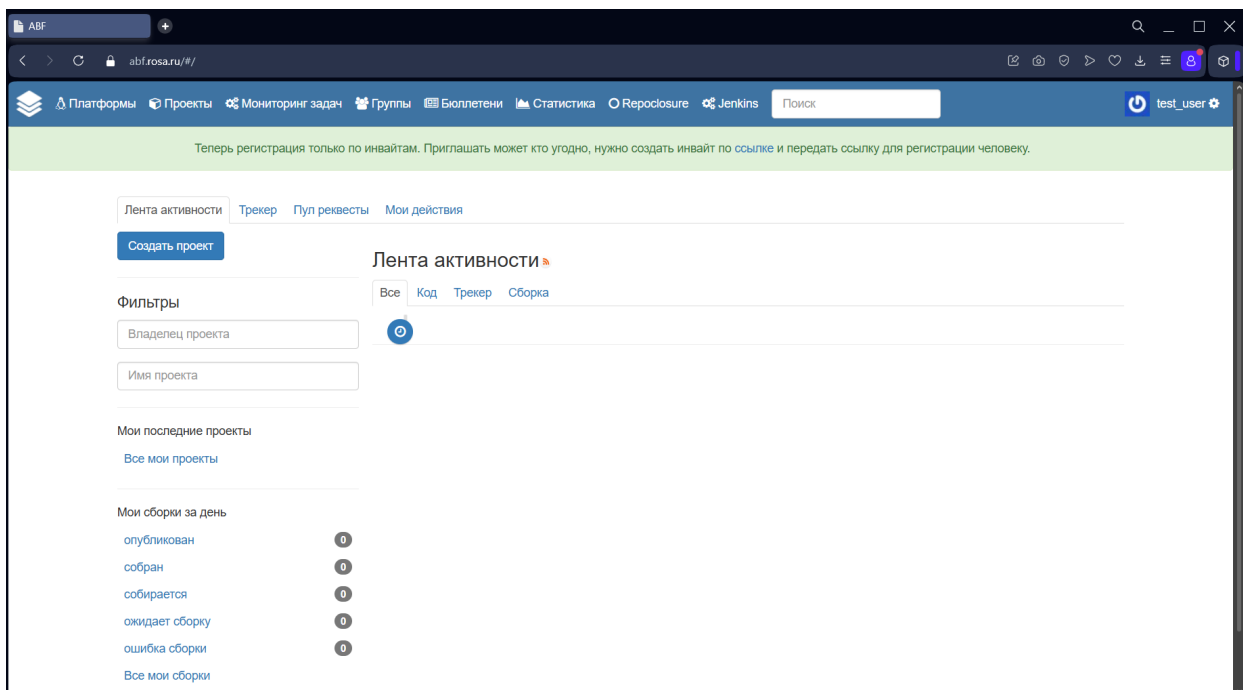


Рисунок 2 – Главное окно Системы после входа, основное меню и активные разделы

## 2.3 Просмотр пользовательских проектов

В разделе "Платформы" (на верхней панели меню) представлен список пользовательских репозиторий.

Для просмотра списка всех пользовательских репозиторий нужно:

- открыть раздел "Платформы" в верхнем навигационном меню;
- на экране отобразится список всех доступных проектов с указанием типа дистрибутива (рисунок 3);
- для перехода в конкретный проект нажать на его имя.

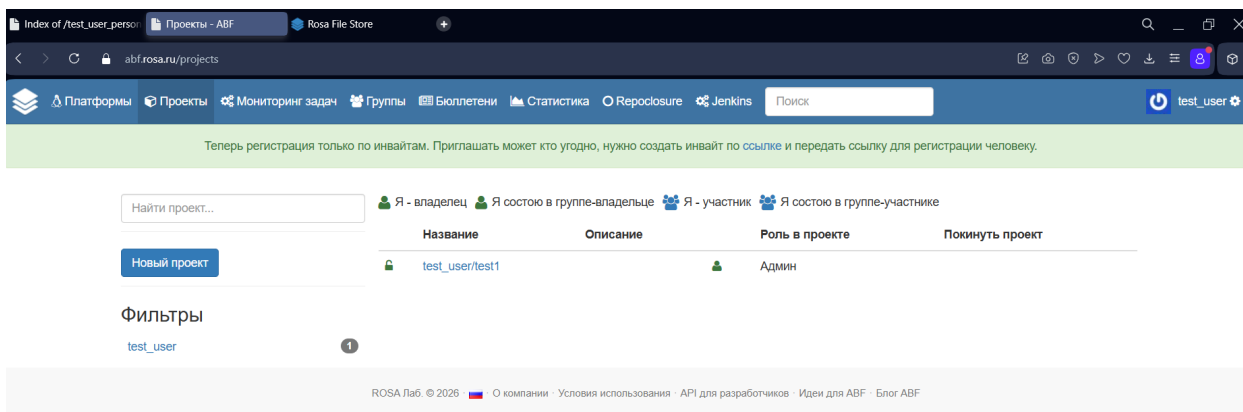


Рисунок 3 – Просмотр проектов пользователя

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 можно перейти в конкретную платформу по нажатию на имя платформы. Для каждого репозитория во вкладке "О платформе" отображаются ветки, история изменений и теги (рисунок 4).

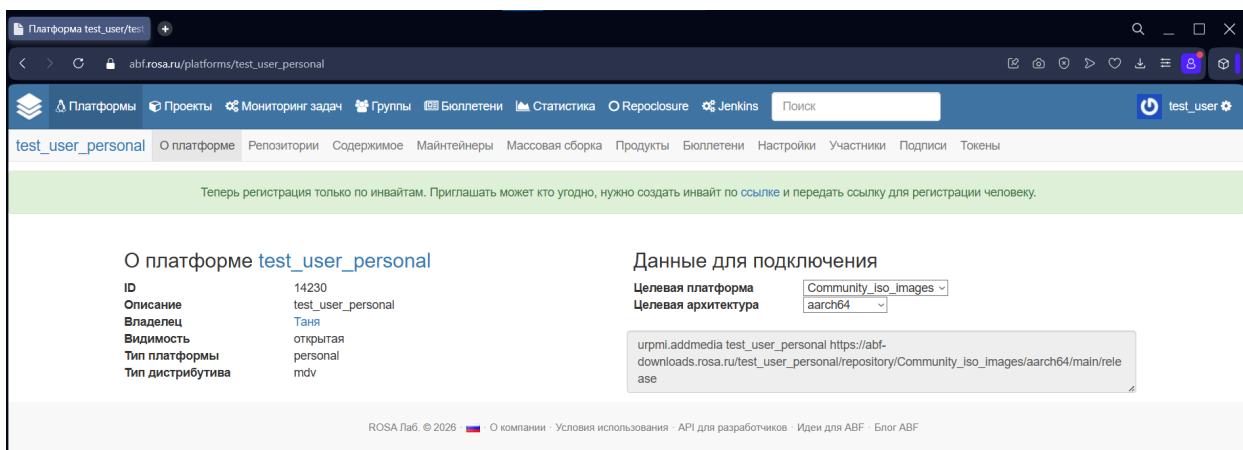


Рисунок 4 – Просмотр подробной информации о платформе

Для просмотра всех пакетов, которые будут опубликованы на соответствующих ресурсах, следует перейти на вкладку "Содержимое" (рисунок 5).

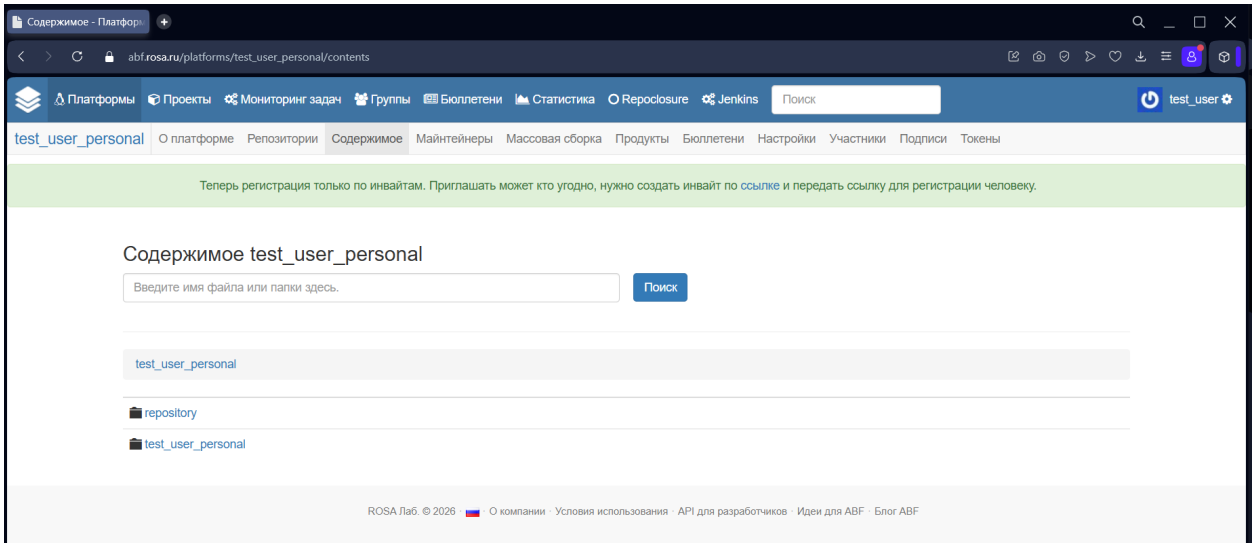


Рисунок 5 – Просмотр содержимого текущего репозитория

## 2.4 Публикация собранных пакетов

Адрес ресурса, на который происходит публикация репозитория, то есть тех пакетов, которые уже собраны, имеет следующий формат (рисунок 6):

abf-downloads.rosa.ru/(или иной адрес данной копии abf) /<название\_платформы>

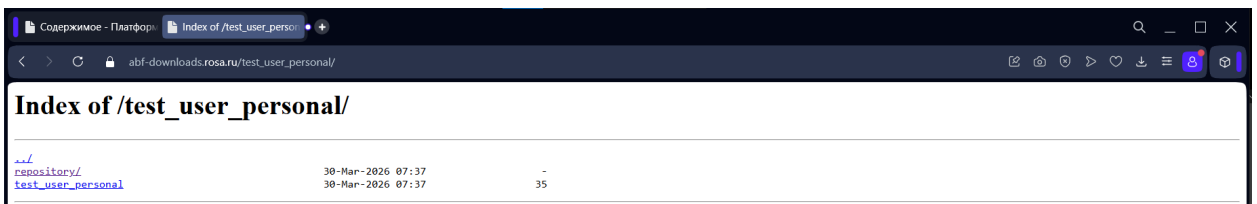


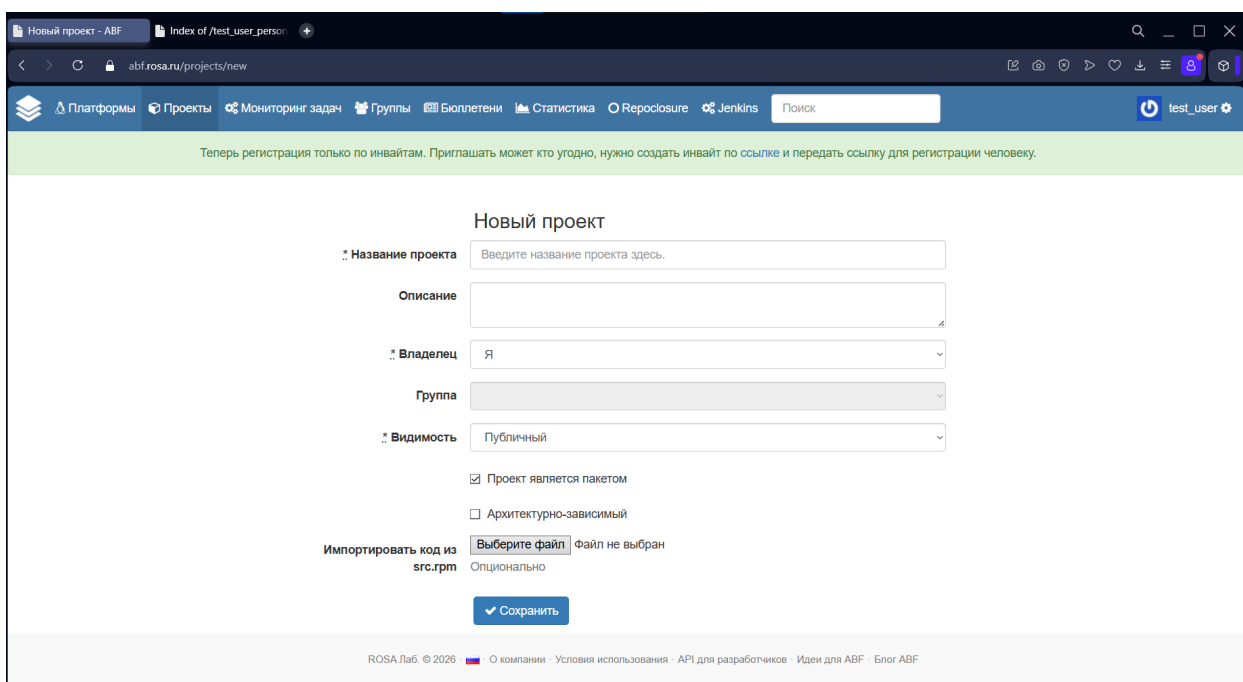
Рисунок 6 – Публикация собранных пакетов

## 3 РАБОТА В ВЕБ-ИНТЕРФЕЙСЕ

### 3.1 Запуск новой сборки

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

- а) перейти на вкладку "Проекты → Новый проект" (рисунок 7);
- б) заполнить форму создания сборки (обязательные для заполнения поля отмечены символом звездочки "\*");
- в) нажать на кнопку **Сохранить**, чтобы продолжить процесс создания проекта.



Новый проект - ABF | Index of /test\_user\_person

abf.rosa.ru/projects/new

Платформы | Проекты | Мониторинг задач | Группы | Биолетени | Статистика | Repoclosure | Jenkins | Поиск

test\_user

Теперь регистрация только по инвайтам. Приглашать может кто угодно, нужно создать инвайт по ссылке и передать ссылку для регистрации человеку.

### Новый проект

\* Название проекта

Описание

\* Владелец

Группа

\* Видимость

Проект является пакетом

Архитектурно-зависимый

Импортировать код из  файл не выбран

src.gpm

ROSA Лаб. © 2026 | О компании | Условия использования | API для разработчиков | Идеи для ABF | Блог ABF

Рисунок 7 – Создание нового проекта

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

- .спес-файл (описание пакета);

– файл `.abf.yml` (описание сторонних источников в репозитории, если они есть).

## 3.2 Этапы сборки проекта

В этом разделе рассматривается пример создания проекта с именем `test1`.

### 3.2.1 Настройка доступа к репозиторию

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

Для настройки доступа через SSH-протокол необходимо выполнить следующие действия:

а) создать ключевую пару с помощью команды:

```
ssh-keygen
```

б) добавить открытый ключ в разделе "Настройки → SSH-ключи" в меню Системы (рисунок 8);

в) после добавления ключа работать с репозиторием через команду:

```
git clone ssh://<адрес_репозитория>
```

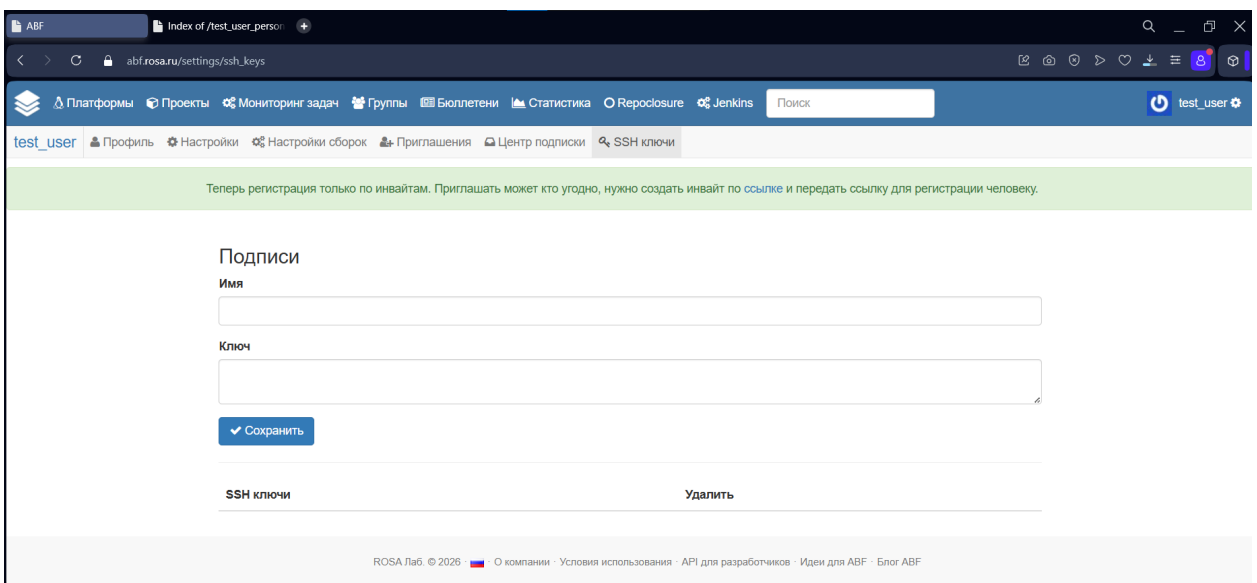


Рисунок 8 – Добавление открытого ключа

### 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 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 локального клонирования репозитория проекта с использованием git-клиента нужно использовать команду:

```
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 можно запускать сборку с использованием актуального состояния проекта.

**Важно** – Клонирование не требуется только в случае повторной сборки проекта без внесения изменений, однако при любой модификации файлов оно 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

Для подготовки нужно создать в локальном репозитории файл с расширением .spec (например, test1.spec).

**Примечание** – Название файла, как правило, совпадает с названием проекта, но это не обязательное требование.

Дополнительно требуется создать 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`. **Следует обратить внимание**, что все указанные исходные файлы при сборке должны быть заранее загружены в службу File Store, т. к. 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 ABF и 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) (рисунок 9);
- с помощью командной утилиты `abf-put`.

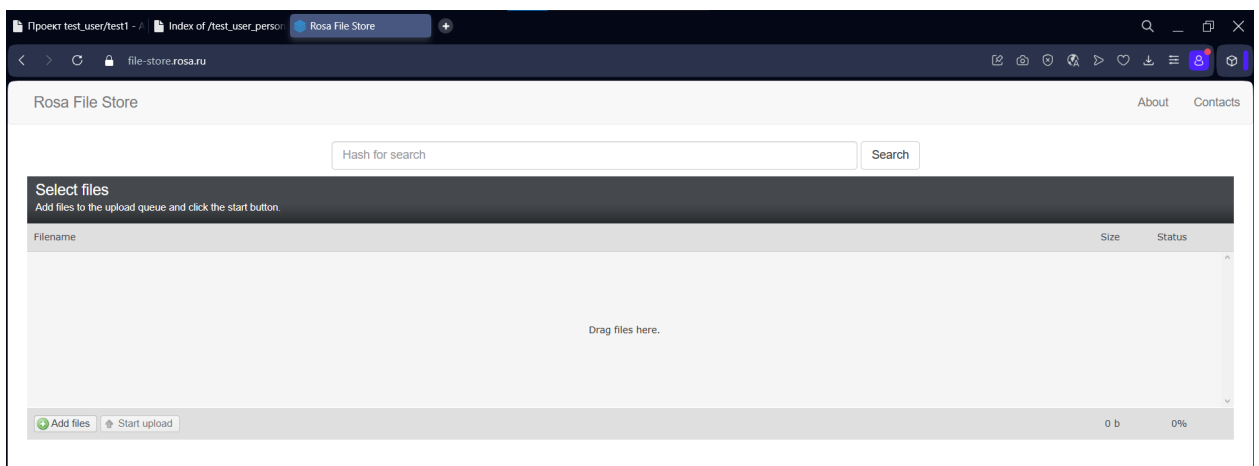


Рисунок 9 – Веб-интерфейс службы File Store

### 3.2.4.1 Утилита `abf-put`

Команда `abf put` предназначена для загрузки изменений, внесённых в локальном репозитории, на сервер ABF. С её помощью можно выполнить следующие действия:

- загрузить бинарные файлы из текущей директории на файловый сервер ABF и добавить их идентификаторы в файл `.abf.yml`;
- определить ненужные для сборки файлы, указанные в `.abf.yml`, и переместить их в секцию `removed_sources` файла `.abf.yml`;
- применить все изменения в проекте, сделанные локально, и отправить их в удалённый git-репозиторий.

Пример использования команды:

```
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 `.spec`-файле.

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

```
spectool -g *.spec
```

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

В результате выполнения следующей команды в `.abf.yml` автоматически добавляется строка с именем файла и его хеш-суммой SHA1:

```
abf-put имя_файла.tar.gz
```

Для поиска и выгрузки исходного файла из службы File Store необходимо скопировать хеш-сумму файла SHA1 и вставить в соответствующее поле "Hash

to search" службы File Store. После нажатия на кнопку **Search** будет сформирована ссылка на скачивание файла.

### 3.2.5 Коммиты и отправка изменений на сервер

Для отправки изменений на сервер нужно:

а) добавить файлы в индекс git:

```
git add
```

б) зафиксировать изменения:

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

в) отправить изменения на сервер:

```
git push
```

После отправки изменений все добавленные в проект файлы будут отображаться в пользовательском веб-интерфейсе Системы в разделе работы с проектом (рисунок 10).

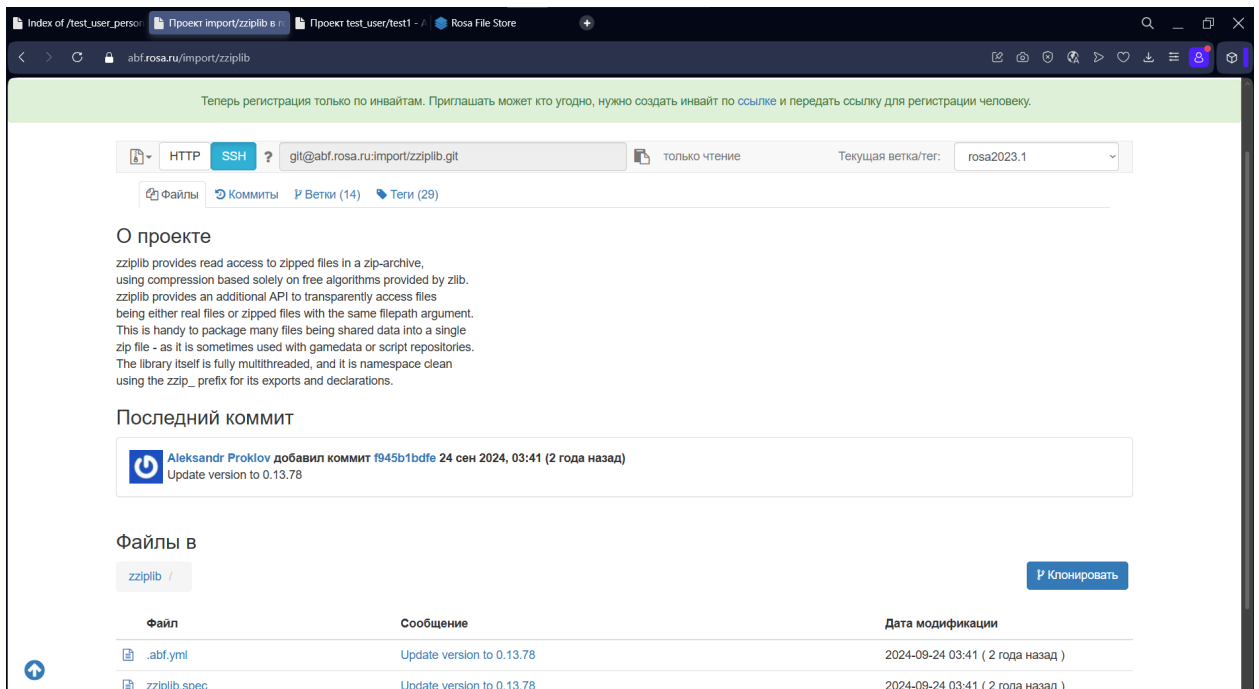


Рисунок 10 – Список файлов, загруженных в пользовательский проект

### 3.2.6 Запуск сборки пакета

После подготовки файлов необходимо инициировать сборку. Для этого нужно:

а) нажать кнопку **Новая сборка** в рабочем интерфейсе проекта ABF, после чего откроется меню сборки, где нужно выбрать необходимые параметры (рисунок 11);

б) в выпадающем списке "Сохранить в репозиторий" выбрать проект, в который будет осуществляться сборка пакетов;

в) в поле "Версия" выбрать ветку для сборки пакетов;

г) в секции "Архитектура" отметить аппаратную архитектуру, под которую будет осуществляться сборка;

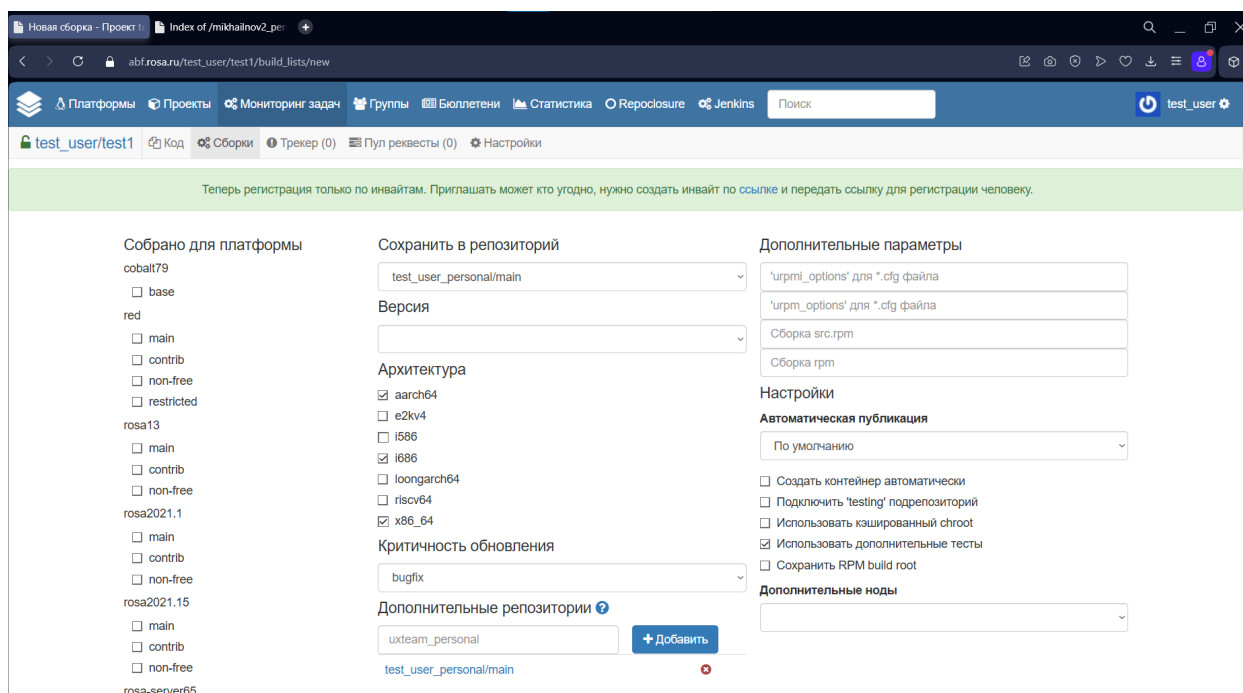


Рисунок 11 – Меню сборки пакетов

**Важно** – при первой сборке пакетов необходимо отключить какие-либо дополнительные репозитории в соответствующем блоке;

д) в секции "Собрано для платформы" выбирать платформы, под которые будет осуществляться сборка.

**Примечание** – При необходимости в секции "Настройки" можно установить 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 дополнительного тестирования собранных пакетов в секции "Настройки" при сборке пакетов нужно установить 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 следует нажать на кнопку **Начать сборку**.

### 3.2.8 Просмотр сборочных листов

Для того чтобы в рабочем поле отобразились все сборочные листы проекта, нужно обновить рабочий интерфейс ABF (рисунок 12).

The screenshot shows the ABF interface for the 'import/zziplib' project. The main content area displays a table of build sheets. The table has the following columns: Id, Статус, Проект, Версия, Изменения, Сохранить в репозиторий, Архит-ра, Пользователь, Hostname, and Обновлено(статус/публикация/и т.д.).

Id	Статус	Проект	Версия	Изменения	Сохранить в репозиторий	Архит-ра	Пользователь	Hostname	Обновлено(статус/публикация/и т.д.)
5348937	опубликован	import/zziplib	0.13.78-1	f945b...f945b	rosa13/main	aarch64	fedya (fedya)	huawei.rosa.ru	2025-02-19 18:32
5071139	опубликован	import/zziplib	0.13.77-1	f6081...2f1a7	rosa13/main	e2kv4	fedya (fedya)	rootfs2	2024-07-13 04:00

Рисунок 12 – Список сборочных листов проекта

Для перехода на сборочную страницу нужно нажать 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, указываются ожидаемое и текущее время сборки.

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

– **Пересоздать сборку** – не отменять текущую сборку, а открыть страницу создания новой сборки, в которой настройки выставлены аналогично пересоздаваемой сборке;

– **Отменить сборку** – досрочно завершить сборку с сохранением логов сборки в сборочном листе.

Также на данной странице в режиме реального времени можно просматривать лог сборки (рисунок 13).

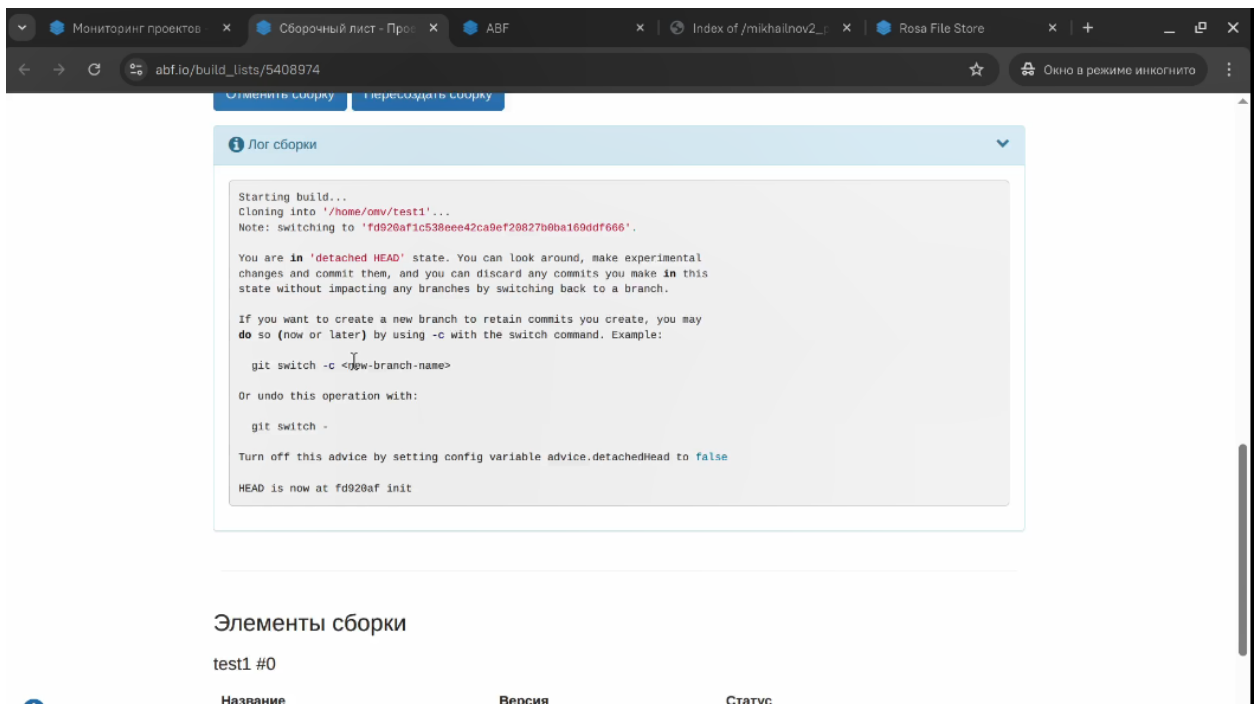


Рисунок 13 – Просмотр лога сборки пакетов в реальном времени

После завершения 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 секции "Пакеты" отображается список всех собранных пакетов, в секции "Логи" – все логи сборки (рисунок 14).

Полное имя	Название	Размер	Эпоха	Версия	Релиз
<a href="#">dos2unix-7.5.2-3-rosa13.x86_64.rpm</a>	dos2unix	428.0KiB		7.5.2	3
<a href="#">dos2unix-7.5.2-3.src.rpm</a>	dos2unix	976.65KiB		7.5.2	3
<a href="#">dos2unix-debuginfo-7.5.2-3-rosa13.x86_64.rpm</a>	dos2unix-debuginfo	70.05KiB		7.5.2	3
<a href="#">dos2unix-debugsource-7.5.2-3-rosa13.x86_64.rpm</a>	dos2unix-debugsource	34.02KiB		7.5.2	3

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

Рисунок 14 – Собранные пакеты и логи по текущему проекту

Файл `script_output.log` является основным логом сборки.

Все собранные пакеты будут опубликованы по адресу, который указывался ранее в п. 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

Система 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 в автоматизированном или скриптовом режиме.

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

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

**Важно** – После установки необходимо выполнить базовую настройку.

### 4.1 Аутентификация и настройка клиента

Перед началом работы необходимо пройти аутентификацию с использованием учетной записи ABF, для чего нужно:

а) выполнить следующую команду:

```
abf login
```

б) ввести логин и пароль, при этом после успешной авторизации клиент сохранит конфигурационные данные в пользовательском каталоге;

в) проверить текущую конфигурацию с помощью команды:

```
abf whoami
```

### 4.2 Получение информации о проекте

С помощью консольного клиента можно:

– просмотреть список собственных проектов:

```
abf projects list
```

– просмотреть информацию 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) следует использовать 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 `abf`. Список основных команд консольного клиента ABF представлен в таблице 1.

Таблица 1 – Основные команды консольного клиента ABF

Команда	Назначение
<code>abf login</code>	Аутентификация пользователя в системе ABF

Команда	Назначение
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 <платформа>	Запуск сборки пакета с указанием архитектуры и платформы
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 система сборки)
API	– Application Programming Interface (программный интерфейс приложения)
CLI	– Интерфейс командной строки
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)
SSH	– Secure shell – защищенная оболочка
VCS	– Version Control System (Система контроля версий)