

---

# Стартовая страница документации

ROBIN RPA Team

мар. 03, 2022

---

## Содержание:

---

1	ROBIN Platform. Описание применения	2
2	ROBIN Studio. Руководство пользователя	14
3	ROBIN Studio. Описание действий	205
4	ROBIN Robot. Руководство пользователя	206
5	ROBIN Management Console. Руководство пользователя	222
6	ROBIN Orchestrator. Руководство системного администратора	253
7	Инструкция по созданию пользовательских действий на C#	272
8	Скрипты по автоматическому развертыванию RMS. Руководство администратора	295
9	Сервер авторизации FreeIPA. Руководство администратора	308
10	Серверные компоненты ROBIN Platform	355
11	Приложения	372
12	Глоссарий	376
13	Release Notes	378
14	Техническая поддержка ROBIN	405



---

## ROBIN Platform. Описание применения

---

### 1.1 Аннотация

Настоящий документ является описанием применения комплекса программных средств Платформы ROBIN (далее – Платформа). Документ содержит общее описание задач и способов их решения, условий применения, входных и выходных данных Платформы.

### 1.2 Назначение

#### 1.2.1 Назначение изделия

Платформа ROBIN предназначена для поддержания жизненного цикла программных роботов – их разработки, отладки, хранения, публикации и эксплуатации.

#### 1.2.2 Состав изделия

Автоматизированное сопровождение жизненного цикла программных роботов достигается за счет объединения на одной платформе комплекса программных средств:

- «ROBIN Studio» – программное обеспечение, предназначенное для разработки, отладки, хранения и публикации программных роботов;
- «ROBIN Robot» – программное обеспечение, предназначенное для локального запуска и контроля – эксплуатации программных роботов, созданных с помощью ПО «ROBIN Studio»;
- «ROBIN Management Console» – программное обеспечение, предназначенное для контроля функционирования на удаленных хостах программных роботов, является составной частью AC «ROBIN Orchestrator»;
- AC «ROBIN Orchestrator» – компонент мониторинга и управления программными роботами, комплекс программных средств серверной части Платформы ROBIN, предназначенных для обеспечения взаимодействия всех компонентов Платформы ROBIN на всех этапах жизненного цикла программных роботов – создания, хранения, изменения, отладки, эксплуатации.



### 1.2.3 Возможности изделия

Платформа ROBIN позволяет автоматизировать процессы выполнения следующих основных задач:

- Автоматизация процесса создания программных роботов.
- Управление эксплуатацией программных роботов.

### 1.2.4 Основные характеристики изделия

#### Простота и удобство

Платформа ROBIN – No-Code инструмент. Бизнес-пользователю системы не нужны навыки программирования.

Функционал платформы ROBIN строго разделяет роли программиста и аналитика. Программист занимается разработкой действий, а аналитик выстраивает процесс в ПО «ROBIN Studio» (далее-Студия). Простота и удобство интерфейса позволяют пользователю начать настраивать робота после прохождения базового курса обучения. Главное – понимать бизнес-процесс, а удобный интерфейс, встроенные в Студию подсказки и документация помогут быстро адаптироваться и начать избавляться от рутины с помощью программных роботов.

#### Программные роботы

Программный робот - программное обеспечение, имитирующее действия человека в информационных системах через интерфейс. Для системы действия робота ничем не отличаются от действий человека. Ключевой функционал платформы ROBIN предназначен для бизнес-пользователей, не имеющих технической подготовки, которые смогут интуитивно понятным способом быстро настроить программного робота также, как пользователь может обучить своего коллегу.

Студия позволяет разработчику создавать программные роботы без написания программного кода. Проекты робота создаются в визуальном редакторе в формате блок-схем. Элементарным функциональным блоком такой блок-схемы является готовое программное действие, входящее в состав библиотеки стандартных действий Студии. Линии, соединяющие функциональные блоки схемы, указывают последовательность выполнения действий. Для построения гибких алгоритмов роботов могут использоваться действия, выполняющие роль операторов управления: логические операторы (if-then, switch), операторы цикла (do-while, for, foreach, break, continue) и обработки ошибок (блоки try-catch-throw).

#### Эффективность

Программные роботы особенно эффективны:

- Когда сотрудник работает сразу с несколькими информационными системами.
- Для миграции данных при переходе на новую информационную систему.
- Для автоматизации рутинных операций, выполняемых сотрудниками.
- Для сбора и передачи информации в разных источниках.
- Когда требуется очень быстрая интеграция между системами.
- Для получения и сверки данных из многих источников.

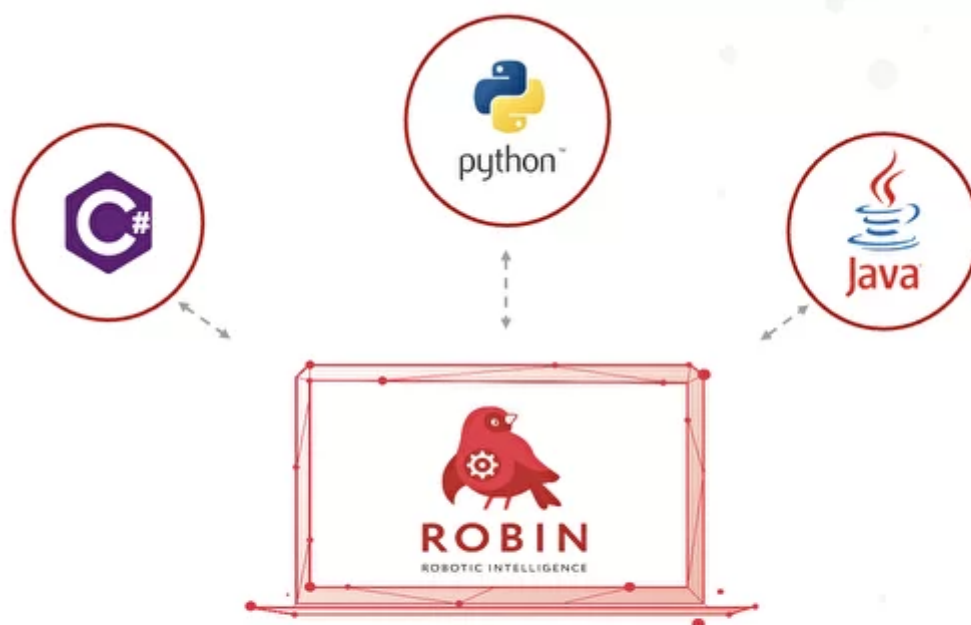
#### Расширяемость

Разработчик с целью расширения функциональных возможностей своих роботов может самостоятельно разработать пользовательские программные действия и использовать их при создании проекта программного робота в Студии.



## Мультиязычность

Мультиязычность платформы ROBIN позволяет разрабатывать дополнительные действия без привлечения узкопрофильных и высокооплачиваемых специалистов. Нет монополии на язык программирования действий – действия для робота могут быть реализованы на языках для любой из трех платформ: .Net (C#, Visual Basic, C++, Jscript), Java, Python.



**Можно использовать любой язык программирования  
Net (C#, VisualBasic, C++, Jscript ), Java, Python**

Штатный программист после базового обучения работе с платформой при необходимости сможет дорабатывать программные роботы ROBIN.

## Кроссплатформенность

ROBIN – кроссплатформенный продукт. Платформа поддерживает исполнение алгоритмов на нескольких операционных системах: Windows, семейство Linux и macOS. Один и тот же робот может выполняться на рабочих местах под любой операционной системой.

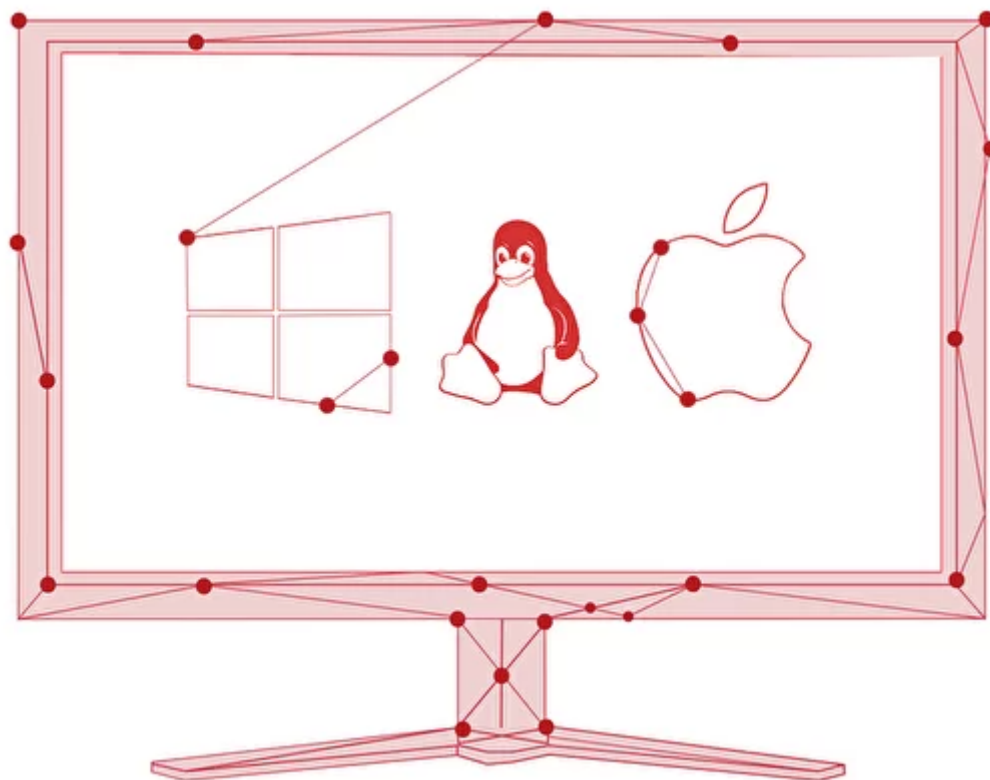
Реальная возможность полного импортозамещения – платформа состоит из компонентов, зарегистрированных или разрешенных реестром российского ПО.

## Библиотека стандартных действий

Библиотека стандартных действий – ключевой компонент Платформы. В библиотеке стандартных действий хранятся и готовы к использованию более 500 программных действий, с помощью которых разработчик в ПО «ROBIN Studio» создает проект программного робота.

Действия из библиотеки стандартных действий позволяют выполнять следующие операции, но не ограничиваются ими:

- выполнять операции над объектами файловой системы – открывать, сохранять, копировать, переименовывать, перемещать файлы и папки, создавать и распаковывать архивы,
- работать с документами в форматах Word, Acrobat PDF, Excel, XML,



- выполнять интеграцию с внешними ресурсами по протоколам HTTP/HTTPS, FTP, SOAP, REST, AMPQ,
- работать с почтовыми службами по протоколам POP3, IMAP, SMTP,
- подключаться к БД и отправлять SQL-запросы на исполнение,
- имитировать работу пользователя с браузерами,
- имитировать работу пользователя с клавиатурой и мышью,
- выполнять действия в циклах типа DoWhile, For, Foreach, с использованием операторов управления циклами Continue и Break,
- выполнять действия с использованием условных операторов If, Switch,
- обрабатывать исключения при выполнении действий,
- конструировать и вычислять арифметические и логические выражения,
- выполнять конвертацию данных,
- выполнять операции над коллекциями,
- выполнять операции над текстом и датами,
- выполнять классификацию текстов с помощью библиотек NLP,
- выполнять операции с таблицами в том числе с Google Sheets,
- распознавать текст на изображениях,
- выполнять работу с оконными интерфейсами приложений Windows,
- выполнять работу с модальными окнами.

### 1.2.5 Исполнения изделия

Изделие имеет следующие варианты исполнений:

- Редакция Desktop в составе:

ПО «ROBIN Studio. Desktop»; ПО «ROBIN Robot. Desktop».

- Редакция Standalone в составе:

АС «ROBIN Orchestrator. Standalone».

- Редакция Enterprise в составе:

АС «ROBIN Orchestrator. Enterprise»; ПО «ROBIN Studio. Enterprise»; ПО «ROBIN Robot. Enterprise»; ПО «ROBIN RMC. Enterprise».

- Редакция Cloud в составе:

АС «ROBIN Orchestrator. Cloud»; ПО «ROBIN Studio. Cloud»; ПО «ROBIN Robot. Cloud»; ПО «ROBIN RMC. Cloud».

**ПО «ROBIN Studio. Desktop»** - лицензированная полнофункциональная версия «ROBIN Studio», предназначенная для разработки и отладки программных роботов, устанавливается на отдельном хосте без подключения к АС «ROBIN Orchestrator».

**ПО «ROBIN Robot. Desktop»** - лицензированная полнофункциональная версия «ROBIN Robot», предназначенная для локального запуска программных роботов, устанавливается на отдельном хосте без подключения к АС «ROBIN Orchestrator».

**АС «ROBIN Orchestrator. Standalone»** - пробная полнофункциональная версия АС «ROBIN Orchestrator», предназначенная для ознакомления пользователя с возможностями Изделия в редакции Enterprise. В этом исполнении все серверные компоненты устанавливаются на одном сервере, имеют полный набор функций, присущий редакции Enterprise, но ограниченный период пользования. Для работы с АС «ROBIN Orchestrator. Standalone» могут использоваться ПО «ROBIN Studio. Enterprise», ПО «ROBIN Robot. Enterprise» и ПО «ROBIN RMC. Enterprise».

**АС «ROBIN Orchestrator. Enterprise»** - версия, предназначенная для использования на Предприятии. Может быть развернута в пределах защищенного контура корпоративной сети на ресурсах Предприятия (On-Premise). В этом случае собственником, эксплуатантом и выгодоприобретателем является само Предприятие, приобретающая права собственности на Изделие в результате его покупки у разработчика. Для работы с АС «ROBIN Orchestrator. Enterprise» используются ПО «ROBIN Studio. Enterprise», ПО «ROBIN Robot. Enterprise» и ПО «ROBIN RMC. Enterprise».

**АС «ROBIN Orchestrator. Cloud»** - облачная версия. Система поставляется на рынок как RPA-платформа (PaaS). При этом собственником Системы является Провайдер Системы, а эксплуатантом – контрагенты, заинтересованные в роботизации рутинных бизнес-процессов на своих Предприятиях. Права собственности на Систему Провайдер приобретает в результате ее покупки у разработчика. А контрагенты приобретают право на пользование услугами Системы путем заключения договоров с Провайдером Системы. Для работы с АС «ROBIN Orchestrator. Cloud» используются ПО «ROBIN Studio. Cloud», ПО «ROBIN Robot. Cloud» и ПО «ROBIN RMC. Cloud».

В версиях АС «ROBIN Orchestrator. Enterprise» и АС «ROBIN Orchestrator. Cloud» серверные компоненты могут быть развернуты на отдельных серверах как с образованием кластеров, так и без образования кластеров.

### 1.2.6 Ограничения, накладываемые на область применения изделия

Программные роботы, разрабатываемые с помощью ПО «ROBIN Studio», могут, имитируя действия сотрудника, работать с любыми информационными системами и офисными приложениями через существующий графический интерфейс. Использование программных роботов не требует доработок в архитектуре этих информационных систем.

## 1.3 Условия применения

Для корректного выполнения комплекса программных средств Платформы ROBIN должны быть соблюдены следующие условия.

### 1.3.1 Требования к техническим аппаратным и программным средствам среды функционирования

**Требования к техническим аппаратным и программным средствам среды функционирования клиентских приложений**

Программы «ROBIN Studio», «ROBIN Robot» и «ROBIN RMC» устанавливаются на АРМ со следующими техническими характеристиками, представленными в Таблице ниже.

Характеристики	Значения		
	«ROBIN Studio»	«ROBIN Robot»	«ROBIN RMC»
CPU	2 ядра x 1.8 GHz или лучше	То же самое	То же самое
RAM	8 ГБ	2 ГБ	2 ГБ
HDD/SSD	Не менее 10 ГБ	То же самое	То же самое
Videocard	Поддержка Full HD (1920x1080)	То же самое	То же самое

Для функционирования программ необходимо установить на АРМе следующее программное обеспечение:

- операционная система: одна из - Windows 7, 8,10, Server 2008, Server 2012 x64;
- .NET Framework 5.0.

Для функционирования программы «ROBIN Studio» дополнительно необходимо установить на АРМе следующее программное обеспечение:

- Microsoft Visual C++ Redistributable Package 2015;
- и включить компонент Windows «Microsoft Media Foundation».

Для функционирования программных приложений в редакциях Enterprise и Cloud необходим доступ в сеть Интернет с помощью Wi-Fi или Ethernet. Для обеспечения корректного запуска и выполнения программного робота, необходимо убедиться, что всё используемое роботом специализированное ПО заранее установлено, например: браузеры, офисные пакеты, корпоративные системы и прочее и доступ пользователя к ним обеспечен.

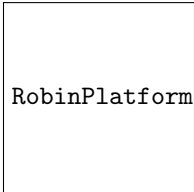
**Требования к техническим аппаратным и программным средствам среды функционирования АС «ROBIN Orchestrator»**

#### Аппаратная конфигурация в исполнении Standalone

Изделие в исполнении АС «ROBIN Orchestrator. Standalone» приобретается для целей оценки состава функциональных возможностей Изделия.

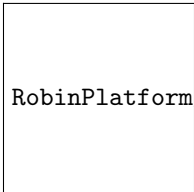
Все серверные компоненты Изделия в исполнении АС «ROBIN Orchestrator. Standalone» размещаются на одном сервере со следующими минимальными требованиями к аппаратному обеспечению.

#### Аппаратная конфигурация в бескластерном исполнении



RobinPlatform/RobinOrchestrator/Rob01.png

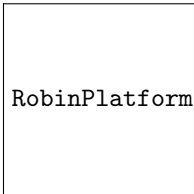
В этой конфигурации все серверные компоненты Платформы ROBIN разворачиваются на отдельных серверах со следующими минимальными требованиями к аппаратному обеспечению, указанные в таблице ниже.



RobinPlatform/RobinOrchestrator/Rob02.png

### Аппаратная конфигурации в кластерном исполнении

В этой конфигурации каждый серверный компонент Платформы ROBIN развернут в кластере для обеспечения отказоустойчивости и возможности дальнейшего масштабирования системы.



RobinPlatform/RobinOrchestrator/Rob03.png

Минимальная аппаратная конфигурация в исполнении cluster может быть такой:

## 1.3.2 Требования организационного характера

Для допуска персонала к работе в системе, сотрудники объекта, на котором производится внедрение Платформы ROBIN должны пройти обучение у специалистов организации, производящей внедрение, и изучить эксплуатационную документацию.

## 1.4 Описание задачи

### 1.4.1 Перечень основных задач

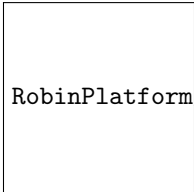
Основные задачи, решаемые с помощью комплекса программных средств Платформы ROBIN:

- Автоматизация процесса создания программных роботов.
- Управление эксплуатацией программных роботов.

### 1.4.2 Автоматизация процесса создания программных роботов

#### Описание задачи

Автоматизация процесса создания программных роботов решается с помощью ROBIN Studio, которая предоставляет разработчику удобную среду для разработки сценариев новых и доработки готовых роботов.


 RobinPlatform/RobinOrchestrator/Rob04.png

## Методы решения задачи

Автоматизация процесса создания программных роботов достигается за счет использования:

- библиотеки готовых стандартных действий – элементарных функциональных блоков, из которых создается сценарий робота;
- визуального конструктора ROBIN Studio, который позволяет с помощью мыши перетаскивать действия из библиотеки стандартных действий на блок-схему робота – при этом функциональный блок действия автоматически захватывается схемой при достаточном сближении его с целевым участком блок-схемы;
- встроенных в Студию механизмов записи пользовательских действий при работе с офисными приложениями – win-рекордера и browser-рекордера;
- встроенного в ROBIN Studio отладчика программных роботов;
- функции сохранения проекта робота как на файловой системе локального АРМ, так и в удаленном хранилище;
- функции экспорта готового программного робота с целью его эксплуатации на АРМе с установленным ПО «ROBIN Robot»;
- функции экспорта проекта с исходниками программного робота с целью его перемещения на АРМ другого разработчика;
- функции публикации готового программного робота в облачном хранилище с целью его последующей эксплуатации другими пользователями;
- функции публикации проекта с исходниками программного робота в облачном хранилище с целью его последующей модификации и доработки другими разработчиками.

### 1.4.3 Управление эксплуатацией программных роботов

Программные роботы исполняются в специальной среде исполнения, поставляемой в составе в составе ПО «ROBIN Studio» с целью отладки робота и «ROBIN Robot» с целью промышленной эксплуатации робота. Управление эксплуатацией программных роботов осуществляется с помощью ПО «ROBIN Robot», если предполагается локальный запуск роботов, и «ROBIN RMC», если предполагается удаленный запуск роботов (только для редакций Enterprise и Cloud).

#### Управление эксплуатацией программных роботов

Программные роботы исполняются в специальной среде исполнения, поставляемой в составе в составе ПО «ROBIN Studio» с целью отладки робота и «ROBIN Robot» с целью промышленной эксплуатации робота. Управление эксплуатацией программных роботов осуществляется с помощью ПО «ROBIN Robot», если предполагается локальный запуск роботов, и «ROBIN RMC», если предполагается удаленный запуск роботов (только для редакций Enterprise и Cloud).

#### Описание задачи (локальный запуск)

Основная задача, решаемая с помощью ROBIN Robot, заключается в предоставлении разработчику возможности запускать программные роботы на локальном АРМе и осуществлять контроль за их исполнением.



## Методы решения задачи (локальный запуск)

Локальный запуск программного робота осуществляется с помощью предустановленного на АРМе ПО «ROBIN Robot», которое позволяет оператору выполнять:

- загрузку программных роботов с внешних носителей в локальный каталог роботов;
- запуск программных роботов;
- просмотр истории запусков;
- настройку и контроль локальных расписаний запусков роботов;
- просмотр логов роботов;
- настройку отправки писем о завершении работы робота.

## Описание задачи (удаленный запуск)

Основная задача, решаемая с помощью ROBIN RMC, заключается в предоставлении разработчику возможности запускать программные роботы на удаленных АРМах и осуществлять контроль за их исполнением.

## Методы решения задачи (удаленный запуск)

Удаленный запуск программного робота осуществляется с помощью предустановленного на АРМе оператора ПО «ROBIN RMC», которое позволяет оператору выполнять:

- просмотр доступных и исполняемых сценариев программных роботов;
- просмотр доступных агентов, на которых могут исполняться программные роботы;
- просмотр истории выполнения сценариев программных роботов;
- настройка и отображение ошибок работы RMC и серверного ПО;
- создание расписания для запуска сценариев роботов;
- скачивание лога с сообщениями компонентов Платформы на АРМ оператора в формате \*.xlsx и \*.csv.

# 1.5 Входные и выходные данные

В данном разделе приводятся сведения о входных и выходных данных ROBIN Studio, ROBIN Robot и ROBIN RMC.

## 1.5.1 ROBIN Studio

### Входные данные

Входными данными для ROBIN Studio являются:

- детальное описание роботизируемого процесса, составленное бизнес-аналитиком, позволяющее разработчику по этому описанию с помощью визуального конструктора составить сценарий будущего программного робота;
- входные параметры действий, взятые из описания роботизируемого процесса и задаваемые разработчиком через предоставляемые формы ROBIN Studio;
- записи пользовательских действий, полученные с помощью рекордеров Студии и используемые разработчиком для составления сценария будущего программного робота;

- команды, инициированные разработчиком через управляющие элементы графического интерфейса программы, позволяющие выполнять операции по загрузке, редактированию, сохранению, отладке и публикации готового робота и его проекта;
- файл, имеющий расширение .rfs (robot file scheme), с исходником проекта готового или находящегося в разработке программного робота, который можно открыть в ROBIN Studio для просмотра и редактирования.

### Выходные данные

Выходными данными ROBIN Studio являются два файла:

- файл с исходником проекта робота с расширением .rfs (robot file scheme), содержащий схему алгоритма робота, которую можно открыть в ROBIN Studio для просмотра и редактирования;
- файл с расширением .rpr (robin program robot), который содержит конфигурационные и исполняемые файлы программного робота. Файл с данным расширением можно запускать только на компьютерах с заранее установленным компонентом ROBIN Robot.

## 1.5.2 ROBIN Robot

### Входные данные

Входными данными для ROBIN Robot являются:

- каталог файлов с расширением .rpr (robin program robot), который содержит конфигурационные и исполняемые файлы программного робота;
- команды, инициированные разработчиком через управляющие элементы графического интерфейса программы, позволяющие выполнять операции:
  1. загрузки и запуска программных роботов;
  2. просмотра истории запусков;
  3. настройки и контроля локальных расписаний запусков роботов;
  4. просмотра логов роботов;
  5. настройки отправки писем о завершении работы робота.

### Выходные данные

Выходными данными ROBIN Robot являются:

- расписания локальных запусков роботов;
- истории запусков роботов;
- результаты, полученные по завершении исполнения сценариев роботов;
- уведомления, полученные по электронной почте, о запуске роботов и завершении выполнения их сценариев;
- диагностические сообщения, полученные от среды исполнения программных роботов о возможных ошибках, имевших место во время исполнения сценариев роботов.

## 1.5.3 ROBIN RMC

### Входные данные

Входными данными для ROBIN RMC являются:

- каталог файлов с расширением .grg (robin program robot), который содержит конфигурационные и исполняемые файлы программного робота;
- команды, инициированные разработчиком через управляющие элементы графического интерфейса программы, позволяющие выполнять операции:
  1. загрузки и запуска программных роботов;
  2. просмотра истории запусков;
  3. настройки и контроля локальных расписаний запусков роботов;
  4. просмотра логов роботов;
  5. настройки отправки писем о завершении работы робота;
- информация о доступных агентах.

### Выходные данные

Выходными данными ROBIN Robot являются:

- расписания удаленных запусков роботов;
- истории запусков роботов;
- результаты, полученные по завершении исполнения сценариев роботов;
- уведомления, полученные по электронной почте, о запуске роботов и завершении выполнения их сценариев;
- диагностические сообщения, полученные от среды исполнения программных роботов о возможных ошибках, имевших место во время исполнения сценариев роботов.

---

ROBIN Studio. Руководство пользователя

---

## 2.1 Общая информация

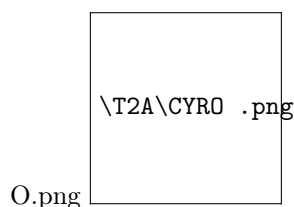
**ROBIN Studio 2.0** - модуль создания программных роботов. Робот создаётся с помощью визуального конструктора, где алгоритмы работы создаются в формате блок-схем.

**Блок-схема** — распространенный тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности. Каждый блок представляет собой уникальное действие. Платформа Robin поддерживает возможность создания действия, не предусмотренного стандартным функционалом студии, и добавления его в Robin Studio.

Robin Studio 2.0 предназначена для автоматизации (роботизации) бизнес-процессов различных компаний посредством создания и выполнения программными роботами рутинных операций вместо пользователей АС. Роботы работают строго по заданному регламенту, имитируя действия соответствующего сотрудника в информационных системах через уже имеющийся графический интерфейс. Так же преимуществом роботизации является неподверженность человеческому фактору и более высокая скорость выполнения задач.

**Целью создания Robin Studio 2.0** является повышение эффективности работы сотрудников компаний за счет автоматизации рутинных операций и уменьшение времени, затрачиваемого на обработку данных операций, а также за счет снижения операционных и технических ошибок.

Примеры операций, которые робот может выполнить за человека:



Создавать роботов в Robin Studio возможно без программирования (ни один элемент или действие не требует написания программного кода для сборки программного робота), обладая минимальной классификацией работы в визуальном редакторе, что делает работу со студией возможной для существующих сотрудников или с привлечением недорогих ресурсов. В студии есть встроенная

интерактивная документация – подсказка о конкретном действии, а также возможность указывать комментарии к шагам робота и переменным.

ПО «ROBIN Studio» представлено тремя редакциями:

- «ROBIN Studio. Desktop Edition» - предназначено для запуска на рабочей станции, не подключенной к корпоративной сети,
- «ROBIN Studio. Enterprise Edition» - предназначено для запуска на рабочей станции, подключенной к корпоративной сети,
- «ROBIN Studio. Cloud Edition» - предназначено для запуска на рабочей станции, расположенной в облаке.

Среда исполнения роботов ПО «ROBIN Studio» обладает свойством кроссплатформенности, т.е. обеспечивает выполнение сценариев, действия которых могут требовать для своего исполнения любую из сред - .Net, Java, Python.

**Robin Studio 2.0 обладает следующими функциональными возможностями:**

1. Работа в браузере: сбор информации и наполнение контентом веб-страницы;
2. Работа в офисных приложениях: сбор, сортировка, редактирование и внесение данных;
3. Отправка, получение и отбор писем с применением фильтров;
4. Работа в desktop-приложениях: запуск приложения и имитация действия пользователя при работе с ним;
5. Работа с файлами: создание, копирование, удаление, перемещение;
6. Поиск визуальных элементов по картинке с помощью машинного зрения;
7. Работа с базами данных: получение и заполнение таблиц в базе данных при помощи SQL-запросов;
8. Извлечение сущности из текста по настроенным шаблонам, проверка и изменение их;
9. Преобразование данных в новые форматы, формирование отчетных документов.
10. Быстрое создание роботов в режиме Recorder, который позволяет автоматически записать действия пользователя и преобразовать их в сценарии/алгоритм для робота;
11. Использование любого созданного робота в качестве «действия» для применения в любых других роботах.

**Выгоды от использования Robin Studio 2.0:**

1. Создание роботов с небольшими затратами;
2. Отсутствие необходимости навыков программирования у сотрудников;
3. Широкий спектр решаемых посредством студии задач;
4. Быстрое создание роботов;
5. Повышение эффективности работы сотрудников компаний за счет автоматизации рутинных операций;
6. Работа роботов не подвержена человеческому фактору.

## 2.2 Требования к конфигурации ПК для ROBIN Studio 2.0

**Аппаратно-программные требования к конфигурации ПК для ROBIN Studio и ROBIN Robot:**

- CPU: 2 ГГц или быстрее. Два ядра или больше.
- RAM: 8 Гб или больше.

- HDD: 10 Гб или больше свободного места.
- Видеоподсистема поддерживает разрешение дисплея WXGA (1366x768) или выше

#### Требования к системному ПО:

- Microsoft Windows 7 Professional или Enterprise (SP1) с обновлением ESU , 8,10, Server 2008 r2 (SP1) с обновлением ESU, 2012, 2016
- .NET Framework 4.8
- .NET Desktop Runtime 5.0.7
- Microsoft Visual C++ Redistributable Package 2015
- Включенный компонент Windows «Media Foundation» (Нужен для работы действий группы «Машинное зрение»).

При установке на Windows Server 2012R, для работы функций зрения на компьютерах с операционной системой необходимо установить компонент операционной системы (Feature).

Для этого необходимо в PowerShell выполнить команду:

*Install-WindowsFeature server-media-foundation*

## 2.3 Установка ROBIN Studio

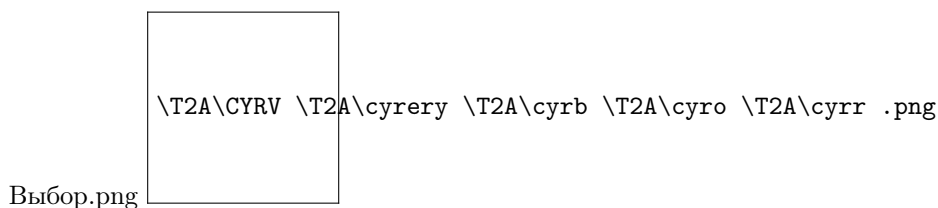
В данном разделе описана установка и настройка Программы для трех редакций – Desktop, Cloud, Enterprise. Установка программного обеспечения производится с помощью мастера установки. Во время установки параметрам конфигурации Программы (адреса, порты, логины, пароли, протоколы и т.д.) будут присвоены значения по умолчанию.

### 2.3.1 Установка ROBIN Studio для редакции Desktop

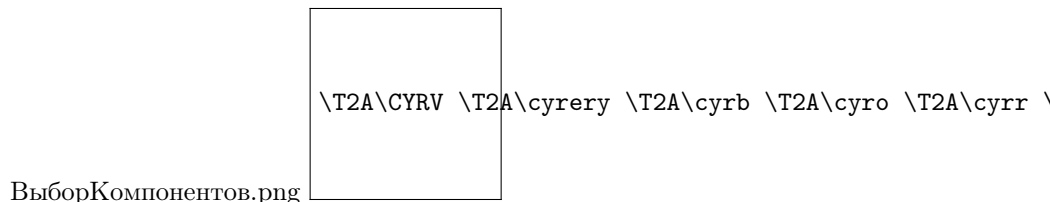
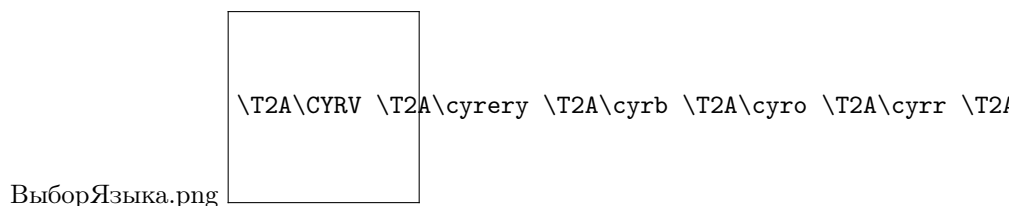
Для установки ПО необходимо скачать по ссылке, предоставленной разработчиками или загрузить на АРМ файл с дистрибутивом ПО, имеющим название Robin Platform-std\_ordinal\_mix-2.0.0-RTM-Standalone

Для установки с помощью мастера установки необходимо выполнить следующую последовательность действий:

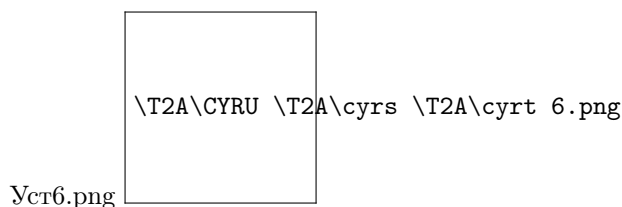
1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла.
2. Выбрать режим установки Программы - для одного или нескольких пользователей АРМа.



3. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].
4. Выбрать компоненты для установки (рекомендуется выбрать все компоненты) и нажать на кнопку [Далее].



5. Подтвердить начало установки, нажав на кнопку [Установить], программа будет установлена в папку по адресу C:/Program Files/Robin Platform.



6. Далее начнется процедура установки.
7. Для завершения установки необходимо нажать на кнопку [Завершить].

---

**Важно:** Для установки файла лицензии, необходимо скопировать файл вида license\*.lic в папку C:Program FilesRobin Platform2.0.0RobinStudio.

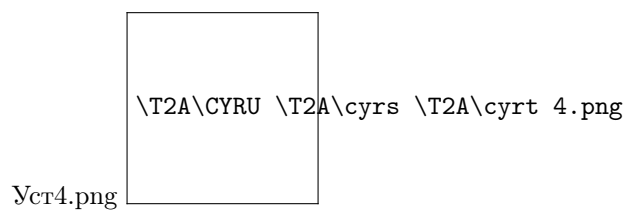
---

### 2.3.2 Установка ROBIN Studio для редакции Enterprise

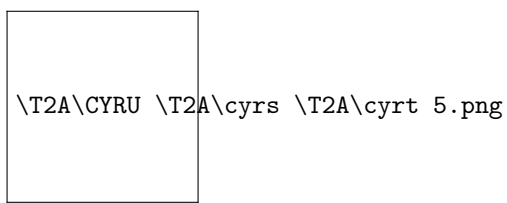
Для установки ПО необходимо скачать по ссылке, предоставленной разработчиками или загрузить на АРМ файл с дистрибутивом ПО, имеющим название Robin Platform-WO\_ENV\_mix-2.0.0-RTM-Enterprise.

Для установки с помощью мастера установки, необходимо выполнить следующую последовательность действий:

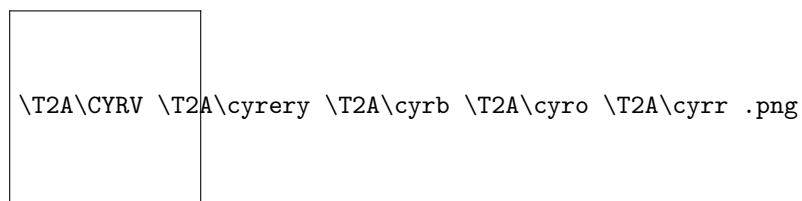
1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла.
2. Выбрать режим установки Программы - для одного или нескольких пользователей АРМа.
3. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].
4. Выбрать компоненты для установки (рекомендуется выбрать все компоненты) и нажать на кнопку [Далее].



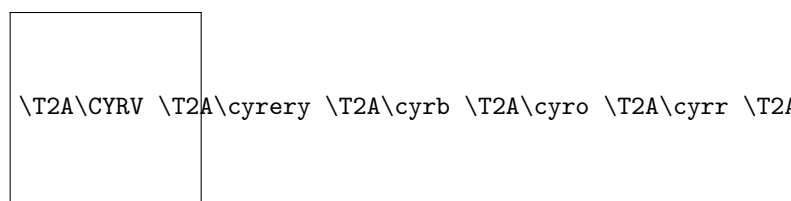
Устр4.png



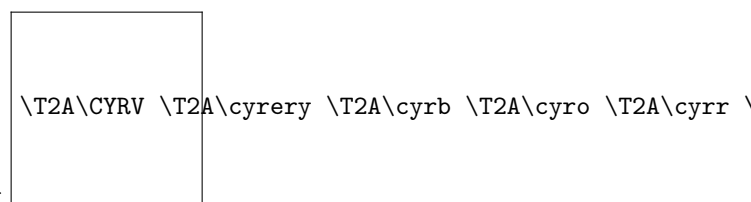
Устр5.png



Выбор.png



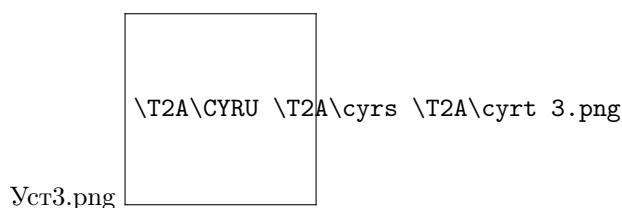
ВыборЯзыка.png



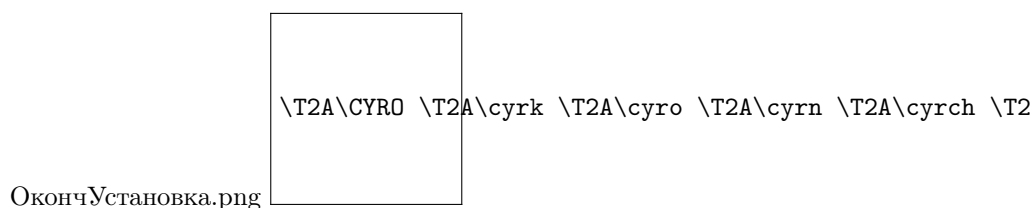
ВыборКомпонентов.png



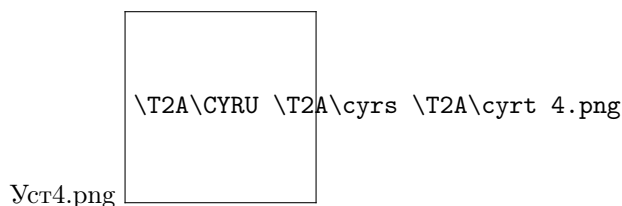
5. Ввести IP-адрес ROBIN сервера и нажать на кнопку [Далее].



6. Подтвердить начало установки, нажав на кнопку [Установить], программа будет установлена в папку по адресу C:/Program Files/Robin Platform.



7. Далее начнется процедура установки.



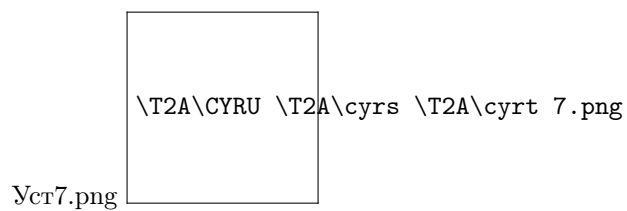
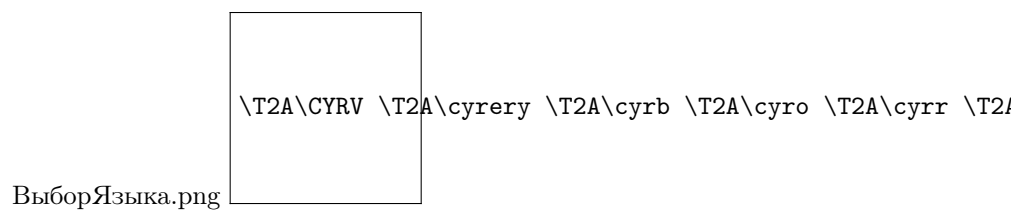
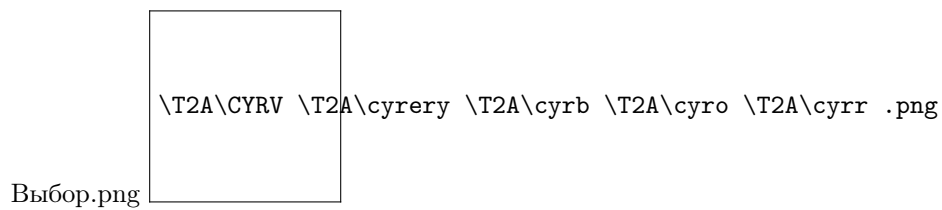
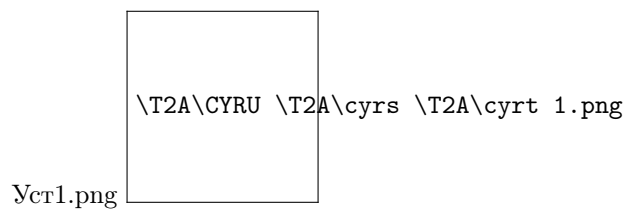
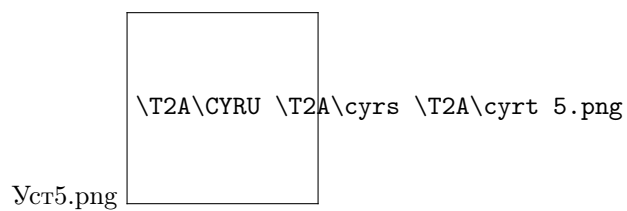
8. Для завершения установки необходимо нажать на кнопку [Завершить].

### 2.3.3 Установка ROBIN Studio для редакции Cloud

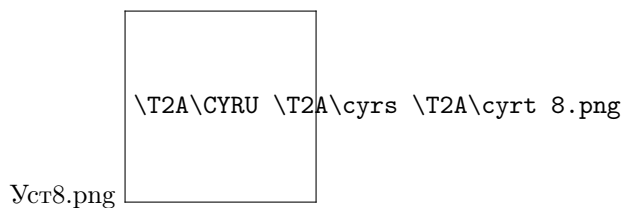
Для скачивания ПО необходимо зарегистрироваться на портале «Облачная фабрика роботизации (ОФР)», адрес портала зависит от реализации проекта. Затем необходимо перейти в раздел портала «Дистрибутивы» и нажать на кнопку [Скачать] в строке с названием «Robin Studio».

Для установки ПО необходимо выполнить следующую последовательность действий:

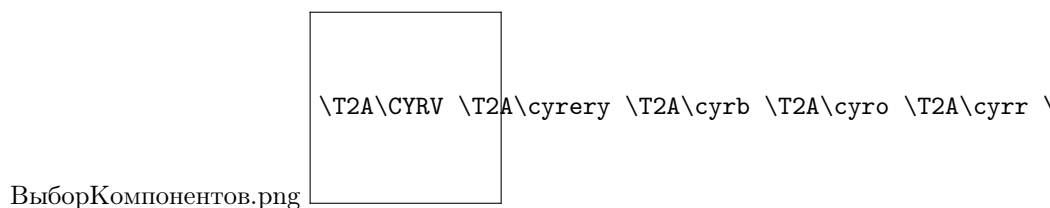
1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла Robin Platform-Studio -2.5.0-RTM-Cloud.exe.
2. Выбрать режим установки Программы - для одного или нескольких пользователей АРМа.
3. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].
4. Ввести токен, который был сгенерирован для пользователя портала ОФР, и нажать на кнопку [Далее].



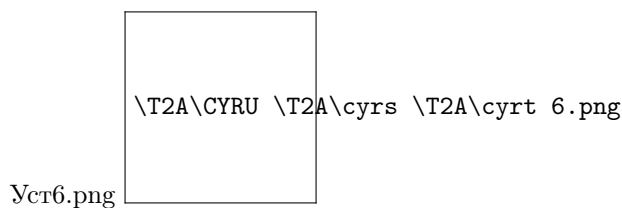
Токен пользователя находится в правом верхнем углу под логином пользователя на портале ОФР.



5. Выбрать компоненты для установки (рекомендуется выбрать все компоненты) и нажать на кнопку [Далее].



6. Подтвердить начало установки, нажав на кнопку [Установить], программа будет установлена в папку по адресу C:/Program Files/Robin Platform.



7. Далее начнется процедура установки.
8. Для завершения установки необходимо нажать на кнопку [Завершить].

## 2.4 Главное меню

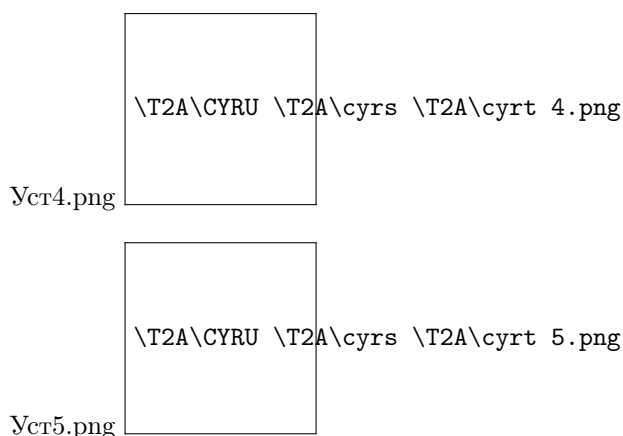
### 2.4.1 Создание нового робота

Для создания новой схемы робота ROBIN Studio есть несколько способов:

#### Способ 1

В меню выбора схемы робота, которую необходимо открыть. Необходимо нажать на кнопку в форме зеленого плюсика.

После нажатия на данную кнопку будет предложено ввести название для нового робота, а затем необходимо нажать на зеленую кнопку «Создать». После этого произойдет переход в рабочее пространство ROBIN Studio.



### Способ 2

Нажав на кнопку, обведенную красной рамкой на скринке ниже, прямо из рабочего пространства ROBIN Studio также можно попасть на форму создания новой схемы робота.

После нажатия на кнопку появится всплывающее окно с предложением сохранить текущую схему робота.

После нажатия на кнопку «Сохранить» или на кнопку «Нет», произойдет выполнение выбранного действия и переход к следующей форме (далее см. Способ №1).

### Способ 3

Произвести переход в меню, затем выбрать пункт «Список роботов».

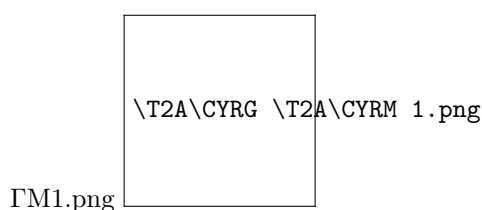
После клика левой клавишей мыши по данному пункту появится всплывающее окно с предложением сохранить текущую схему робота.

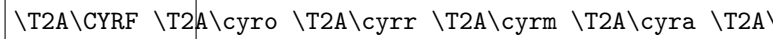
Вне зависимости от выбранного варианта действия («Сохранить» или «Нет»), данное окно будет закрыто с последующим выполнением выбранного действия, затем появится следующая форма:

После нажатия на кнопку «Сохранить» или на кнопку «Нет», произойдет выполнение выбранного действия и переход к следующей форме (далее см. Способ №1).

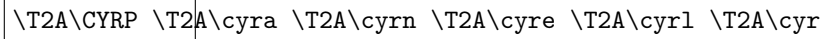
## 2.4.2 Открытие существующей схемы робота

Для продолжения редактирования уже существующей схемы робота при повторном открытии ROBIN Studio, нажмите на кнопку «Открыть» в группе действий в форме «Роботы». После нажа-





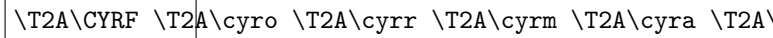
ФормаРоботы1.png



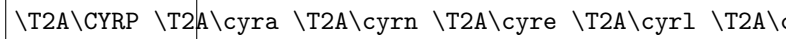
Панель.png



Окно.png



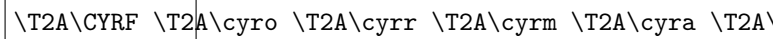
ФормаРоботы2.png



Панель Меню.png

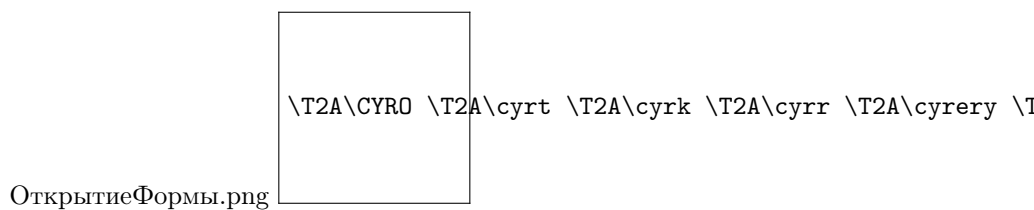


Окно1.png



ФормаРоботы3.png

тия кнопки «Открыть» откроется рабочее пространство ROBIN Studio.



В таблице, указанной ниже приведено описание основных элементов и их назначение на форме «Роботы».

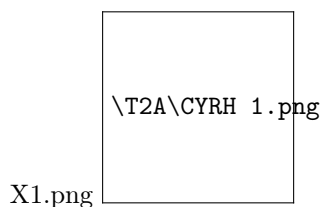
Форма «Роботы»		
Кнопка «Создать нового робота»	При нажатии на кнопку появится текстовое поле для ввода названия робота. После ввода названия робота, нажав на кнопку «Создать», откроется рабочее пространство для построения блок-схемы робота.	<div>Плюс.png</div> <div>\T2A\CYRP \T2A\cyrl \T2A\cyru \T2A\cy</div>
Кнопка «Открыть»	При нажатии на кнопку произойдет открытие уже существующей схемы робота для просмотра и/или редактирования схемы.	<div>Открыть.png</div> <div>\T2A\CYRO \T2A\cyrt \T2A\cyrk \T2A\cy</div>
Загрузить робота с компьютера	Открывается окно проводника для выбора файла с роботом.	<div>Загрузить.png</div> <div>\T2A\CYRZ \T2A\cyra \T2A\cyrg \T2A\cy</div>
Строка поиска	Текстовое поле для поиска уже ранее добавленных роботов.	<div>Поиск.png</div> <div>\T2A\CYRP \T2A\cyro \T2A\cyri \T2A\cy</div>
Настройки	Открывается окно настроек ROBIN Studio.	<div>Настройки11.png</div> <div>\T2A\CYRN \T2A\cyra \T2A\cyrs \T2A\cy</div>
Хранилище	Открывается список хранилищ.	<div>Хранилище.png</div> <div>\T2A\CYRH \T2A\cyrr \T2A\cyra \T2A\cy</div>

### 2.4.3 Подключение хранилища

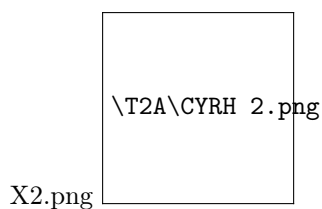
В ROBIN Studio 2.0 есть возможность подключения хранилища исходных и готовых роботов.

Для подключения хранилища необходимо:

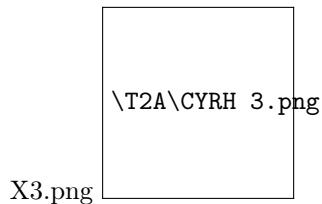
1. Перейти в настройки, на панели, которая отображается при запуске ROBIN Studio:



Или при открытии списка роботов, или через меню в интерфейсе студии:



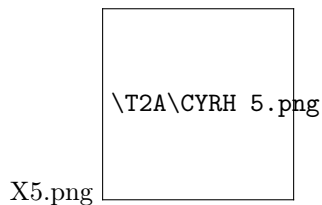
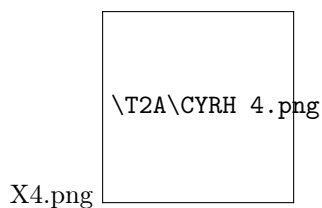
2. В списке слева выбрать «Хранилище готовых роботов».



3. Нажать кнопку «Новое».
4. В открывшемся поле «Новое хранилище» заполнить следующие данные:
  - Название хранилища
  - URL-Адрес
  - Логин
  - Пароль
5. Нажать кнопку «Подключить хранилище»

В таблице ниже приведено описание элементов и их действия на форме «Подключить хранилище».

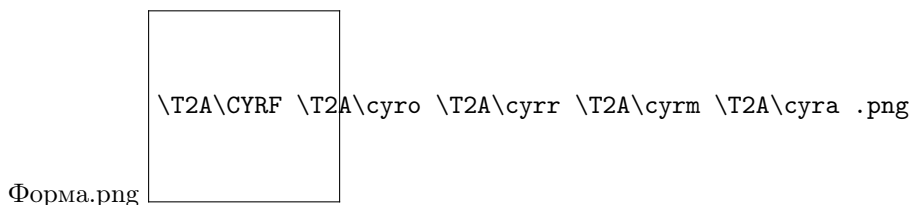




Форма «Подключить хранилище»	
Поле «Название хранилища»	Текстовое поле для указания наименования подключаемого хранилища.
Поле «URL-адрес»	Текстовое поле для указания URL-адреса для подключаемого хранилища.
Поле «Логин»	Текстовое поле, используемое для ввода логина для подключаемого хранилища.
Поле «Пароль»	Текстовое поле для ввода пароля при входе в хранилище.
Кнопка «Подключить хранилище»	Используется для сохранения введенных данных и подключения хранилища.

#### 2.4.4 Загрузка робота с компьютера или флешки

При нажатии на кнопку «Загрузить» откроется окно, в котором необходимо выбрать файл робота для загрузки (кнопка выделена красной рамкой на скриншоте).

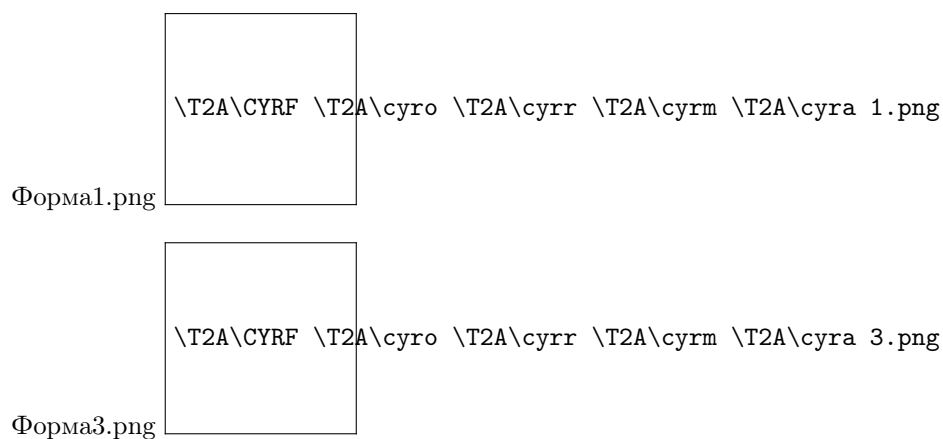


Далее будет открыто окно, в котором необходимо выбрать файл робота для загрузки.

#### 2.4.5 Настройки

Данный раздел служит для изменения настроек ROBIN Studio 2.0. На форме «Роботы» следует нажать на «Настройки».

Вторым способом является переход через меню в верхней панели рабочего пространства и выбор «Настройки программы».

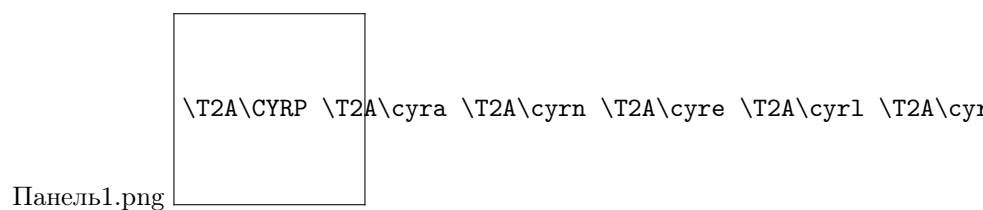


Далее, будет открыта форма, где имеется возможность установить настройки языка, внешнего вида платформы, время автосохранения, историю действий, вид условий/ветвлений. Также имеется возможность настройки хранилища и сертификатов. Изменения в настройках сохраняются автоматически.

## 2.5 Верхняя панель интерфейса

На верхней панели интерфейса располагается основное меню и кнопки необходимые в процессе создания робота.

В таблице ниже указано предназначение кнопок на верхней панели:



Настройки1.png

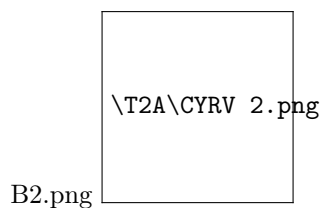
\T2A\CYRN \T2A\cyra \T2A\cyrs \T2A\cyrt \T2A\cyrr \T2A\c

ВерхПанель.png

\T2A\CYRV \T2A\cyre \T2A\cyrr \T2A\cyrh \T2A\CYRP \T2A\c

Иконка	Название	Действие
ВП1.png 	Меню	Переход в главное меню ROBIN Studio.
ВП2.png 	Сохранить робота	Сохранение изменений, внесенных на схе- му робота. Схема сохраняется в папку C:\Users\UserName\DocumentsRobinStudioProjects\ProjectName%.
ВП3.png 	Опубли- ковать проект робота	Позволяет опубликовать проект робота в хранилище.
ВП4.png 	Опубли- ковать готового робота	Позволяет опубликовать готового робота в хранилище.
ВП5.png 	Специ- фикации	Позволяет изменить информацию о роботе, входные и выходные зна- чения, отслеживать версии движков, а также настроить робота (Бра- узеры, E-mail, Необходимое ПО, Требования к ресурсам).
ВП6.png 	Запись действий	Позволяет записать действия, произведенные во время работы в брау- зере или Windows.
ВП7.png 	База зна- ний	При нажатии открывается web-сайт со справкой по работе в ROBIN Studio.
ВП8.png 	Создать нового робота	Переход к экранной форме «Роботы», где имеется возможность созда- ния нового робота путем нажатия на зеленую кнопку плюса.
ВП9.png 	Специ- фикация	Переход к настройке спецификации робота.
В10.png 	Поиск	Осуществление поиска по действиям или экземплярам.
ВП10.png 	Массовые	Применение массовых действий на схеме робота. Например: включить



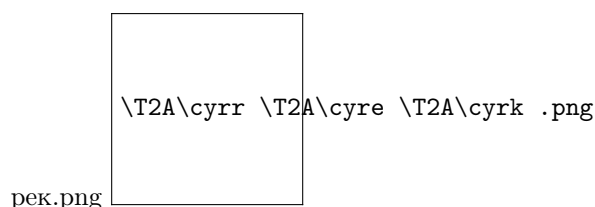


5. После, записанный алгоритм будет отображен на рабочем пространстве студии в виде действий;
6. Запустить робота. Робот выполнит необходимый алгоритм.

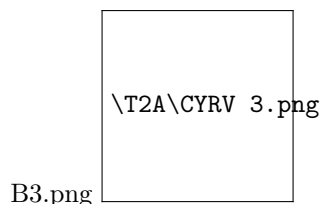
### Запись действий при работе с десктопными приложениями

Для записи действий при работе с десктопными приложениями:

1. Нажать кнопку Rec. Она находится на верхней панели интерфейса;

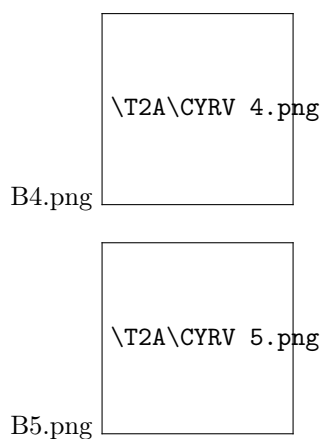


2. В открывшемся окне перейти на вкладку «Запись действий Win»;



3. Нажать кнопку «Начать»;
4. Произвести необходимые действия с десктопным приложением. Записанные действия будут отображаться в окне рекордера;
5. Остановить запись действий с помощью нажатия кнопки «Stop recording» в окне рекордера или с помощью кнопки «Стоп» на вкладке «Запись действий Win» в студии;
6. После, записанный алгоритм будет отображен на рабочем пространстве студии в виде действий;
7. Запустить робота. Робот выполнит необходимый алгоритм.

При работе с десктопными приложениями работа осуществляется аналогично работе в браузере, т.е. необходимо произвести требуемые действия в приложении, и далее они будут отображены на схеме робота в виде действий.



### 2.5.2 Массовые действия

При работе со схемой робота иногда возникает потребность применения массовых действий к действиям из которых состоит робот.

Для применения массовых действий к схеме робота необходимо на верхней панели нажать кнопку.

При нажатии на кнопку «Массовые действия» будет открыто меню.

В таблице приведены примеры работы с массовыми действиями:

Массовые действия так же применяются при работе с блоками Try-Catch, циклами, условиями.

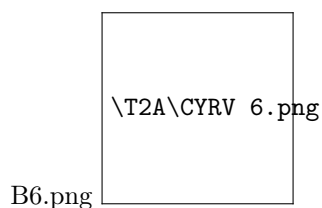
А также массовые действия применимы для работы с ветвлениями.

### 2.5.3 Меню верхней панели

Меню вызывается кнопкой в верхней панели ROBIN Studio:

После клика по иконке меню появится выпадающая форма:

Из этой формы можно перейти к списку роботов для открытия одной из существующих схем роботов, получить доступ к настройкам программы, а также можно выбрать какие-либо параметры, относящиеся к элементам: робот, операции с действиями, массовые операции, параметры запуска, debug, вид, горячие клавиши, руководство пользователя, настройки программы.



\T2A\CYRV 7.png

B7.png

\T2A\cyrm \T2A\cyra \T2A\cyrs \T2A\cyrd .png

масд.png

\T2A\cyrm \T2A\cyre \T2A\cyrn \T2A\cyryu \T2A\cyrm \T2A\cy

менюмд.png

\T2A\cyrt \T2A\cyra \T2A\cyrb \T2A\cyr1 \T2A\cyri \T2A\cy

таблица1.png

\T2A\cyrt \T2A\cyra \T2A\cyrb \T2A\cyr1 \T2A\cyri \T2A\cy

таблица2.png

\T2A\cyrt \T2A\cyra \T2A\cyrb \T2A\cyr1 \T2A\cyri \T2A\cy

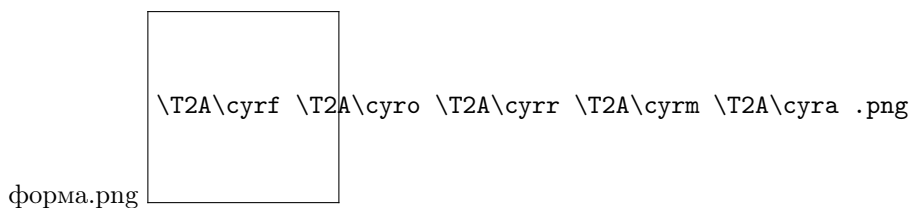
таблица3.png

\T2A\cyrm \T2A\cyre \T2A\cyrn \T2A\cyryu .png

меню.png

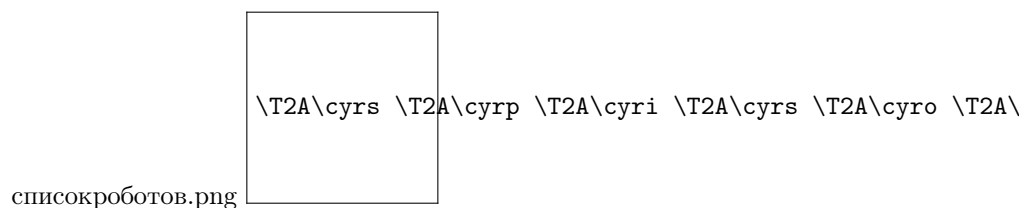
\T2A\CYRV \T2A\cyre \T2A\cyrr \T2A\cyrh \T2A\CYRP \T2A\c

ВерхПанель.png

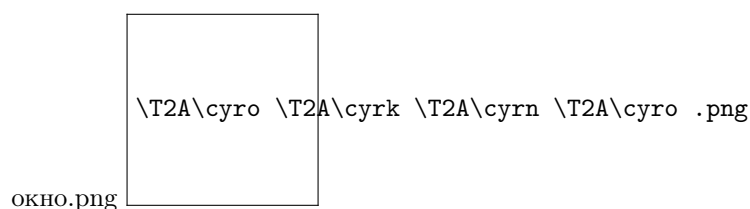


## Список роботов

Для перехода к списку роботов необходимо выбрать в меню ROBIN Studio «Список роботов».



Далее будет открыто окно с предложением сохранить текущую схему перед закрытием.



Затем будет отображена форма со списком всех доступных для открытия и/или редактирования роботов.

## Робот

При наведении курсора на элемент меню ROBIN Studio «Робот», появится следующая выпадающая форма:

В данной форме можно выбрать такие элементы, как: создание нового робота, сохранение текущего робота, публикация схемы робота в хранилище, публикация готового робота, экспорт готового робота, экспорт исходника робота, печать, спецификация робота, сводная информация.

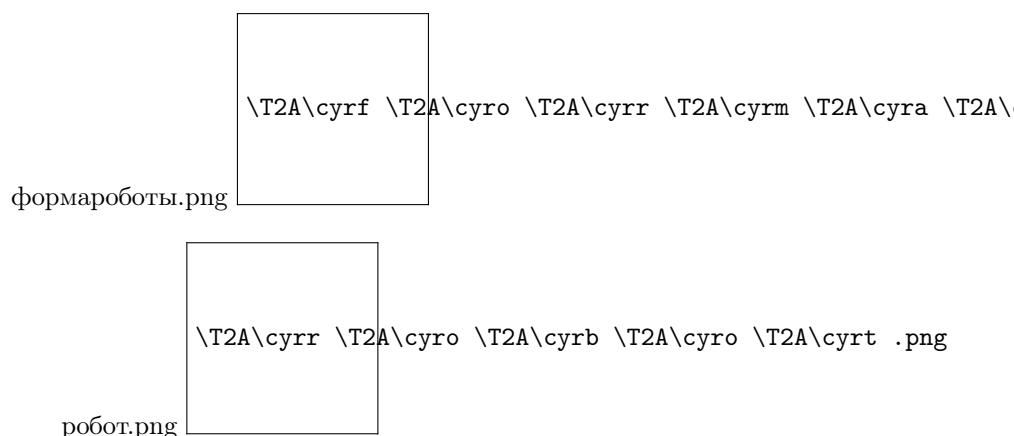
## Создать новый

Создание нового робота возможно несколькими способами, описанными на странице «Создание нового робота при последующих запусках ROBIN Studio.»

## Сохранить

Данная функция отвечает за сохранение проекта по нажатию кнопки «Сохранить» в меню «Робот», а также при нажатии сочетания клавиш Ctrl+S.





### Опубликовать в хранилище

Данная функция отвечает за публикацию схемы робота в хранилище путем нажатия кнопки «Опубликовать в хранилище» в меню «Робот», а также при нажатии сочетания клавиш Ctrl+O.

После нажатия кнопки «Опубликовать в хранилище» появится форма для загрузки робота в хранилище.

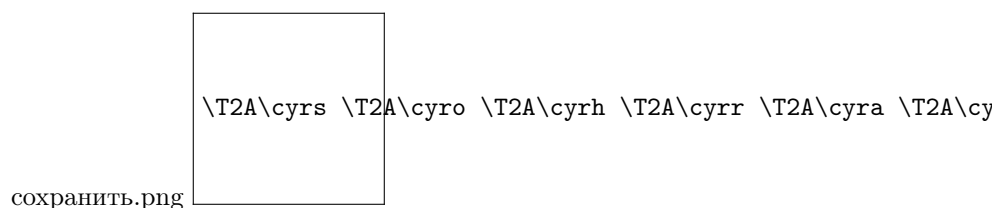
В поле «Хранилище» необходимо выбрать тип хранилища. Также можно указать версию публикуемого робота. «Пространство имен» является не редактируемым полем. При необходимости можно указать комментарии в поле «Изменения в работе». После нажатия на кнопку «Опубликовать» будет произведена публикация робота в хранилище. После нажатия на кнопку «Отмена» будет произведен возврат в рабочее пространство ROBIN Studio.

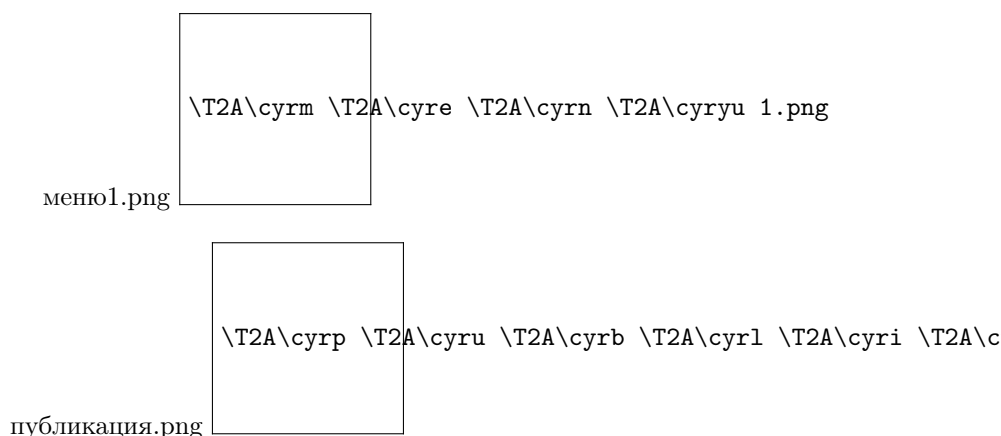
### Публикация готового робота

Данная функция отвечает за публикацию готового робота в хранилище путем нажатия кнопки «Публикация готового робота» в меню «Робот», а также при нажатии сочетания клавиш Ctrl+Shift+O.

После нажатия кнопки «Публикация готового робота» появится форма для загрузки робота в хранилище.

В поле «Хранилище» необходимо выбрать тип хранилища. Также можно указать версию публикуемого робота. «Пространство имен» является не редактируемым полем. При необходимости можно указать комментарии в поле «Изменения в работе». После нажатия на кнопку «Опубликовать» будет произведена публикация робота в хранилище. После нажатия на кнопку «Отмена» будет произведен возврат в рабочее пространство ROBIN Studio.





### Экспорт готового робота

Данная функция отвечает за загрузку готового робота путем нажатия кнопки «Экспорт готового робота» в меню «Робот», а также при нажатии сочетания клавиш Ctrl+Shift+E.

После нажатия кнопки «Экспорт готового робота» появится форма «Обзор папок» для экспорта робота.

После выбора файла и нажатия кнопки «ОК» произойдет экспорт робота. После нажатия кнопки «Отмена» будет произведен возврат в рабочее пространство ROBIN Studio.

### Экспорт исходника робота

Данная функция отвечает за загрузку исходника робота путем нажатия кнопки «Экспорт исходника робота» в меню «Робот».

После нажатия кнопки «Экспорт исходника робота» появится форма «Обзор папок» для экспорта исходника.

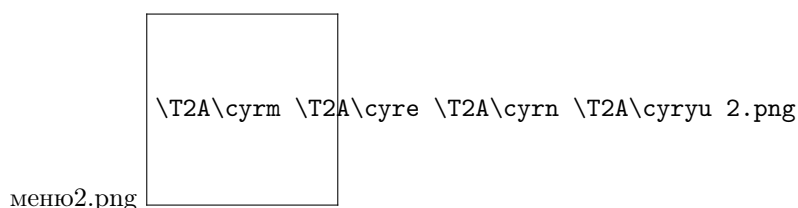
После выбора файла и нажатия кнопки «ОК» произойдет экспорт робота. После нажатия кнопки «Отмена» будет произведен возврат в рабочее пространство ROBIN Studio.

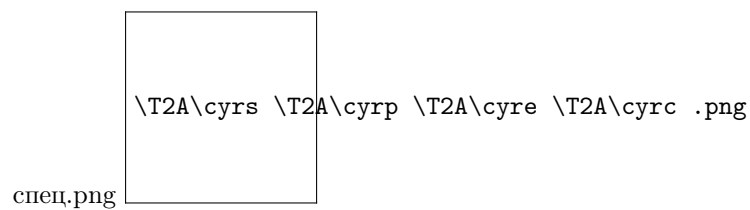
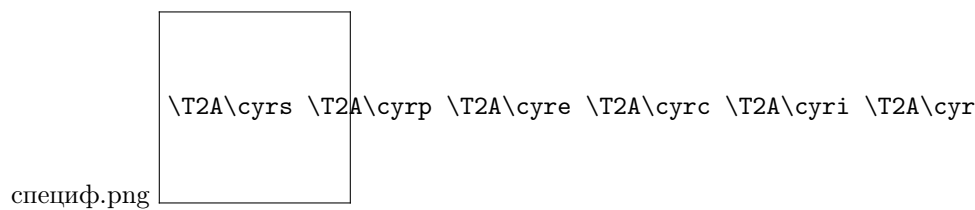
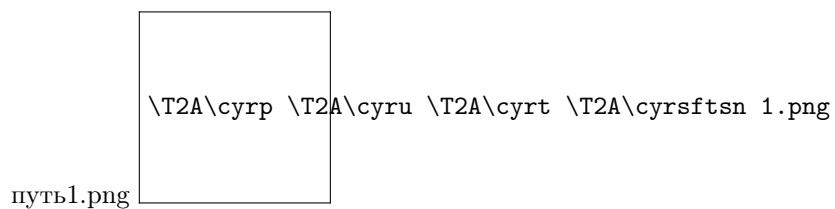
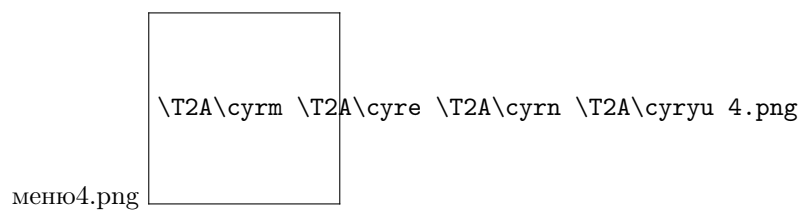
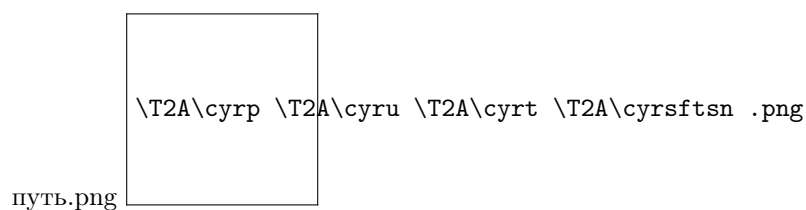
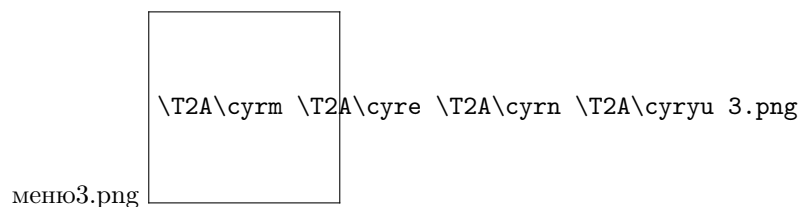
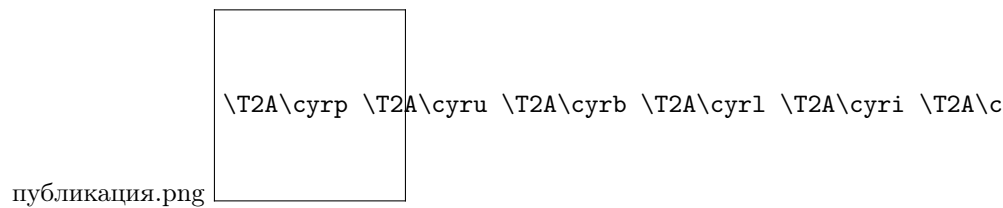
### Спецификация робота

Спецификация робота – данная функция предназначена для сохранения созданной схемы робота в виде одного действия, которое будет расположено в студии в заданной группе и подгруппе.

Для заполнения спецификации робота необходимо перейти по следующему пути: Меню → Робот → Спецификация робота.

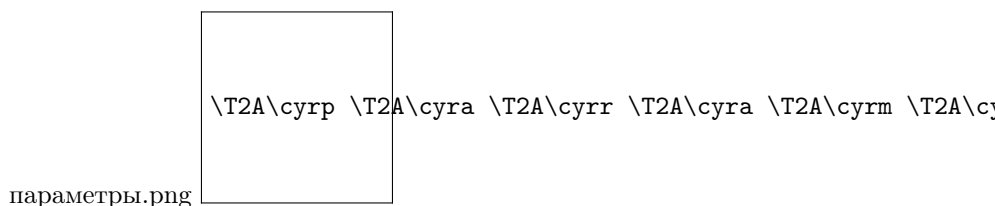
Также в Спецификацию робота можно попасть выбрав на панели кнопку:





### Вкладка «Основная информация»

При нажатии на кнопку «Спецификация робота» в меню «Робот» будет открыта следующая форма.



Во вкладке «Основная информация» указываются наименование контракт действия, заголовок, а также описание робота.

- Контракт действия состоит из трех разделов: «Компания», «Группа» и «Название».

Формат наименования контракта:

- <Vendor>.<Group>.<ActionName>;
- <Vendor> - наименование компании разработчика действия;
- <Group> - наименование группы действия, к которому будет принадлежать действие;
- <ActionName> - название действия.

Во всех названиях действий каждое слово выделяется заглавной буквой, например: MyCustomAction.

- Заголовок может быть как на русском, так и на английском языке. В случае необходимости можно продублировать заголовок на английском/русском языке, для этого необходимо нажать на зеленый плюсик возле слова «Заголовок».

Максимальное количество элементов в блоке «Заголовок» - два, то есть у робота не может быть больше двух заголовков.

- Описание может быть как на русском, так и на английском языке. В случае необходимости можно продублировать описание на английском/русском языке, для этого необходимо нажать на зеленый плюсик возле слова «Описание».

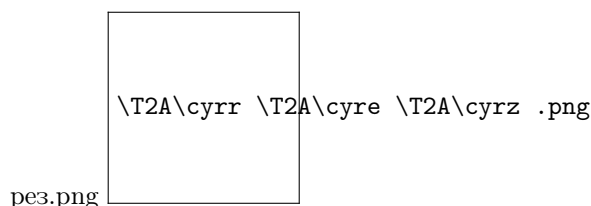
Максимальное количество элементов в блоке «Описание» - два, то есть у робота не может быть больше двух описаний.

После нажатия на кнопку «Сохранить» произойдет сохранение спецификации робота, при нажатии на кнопку «Отмена» произойдет возврат в рабочее пространство ROBIN Studio.

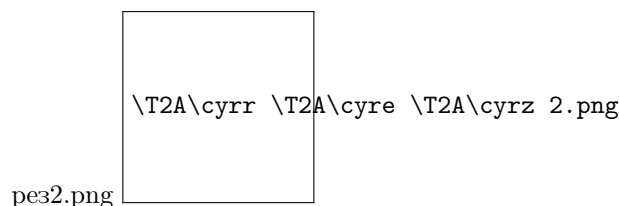
### Вкладка «Параметры»

Вкладка необходима для создания входных параметров, в которые вносятся данные для созданного действия из схемы робота.

В данной форме можно создать необходимые переменные путем нажатия на зеленый плюсик:



Открывается окно с созданием параметра:

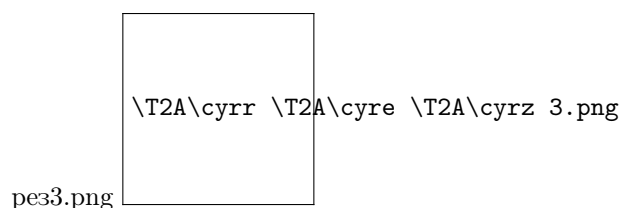


Необходимо будет указать следующие данные:

- Имя переменной (буквы латинского алфавита, цифры, нижнее подчёркивание, тире);
- Тип переменной (выбирается из выпадающего списка);
- Обязательное значение;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний).

Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Заккрыть» произойдет возврат в начальное окно вкладки «Параметры».

Также на вкладке отображен список ранее созданных переменных с возможностью их редактирования. Для этого необходимо нажать на название переменной.



Редактировать созданный параметр можно только по данным:

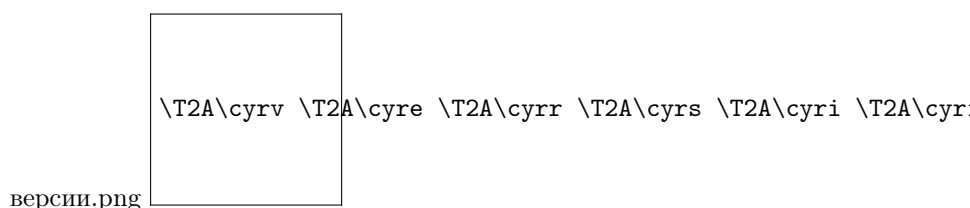
- Обязательное значение;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний);

Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Отмена» произойдет возврат в рабочее пространство ROBIN Studio.

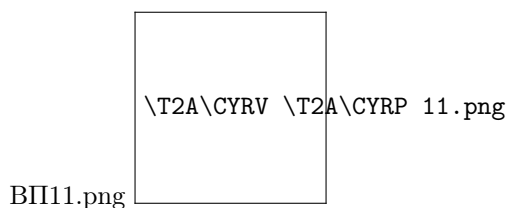
### Вкладка «Результаты»

Вкладка необходима для создания выходных параметров, которые действие, созданное из схемы робота, будет возвращать в переменную «Результат».

В данной форме можно создать переменную для записи в нее результата, путем нажатия на кнопку «Добавить результат»:



Открывается окно с созданием параметра:

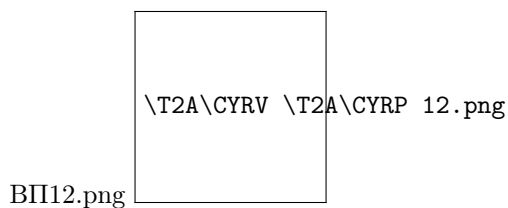


Необходимо присвоить следующие параметры:

- Имя переменной (буквы латинского алфавита, цифры, нижнее подчёркивание, тире);
- Тип переменной (выбирается из выпадающего списка);
- Обязательное значение;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний);

Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Закрыть» произойдет возврат в начальное окно вкладки «Результаты».

Также на вкладке отображен список ранее созданных переменных с возможностью их редактирования. Для этого необходимо нажать на название переменной.



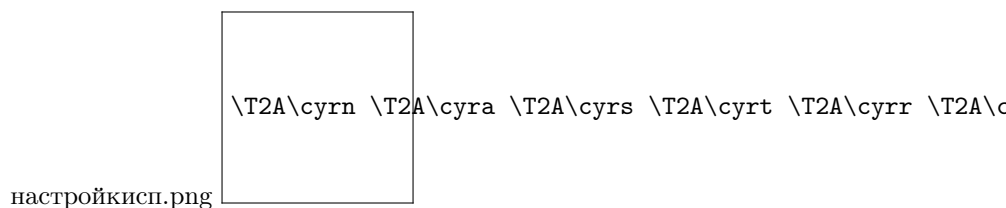
Редактировать созданный параметр можно только по данным:

- Обязательное значение;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний);

Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Отмена» произойдет возврат в рабочее пространство ROBIN Studio.

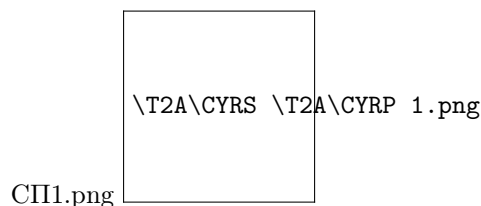
### Вкладка «Настройки робота»

В данной форме указываются Группа, Подгруппа, Супердействие, Контракт.



Для внесения данных по каждому параметру необходимо нажать на кнопку «Добавить новую».

1. Группа – группа действий в студии, в которой будет располагаться созданное действие:

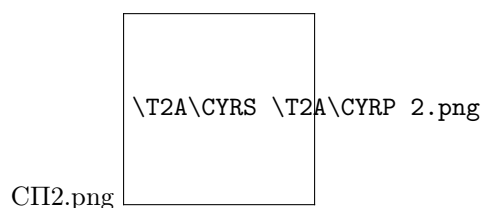


Необходимо присвоить следующие параметры:

- Имя;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний).

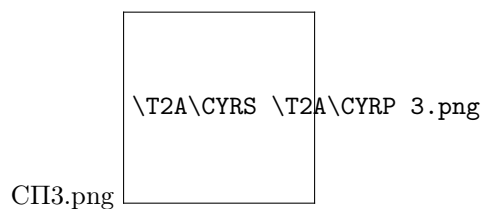
Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Закрыть» произойдет возврат в начальное окно вкладки «Настройки робота».

Также на вкладке будет отображена ранее созданная группа с возможностью ее редактирования либо удаления. Для этого необходимо выбрать нужный знак из выпадающего списка:



Больше одной группы добавить нельзя.

2. Подгруппа – подгруппа действий в студии, расположенная в группе действий, созданной в пункте 1, в которой будет располагаться созданное действие:

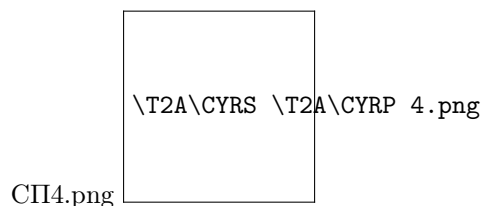


Необходимо присвоить следующие параметры:

- Имя;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний).

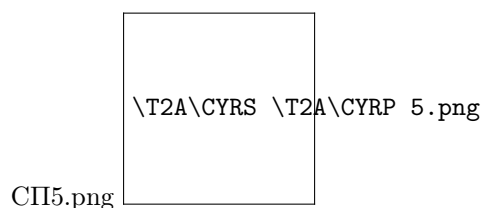
Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Закрыть» произойдет возврат в начальное окно вкладки «Настройки робота».

Также на вкладке будет отображена ранее созданная подгруппа с возможностью ее редактирования либо удаления. Для этого необходимо выбрать нужный знак из выпадающего списка:



Больше одной подгруппы добавить нельзя.

3. Супердействие – включает в себя уникальный идентификатор созданного действия (ID действия), а также его внешний вид в студии (SVG-иконка):



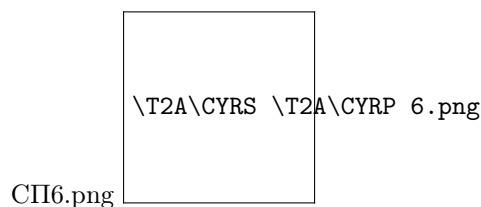
Необходимо присвоить следующие параметры:

- ID действия;
- SVG-иконка;
- Заголовок (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух заголовков);
- Описание (может быть на русском и английском языках, максимальное значение – 2, то есть у робота не может быть больше двух описаний).

Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Закрыть» произойдет возврат в начальное окно вкладки «Настройки робота».

Также на вкладке будет отображено ранее созданное супердействие с возможностью его редактирования либо удаления.

Для этого необходимо выбрать нужный знак из выпадающего списка:

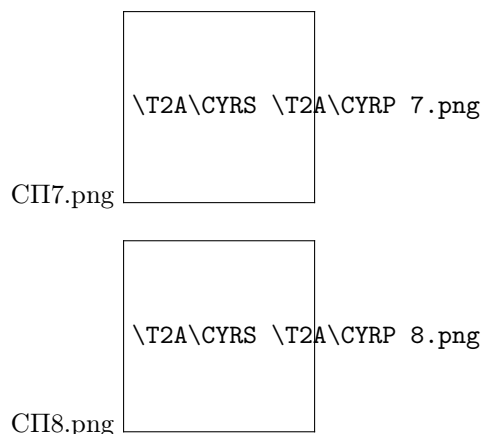


Больше одного супердействия добавить нельзя.

4. Контракт – включает в себя информацию о созданном действии, а именно наименование компании, создавшей контракт, группу и подгруппу, где расположено действие, версию действия и версию платформы, на которой создано действие:

Необходимо присвоить следующие параметры:



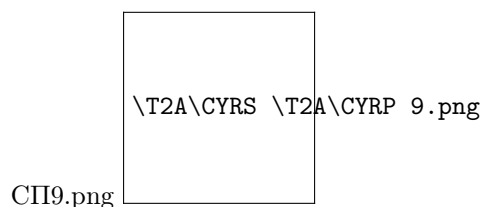


- Контракт, состоит из:
- Платформа – версия платформы, на которой создается действие;
- Версия определения действия, записывается в формате X.Y.Z., из цифр и точек;
- Устаревшее;
- Пользовательская версия.

Для сохранения изменений необходимо нажать на кнопку «Сохранить», при нажатии на кнопку «Закрыть» произойдет возврат в начальное окно вкладки «Настройки робота».

Также на вкладке будет отображен ранее созданный контракт с возможностью его редактирования либо удаления.

Для этого необходимо выбрать нужный знак из выпадающего списка:



Больше одного контракта добавить нельзя.

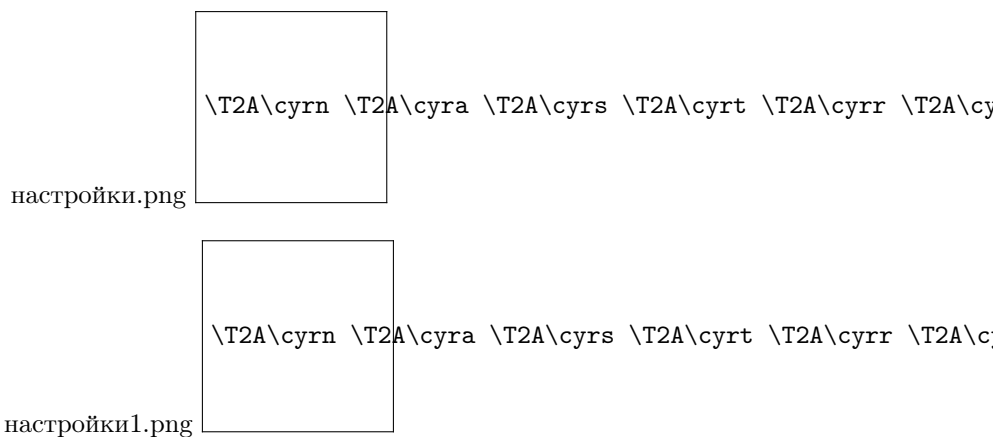
## Настройки программы

В разделе настройки есть возможность настройки языка, внешнего вида платформы, автосохранения, истории действий, вида условий/ветвлений, настройки хранилищ исходных и готовых роботов, а также настройки сертификатов

Экранная форма настройки разделяется на 4 вкладки: основные, хранилище исходных роботов, хранилище готовых роботов, сертификаты. Все сделанные изменения сохраняются автоматически.

Для создания нового хранилища необходимо указать название, URL-адрес, логин и пароль.

На вкладке «Сертификаты» имеется возможность настройки сертификатов.



## 2.6 Панель действий

Панель действий располагается слева в интерфейсе студии.

Панель состоит из двух вкладок: «Действия» и «Ресурсы».

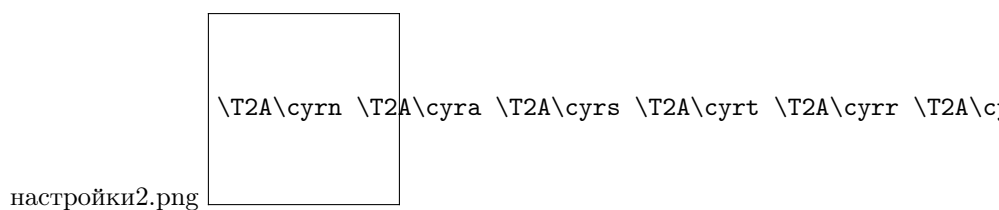
Разделы вкладки:

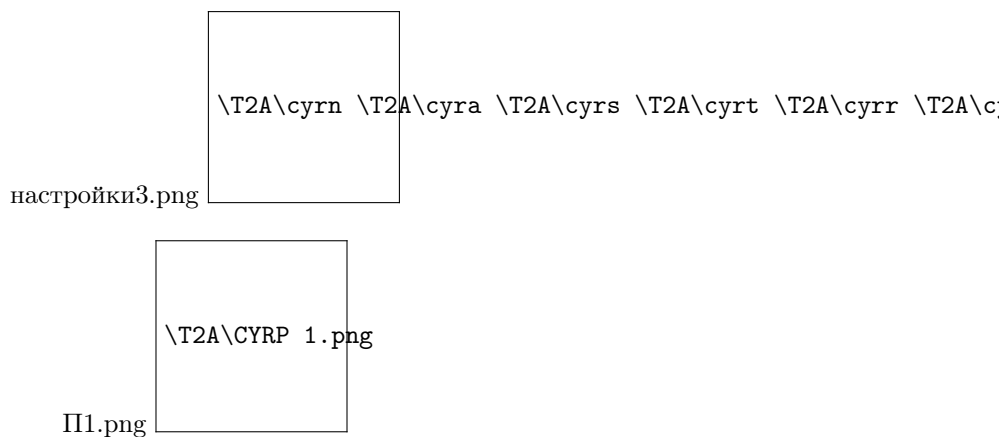
- «Переменные»
- «Файлы»
- «Веб-элементы»
- «Win-элементы»
- «Отслеживаемые переменные»

Переключение между разделами осуществляется внизу вкладки.

В данной вкладке находятся все группы действий студии:

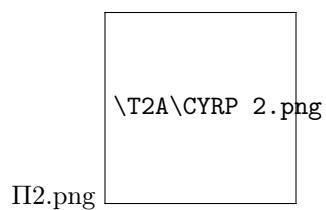
- «Базовые действия»
- «Коллекции»
- «Конвертация дат»
- «Базы данных»
- «Диалоговые окна»
- «Почта»
- «Excel»
- «Операторы»

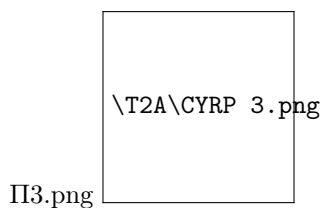




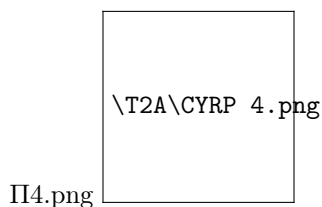
- «Файловая система»
- «FTP»
- «Интеграция»
- «ОС»
- «OUTLOOK»
- «PDF»
- «Процессы»
- «Безопасность»
- «Браузеры»
- «Имитация устройств»
- «GOOGLE SHEETS»
- «Таблицы»
- «Текст»
- «Распознавание текста»
- «Машинное зрение»
- «Браузеры (1.3)»
- «WINDOWS»
- «Word»
- «XML»

Более подробно с группами существующих в ROBIN Studio действий и порядком работы с ними можно ознакомиться в разделе Действия. Все действия собраны в группы и подгруппы, что облегчает процесс нахождения нужного действия для схемы робота.

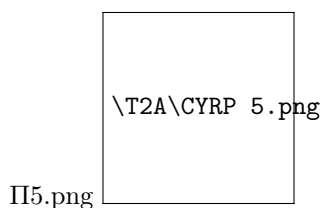




Для поиска нужного действия существует поисковая строка, располагается поисковая строка, расположенная в верхней части Панели действий.



Также для каждого действия имеется подсказка - если навести курсор мыши на действие в Панели действий, то на действии появится знак вопроса, щелкнув по которому, можно кратко узнать о том, для чего предназначено данное действие.



Если произвести клик левой кнопкой мыши по кнопке «Подробнее», то откроется браузер с подробной документацией по действию.

Информация по действию представлена в виде описания параметров.

## 2.7 Рабочее пространство студии

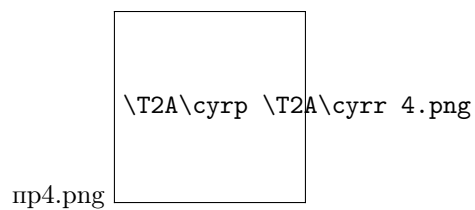
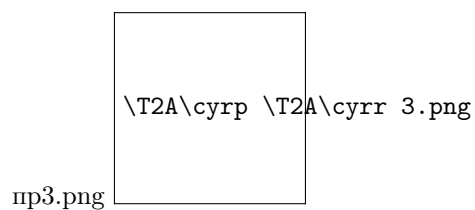
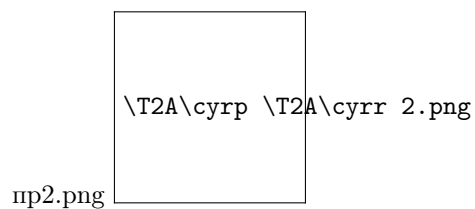
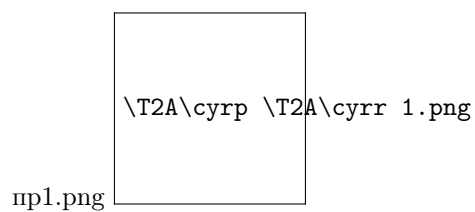
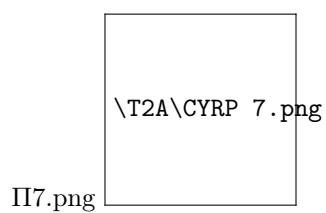
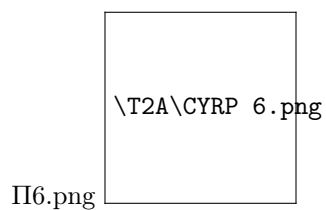
### 2.7.1 Размещение действий на рабочей панели

Для помещения действия в рабочее пространство ROBIN Studio необходимо:

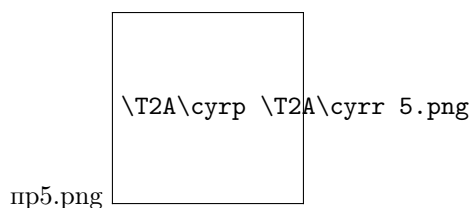
- Открыть вкладку «Действия» на панели действий слева;
- Кликнуть на необходимое действие и мышью перетащить его на рабочее пространство (щелкнуть по действию, в данном случае красный прямоугольник №1, и, зажав ЛКМ, перенести его в пространство между надписями «Начало» и «Конец», в данном случае красный прямоугольник №2, отпустить ЛКМ);
- Действие необходимо вставить между словами «Начало» и «Конец» в рабочем пространстве.

Для переноса действий по схеме работа необходимо:

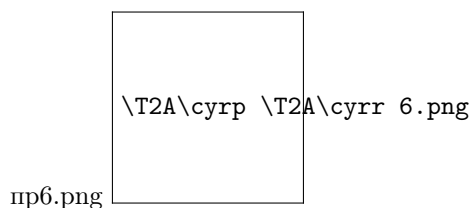
- Кликнуть на нужное действие в схеме (появится рамка зеленого цвета);



- Перетянуть в нужную часть схемы (зажав действие ЛКМ);

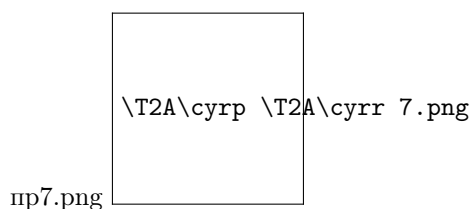


- Зеленая прямая показывает куда именно вставится действие (результат см. на следующем скриншоте);



### 2.7.2 Панель по управлению действиями

Данная панель будет отображена, если щелкнуть правой кнопкой мыши по действию в схеме робота:



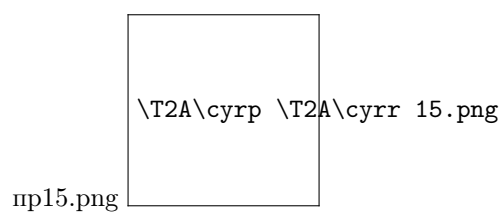
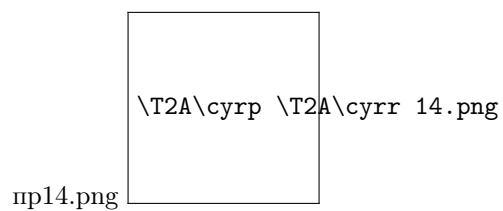
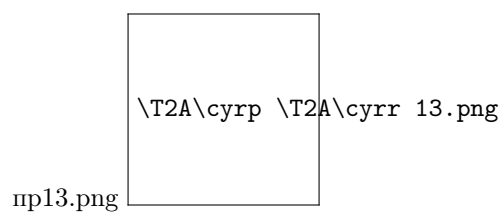
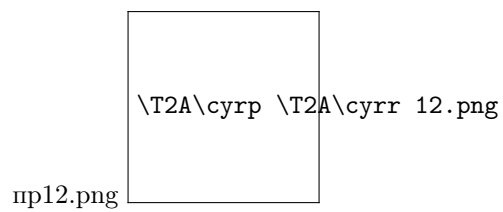
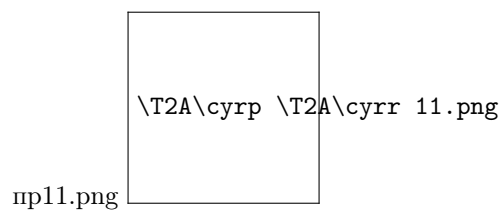
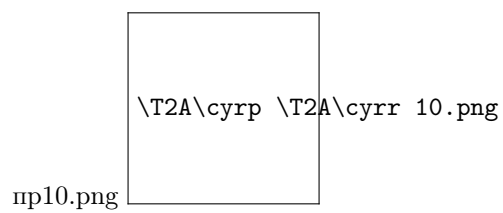
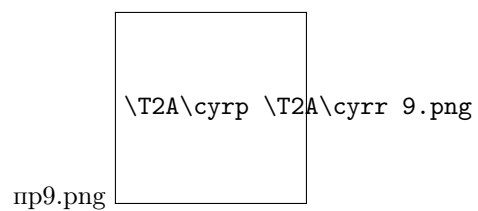
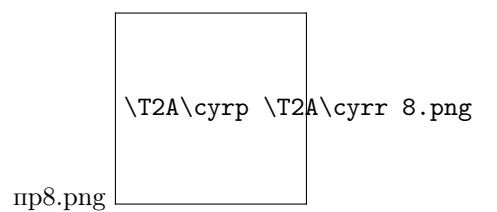
Рассмотрим её подробнее:

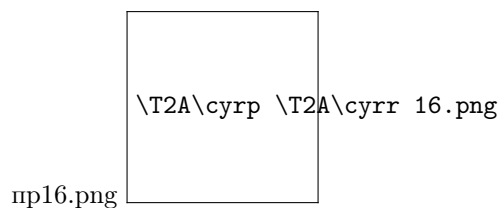
- Точка остановки - позволяет остановить выполнение схемы на данном действии и далее вручную последовательно выполнять следующие действия схемы (напротив действия находится красная точка):
- Выключить действие/Включить действие - позволяет активировать/деактивировать действие (действие окрашено в красный цвет):
- Оставить комментарий - позволяет оставить комментарий напротив действия:
- Удалить действие - удаляет действие из схемы.
- Объединить в группу - позволяет объединить действие в группу:
- Добавление в данную группу другого действия:
- Чтобы разгруппировать действие, необходимо нажать правой кнопкой мыши на надпись «Группа действий 1», выбрать необходимо действие, см. на скриншоте ниже:

Чтобы увеличить или уменьшить масштаб схемы, необходимо зажать клавишу Ctrl и прокрутить колесико мыши вперед или назад.

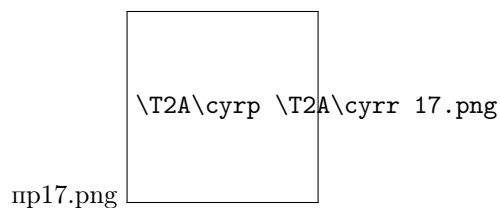
### 2.7.3 Панель навигации

При нажатии на кнопку «Открыть панель навигации» будет открыта панель с помощью которой можно управлять масштабом, расположением на рабочей области схемы робота.

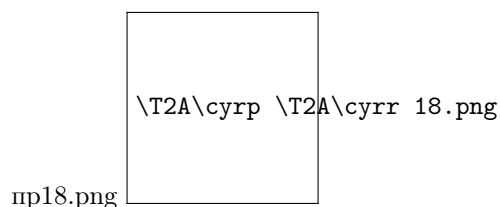




Панель навигации будет открыта в нижнем правом углу рабочей области.



На панели навигации находятся кнопки:



## 2.8 Панель логов

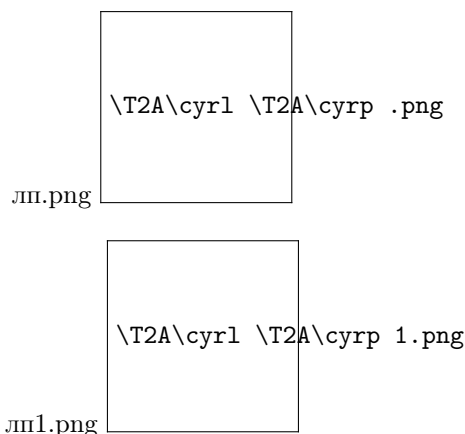
Панель логов находится в нижней части Robin Studio в разделе «Лог». В панели логов отображается подробный ход выполнения задач роботом. Панель логов выглядит следующим образом:

Лог включает в себя:

- Подготовка к запуску робота
- Время старта робота.
- Время начала выполнения каждой задачи.
- Время завершения выполнения каждой задачи.
- Информацию о возникших ошибках.
- Информацию о возникших предупреждениях.
- Время завершения работы робота.

Также в панели логов есть возможность просмотра значений, полученных из параметров действия до начала работы и просмотр результатов, полученных после отработки действия:





### 2.8.1 Поиск логов действий и копирование лога

В окне «Название или номер действия» осуществляется поиск необходимого действия.

Введенное наименование будет подсвечено зеленым цветом в логах, можно переходить по найденным наименованиям при помощи стрелок:

При нажатии на крайнюю правую иконку «Копировать лог» в поле лога, будет скопирован весь записанный лог работы робота.

### 2.8.2 Виды логов

Выводящиеся логи во вкладке «Лог» можно отфильтровать по следующим уровням:

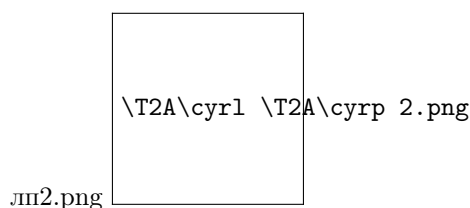
В таблице приведены уровни фильтрации логов.

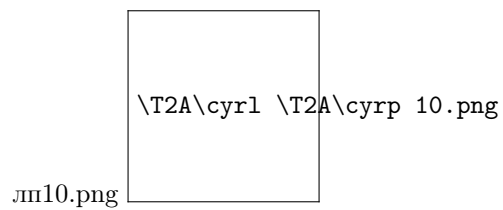
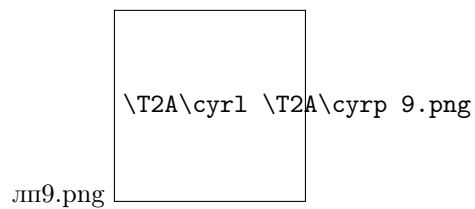
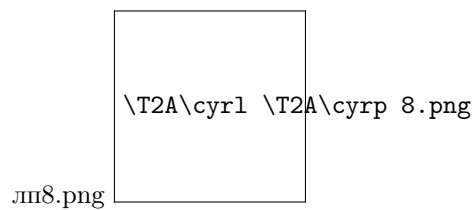
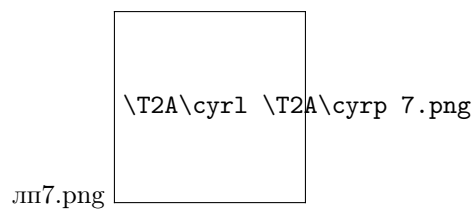
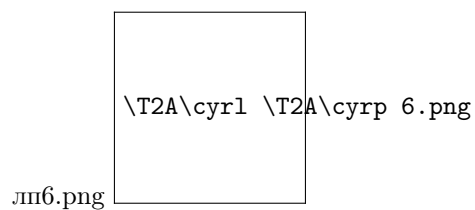
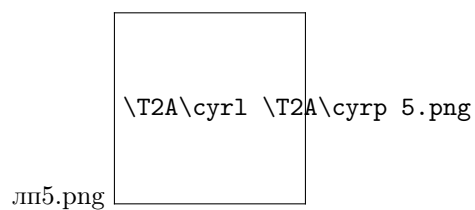
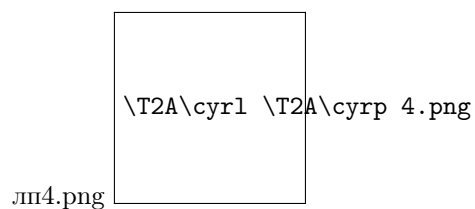
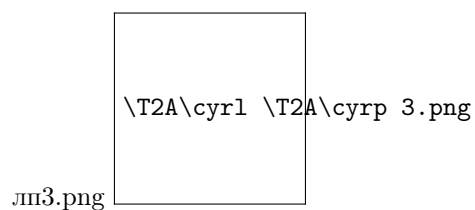
Также можно выставить уровень записи логов на панели управления, расположенной в верхней части студии по кнопке:

В таблице ниже приведены уровни записи логов:

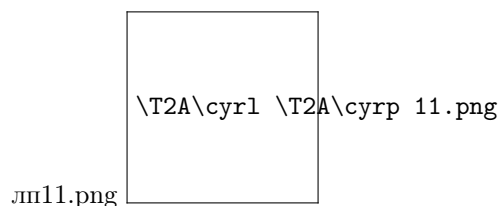
### 2.8.3 Открепление/Закрепление панели логов

Для открепления панели вкладки «Лог» необходимо нажать левой кнопкой мыши на требуемую вкладку и потянуть на рабочую область, вкладка будет откреплена и расположена на рабочей области.

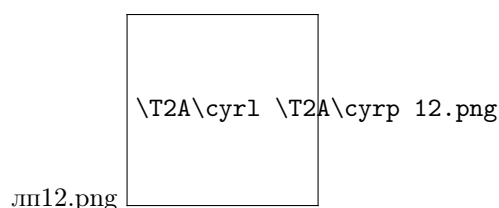




Далее вкладку «Лог» можно разместить с четырех сторон от рабочего поля со схемой. Для этого необходимо потянуть вкладку левой кнопкой мыши к выбранной стороне панели навигации, расположенной в центре:

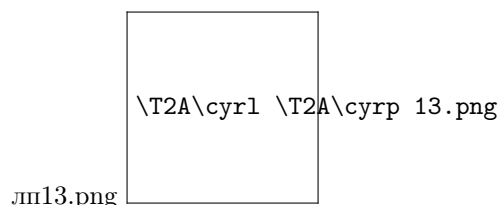


После появления синей области отпустить левую кнопку мыши. Вкладка «Лог» закрепится с выбранной стороны от рабочего поля со схемой:



Точно также можно разместить вкладку «Лог» по краям студии.

Для этого свободно плавающую вкладку «Лог» необходимо потянуть левой кнопкой мыши к выбранной стороне панели навигации по краям студии:



После появления синей области отпустить левую кнопку мыши. Вкладка «Лог» закрепится с выбранного края студии:

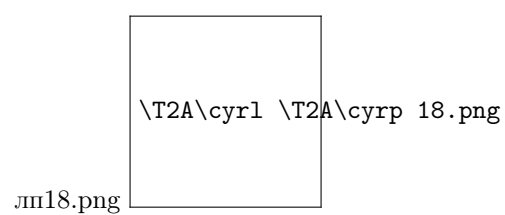
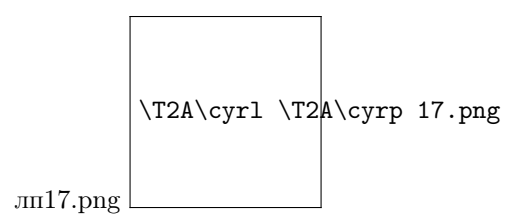
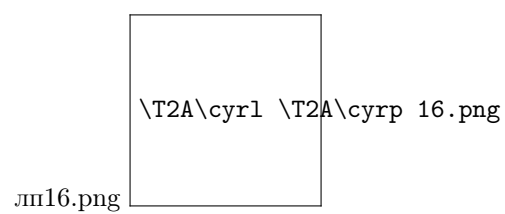
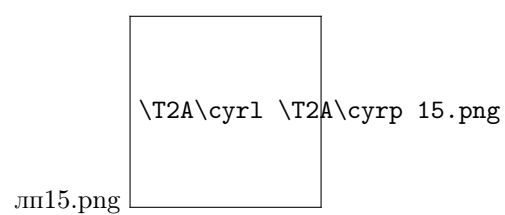
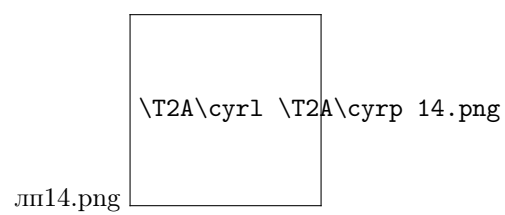
Вкладка «Лог» так же может быть закреплена рядом с вкладкой «Схема робота».

Для этого необходимо нажать правой кнопкой мыши на плавающую вкладку «Лог» и выбрать значение «Tabbed document».

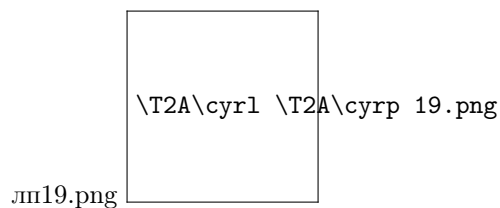
Для возврата окна с логами необходимо нажать правой кнопкой мыши на соответствующую вкладку и выбрать значение «Плавающая область».

Вкладка «Лог» так же может быть закреплена рядом с любой вкладкой в Robin Studio.

Для этого необходимо свободно плавающую вкладку «Лог» потянуть левой кнопкой мыши к выбранной вкладке до появления синего прямоугольника:



А затем отпустить кнопку мыши. Вкладка «Лог» закрепится около нужной вкладки в Robin Studio.



## 2.9 Свойства действий

При клике на любое действие, которое находится на рабочей области, в правой части экрана будет открыта панель «Свойства действия».

Данная панель разделена на три блока: информация, параметры и результаты.

Свойства действия

ИНФОРМАЦИЯ ^

Уровень логирования	Ошибка	▼
* Версия действия	1.3.1 net	▼
* Название действия	Открыть 1	
Задержка старта, мс	0	
Описание		

ПАРАМЕТРЫ ^

* Путь до файла		...
Видимый	<input type="checkbox"/>	...

РЕЗУЛЬТАТЫ ^

Контекст		
----------	--	--

1. Блок «Информация» - это общие свойства для всех действий. Блок будет неизменным независимо от выбранного действия.

№	Название поля	Характеристика
1	Уровень логирования	Доступные значения: ошибка, предупреждение, информация, отладка, нет
2	Версия действия	Каждое действие имеет свою версию
3	Название действия	Поле, в которое можно указать собственный вариант названия действия. Заполнено по умолчанию стандартным названием.
4	Задержка старта, мс	Время, которое Программный робот будет ожидать до выполнения действия. Например, если в поле указано значение «1000», то Программный робот будет ожидать выполнения 1 секунду.
5	Описание	Произвольный комментарий к действию.

2. Блок «Параметры» - это те параметры с которым запускается действие.

В данном блоке около каждого параметра расположена кнопка с тремя точками.



При нажатии на данную кнопку отобразятся дополнительные кнопки для заполнения данных параметра ( в зависимости от параметра):



При помощи данной кнопки происходит открытие всех ресурсов, которые есть в данном роботе и подходящие по типу



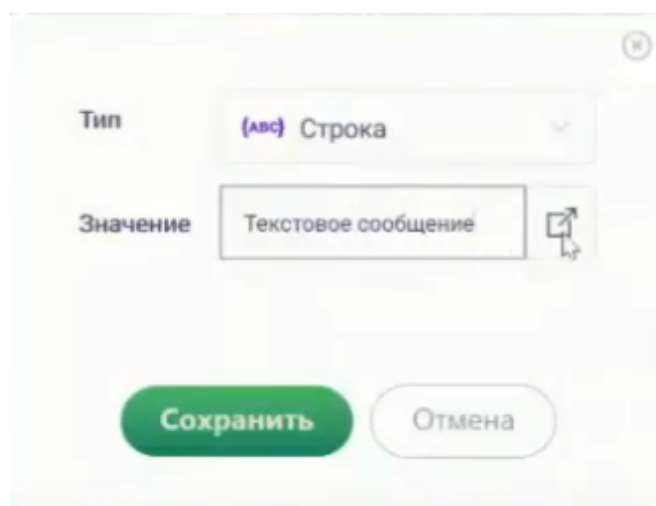
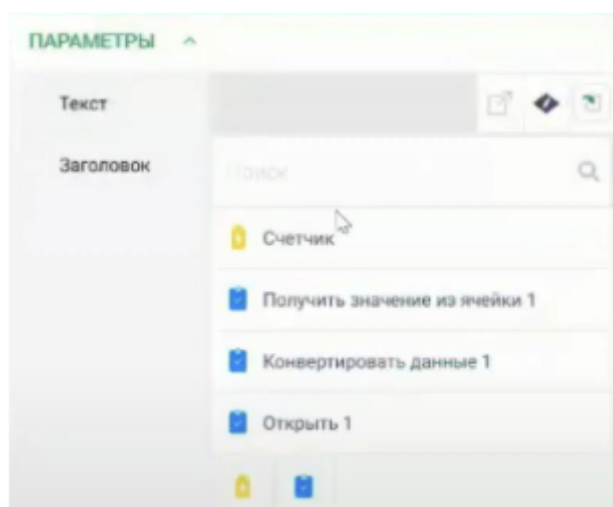
Будет открыт выпадающий список:

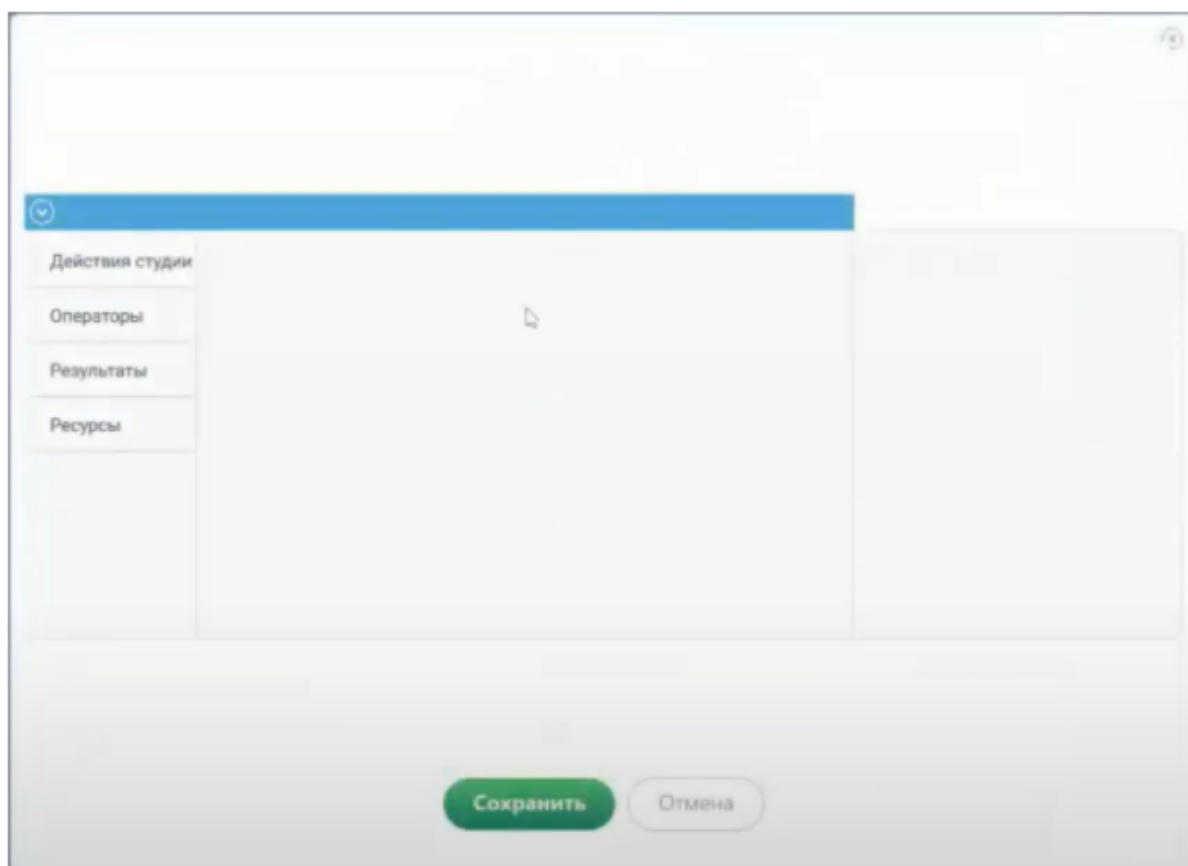
Кнопка редактора:

При нажатии на кнопку будет отображено окно для указания наименования параметра.

Кнопка конструктора выражений:

При нажатии на кнопку конструктора выражений будет открыта форма с вкладками: действия студии, операторы, результаты, ресурсы.





Возможность выбрать ресурсы, которые находятся в работе, путем перехода на вкладку «Ресурсы»:

Возможность выбора результатов на вкладке «Результаты»:

Возможность выбора действий на вкладке «Действия»:

Более подробно работа с конструктором выражений расписана в разделе Конструктор выражений.

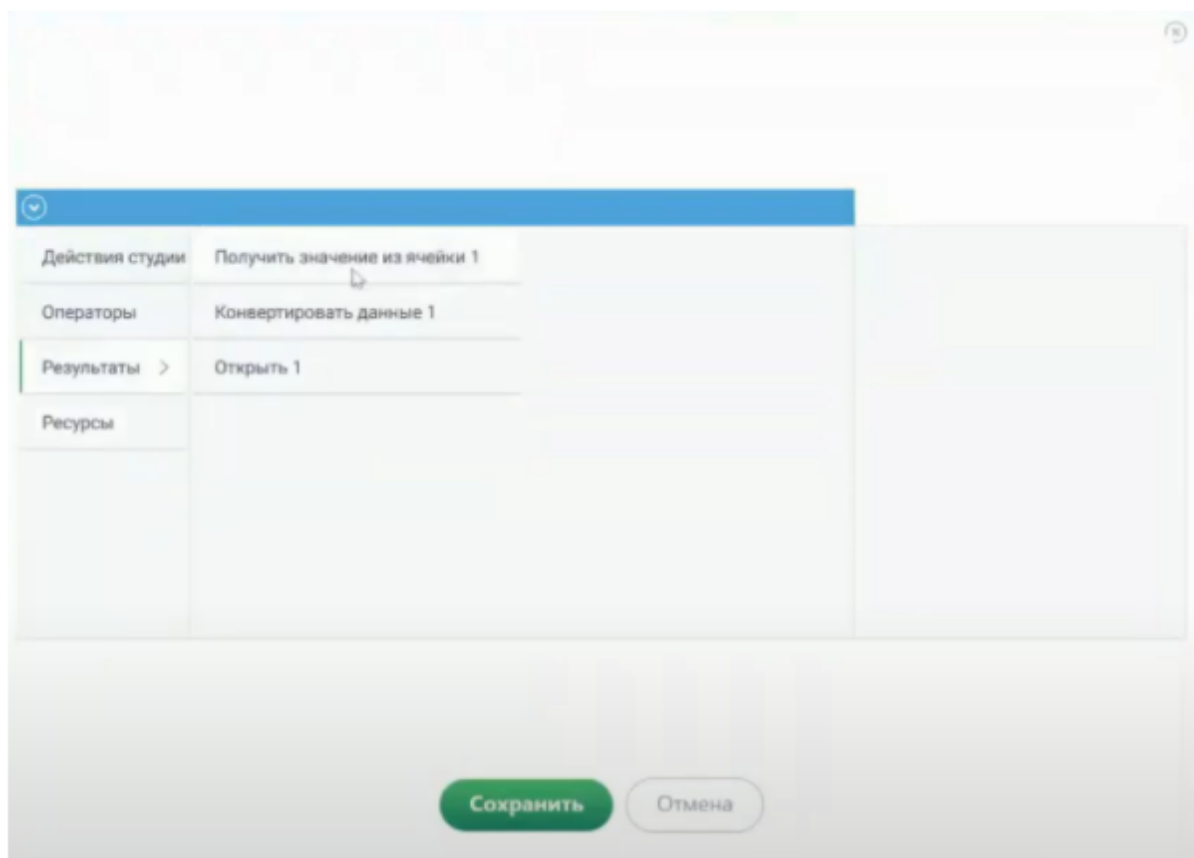
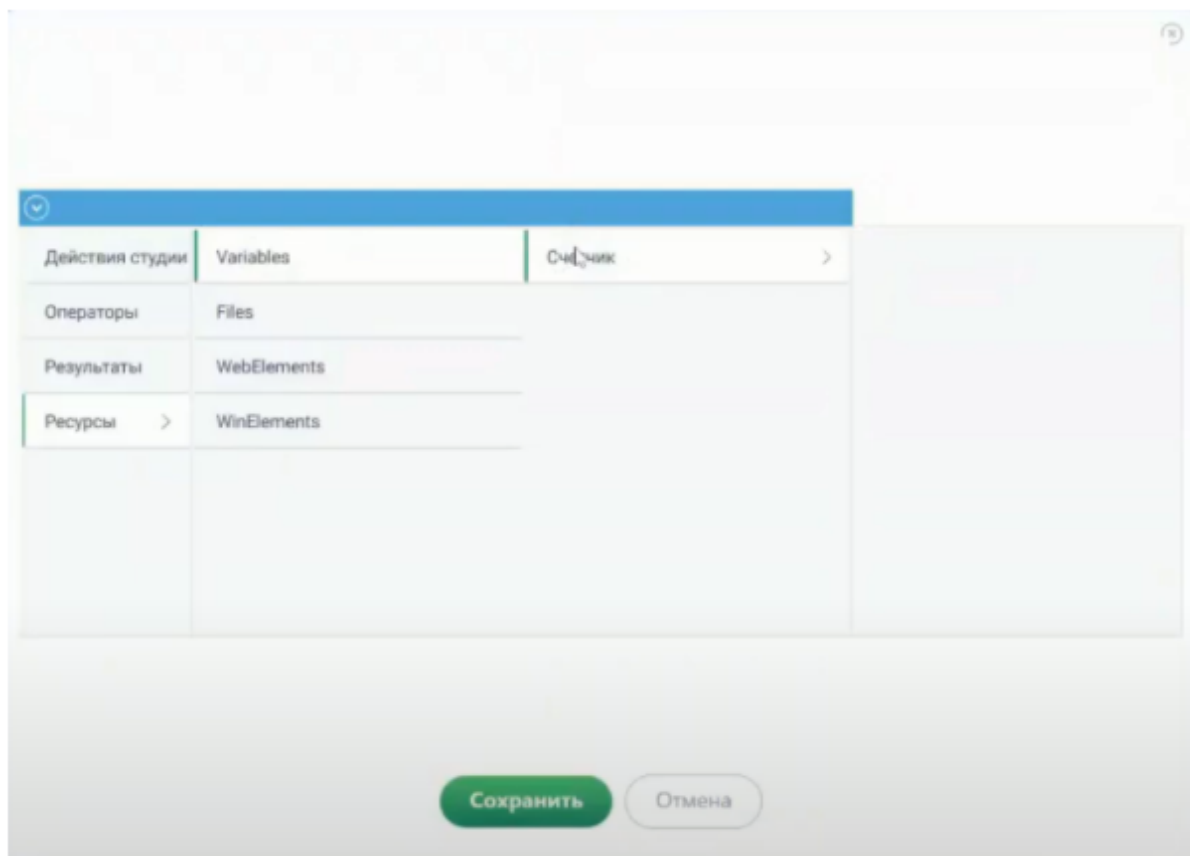
3. Блок «Результаты»- это результат выполнения действия.

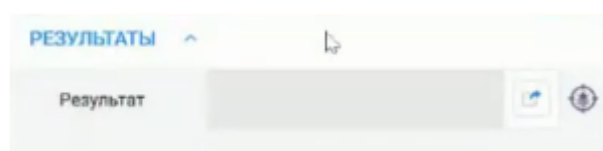
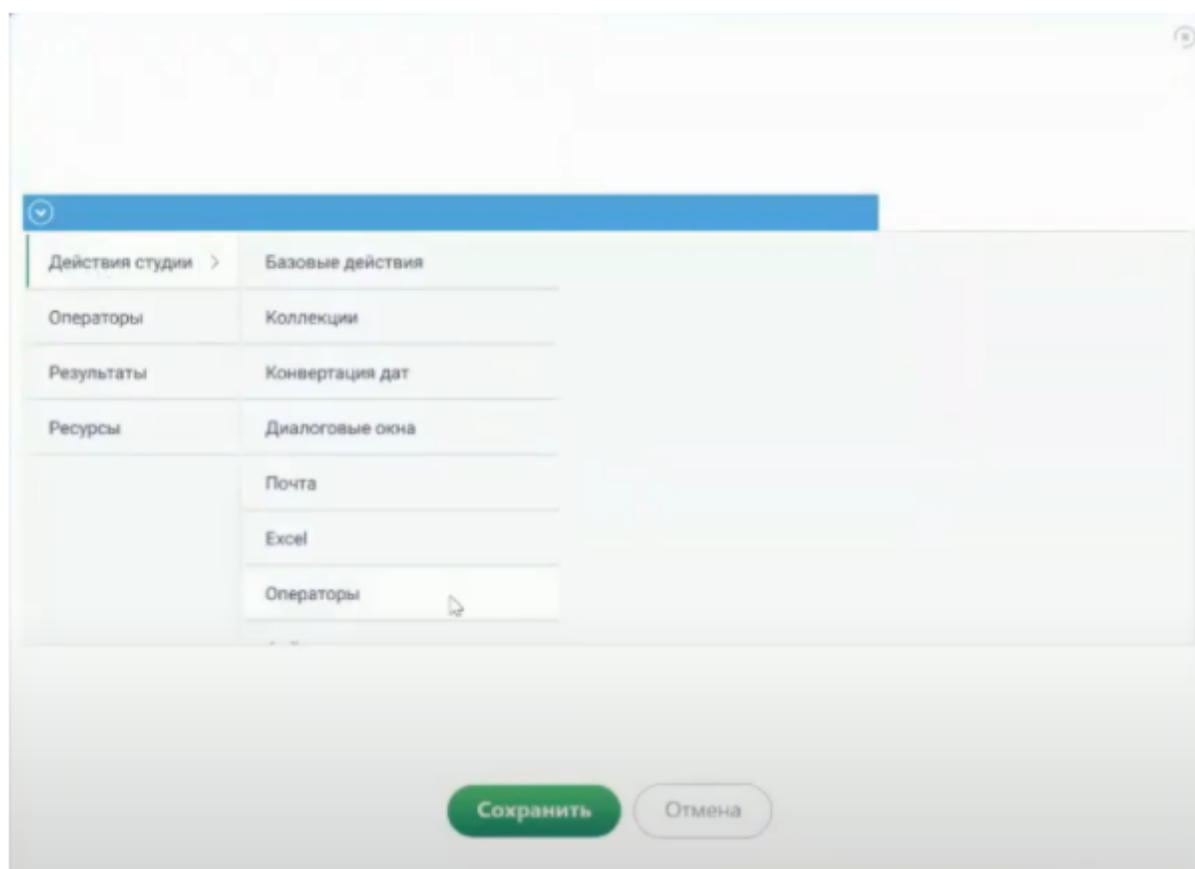
Поле, названное по-умолчанию «Результат» может быть переименовано.

В данном блоке есть кнопки:

- Поиск переменной
- Указание переменной, как «отслеживаемая»







## 2.10 Панель ресурсов

В ROBIN Studio 2.0 есть следующие виды ресурсов, используемые роботом в ходе выполнения процесса:

- Переменные
- Файлы
- Win-элементы
- Web-элементы
- Отслеживаемые переменные

Для создания или изменения ресурсов, необходимо открыть на левой панели вкладку «Ресурсы» и выбрать внизу на кнопку необходимого вида ресурса:

### 2.10.1 Переменные

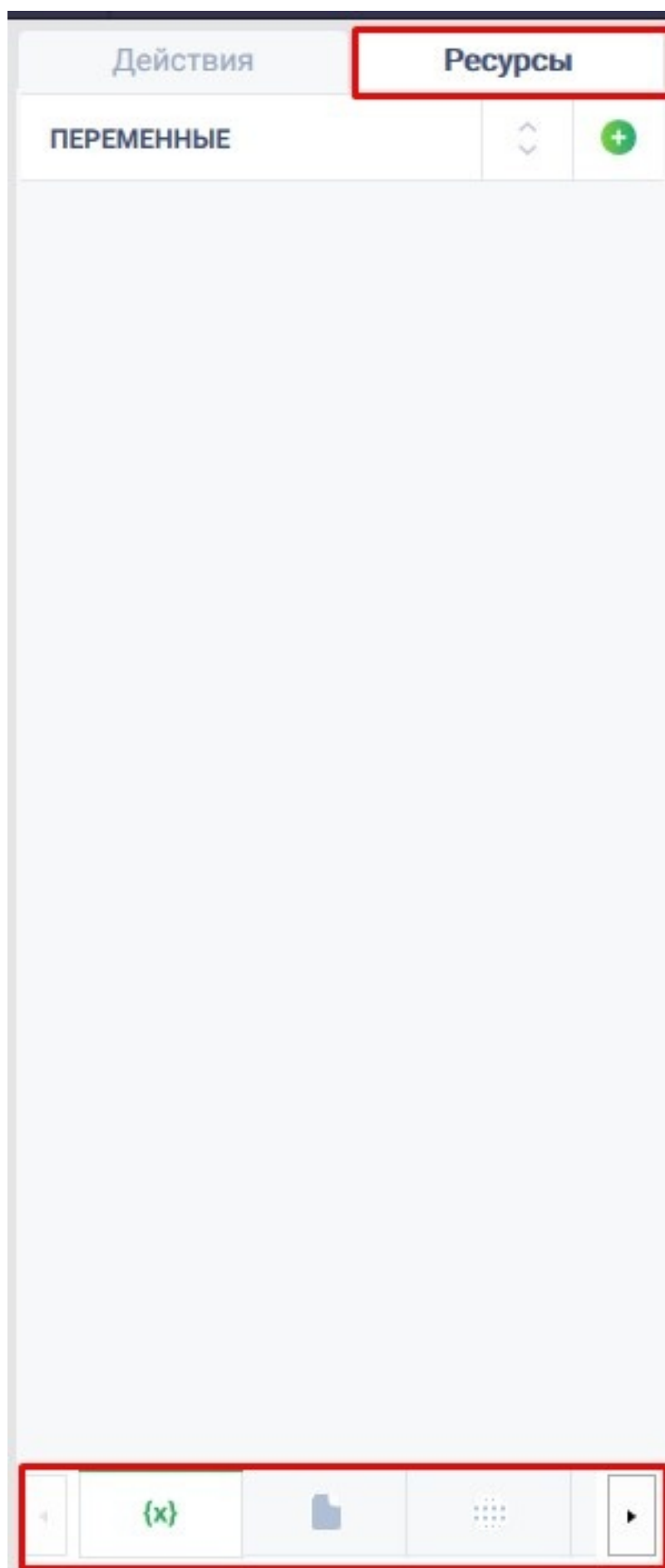
#### Создание переменных

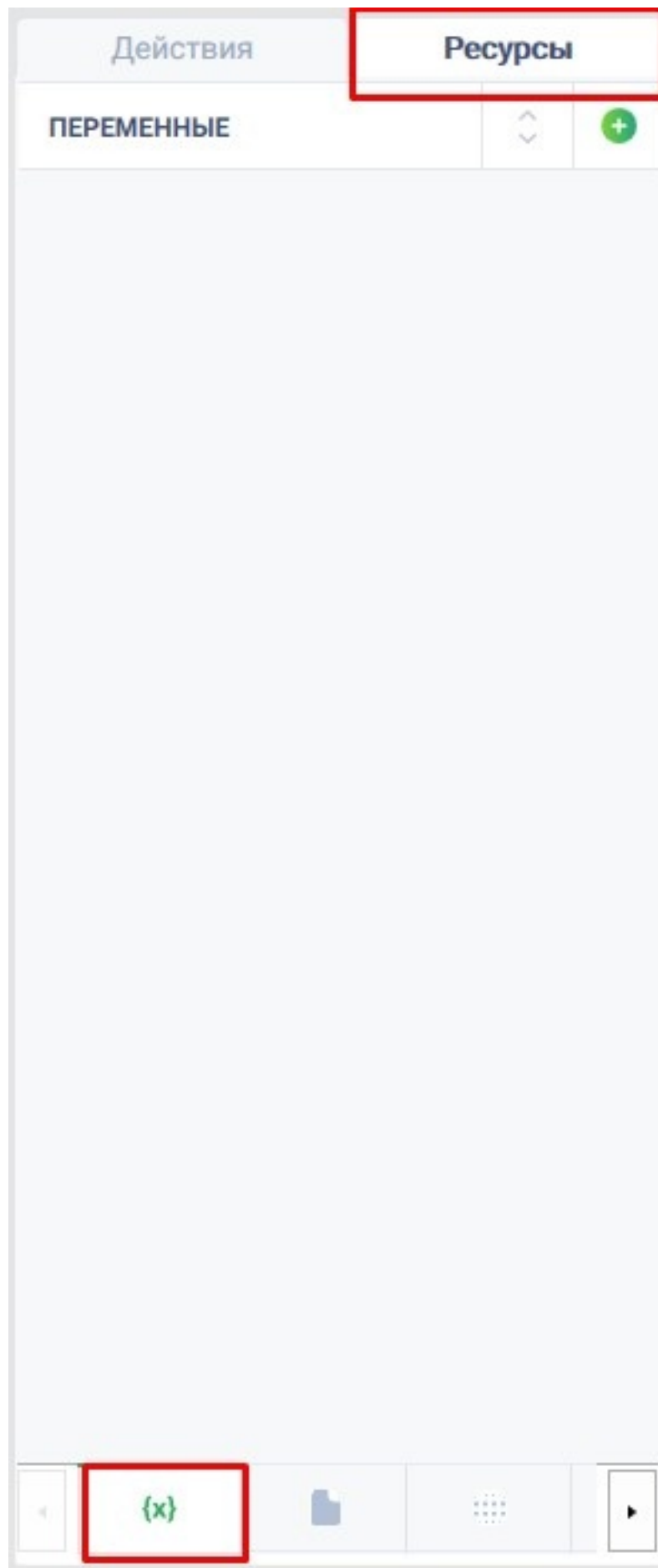
Все исходные данные, а также те, которые Программный робот получает и обрабатывает в ходе выполнения процесса, записываются и хранятся в разделе «Переменные». В ROBIN Studio 2.0 для создания и просмотра всех переменных необходимо на левой панели открыть вкладку «Ресурсы» и нажать на кнопку «Переменные».

Для создания новой переменной необходимо нажать на кнопку плюса:

На левой панели откроется форма «Новая переменная», на которой можно установить параметры создаваемой переменной.

Параметры переменных:







Параметр	Назначение	Тип	Внешний вид
Имя переменной	Для ввода названия создаваемой переменной.	Поле	<div>Имя переменной</div> <input type="text"/>
Тип переменной	Выбор типа переменной.	Выпадающий список	<div>Тип переменной</div> <input type="text"/>
Значение	Для ввода начального значения переменной, если это необходимо.	Поле/чекбокс	Зависит от выбранного типа переменной. Подробнее в разделе Типы переменных.
Отслеживать при debug	При активации этого атрибута значения переменной будут отображаться в логе Программного робота.	Чекбокс	<input type="checkbox"/> Отслеживать при debug
Входной параметр робота	При активации этого атрибута, параметры (значения), поступающие в Программного робота при его запуске, будут записаны в эту переменную.	Чекбокс	<input type="checkbox"/> Входной параметр робота
Результат работы робота	При активации этого атрибута последнее значение переменной по выполнению сценария будет записано в отдельный xml-файл в виде схемы.	Чекбокс	<input type="checkbox"/> Результат работы робота
Комментарий	Для ввода комментария к создаваемой переменной.	Поле	<div>Комментарий</div> <input type="text"/>

При нажатии на кнопку «Отмена» или на знак крестика произойдет отмена введенных параметров и возврат к общему списку переменных. При нажатии на кнопку «Создать» будет создана новая переменная, которая будет отображена в списке переменных.

### Сортировка переменных

Для того, чтобы выполнить сортировку переменных необходимо нажать на кнопку сортировки, далее можно сделать сортировку по порядку добавления, проставив чекбокс около нужного значения (новые/старые) и сортировку по имени (А-Я /Я-А). Для сохранения внесенных изменений следует нажать на кнопку «Применить», для отмены и переходу к списку переменных на кнопку «Отмена».

Действия

Ресурсы

ПЕРЕМЕННЫЕ

⌵


✕


НОВАЯ ПЕРЕМЕННАЯ


Имя переменной

Тип переменной

Значение

☐  Отслеживать при debug

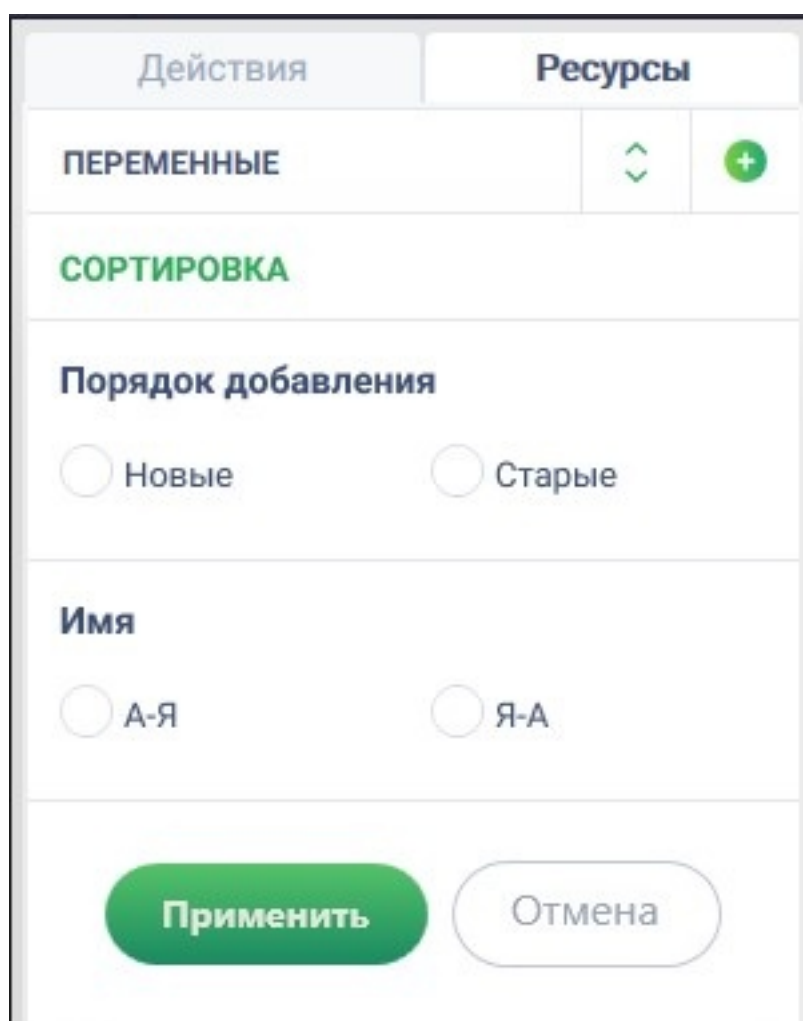
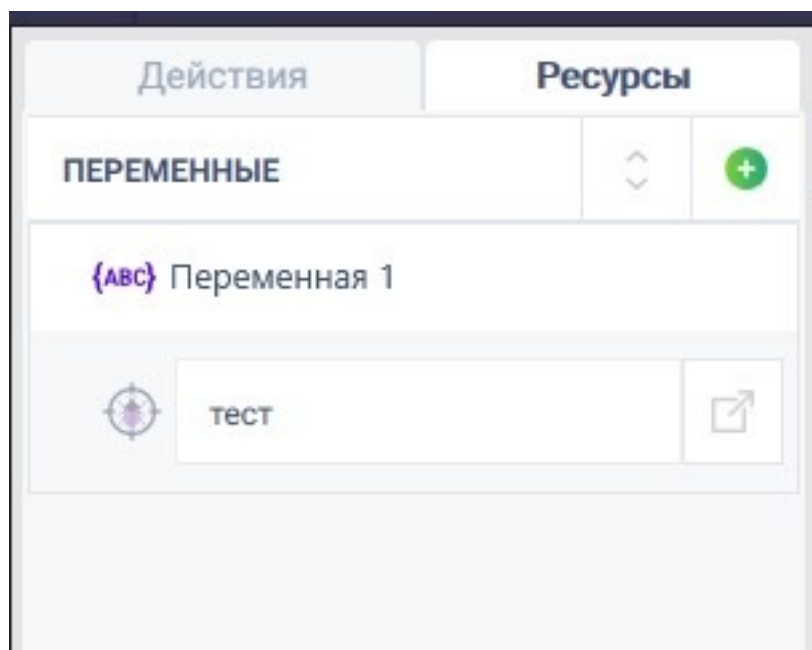
☐  Входной параметр робота

☐  Результат работы робота

Комментарий

Создать

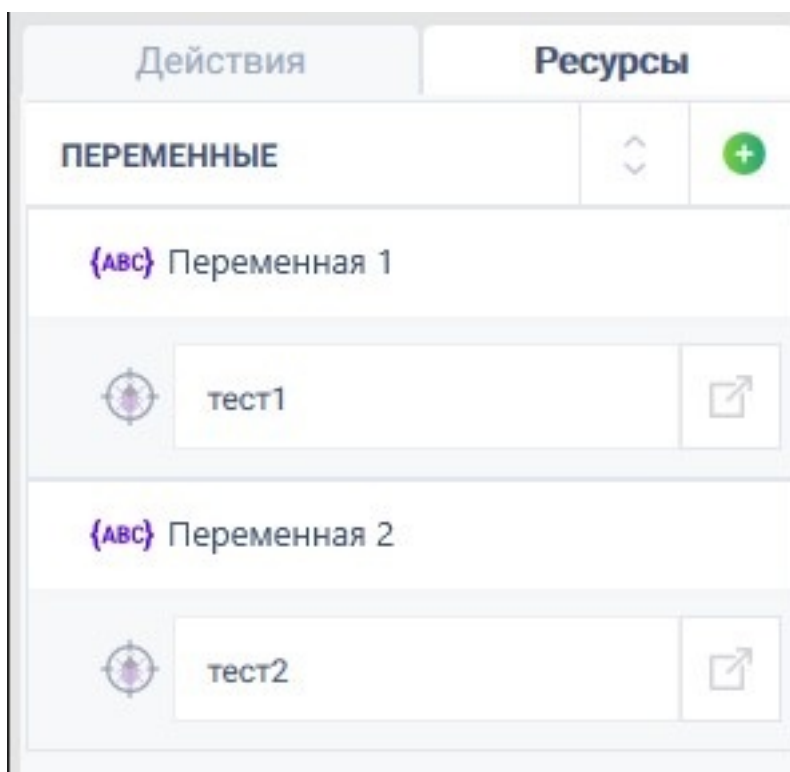
Отмена









## Редактирование переменных

Для редактирования уже созданных переменных необходимо перейти на общий список всех переменных (отображается на левой панели на вкладке «Ресурсы»).



У всех созданных переменных, которые отображаются в общем списке, есть:

Параметр	Назначение	Тип	Внешний вид
Имя	Название переменной. Можно изменить только при нажатии кнопки редактирования.	Поле	<div>Имя переменной</div> <div>Переменная 1</div>
Тип переменной	Графическое изображение типа переменной.	Изображение	Зависит от выбранного типа переменной
Отслеживать при debug	При активации кнопки значения переменной будут отображаться в логе Программного робота.	Кнопка	<p>1. Параметр не активирован</p>  <p>2. Параметр активирован</p> 
Значение	Для отображения и редактирования начального значения переменной.	Поле/чекбокс/кнопка	Внешний вид зависит от выбранного типа переменной. Подробнее в разделе Типы переменных.
Редактировать	Кнопка отображается только при наведении на конкретную переменную. По нажатии откроется форма «Редактирование», содержащая все параметры, которые были доступны при создании переменной.	Кнопка	
Удалить	Кнопка отображается только при наведении на конкретную переменную. По нажатии выбранная переменная будет удалена.	Кнопка	

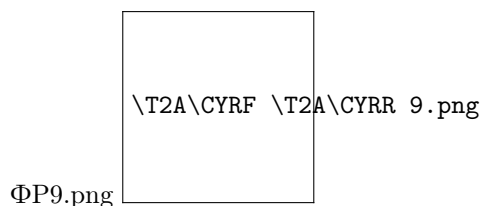
## Типы переменных

Ниже рассмотрены доступные типы переменных:

### 1. Горячие клавиши

В переменную данного типа можно записать нажатие на какую-либо клавишу или комбинацию клавиш клавиатуры. Пример заполнения полей при создании переменной:

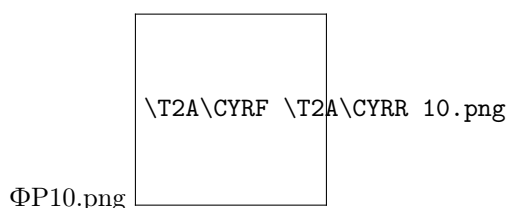
Данную переменную можно использовать для заполнения параметров действий «Ожидание нажатия горячих клавиш» и «Нажатие клавиш» из группы действий «Имитация устройств» или



«Отправить нажатие горячих клавиш» из группы действий «Браузеры».

## 2. Дата-время

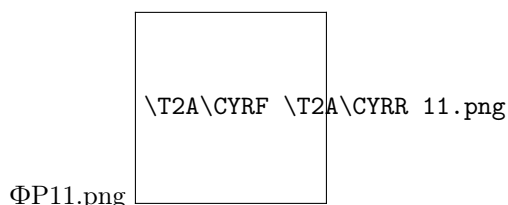
В переменную этого типа записывается дата и время в формате: dd-mm-yyyy hh:mm:ss. При создании переменной параметр «Значение» заполняется путем выбора даты и времени из открывшегося окна. Пример:



Переменная данного типа может пригодиться для работы с действиями группы «Конвертация дат».

## 3. Коллекция

В переменную типа «Коллекция» можно записать набор значений одного или нескольких типов. Элементом коллекции так же может быть коллекция. При создании коллекции для каждого ее значения можно выбрать тип. Пример:



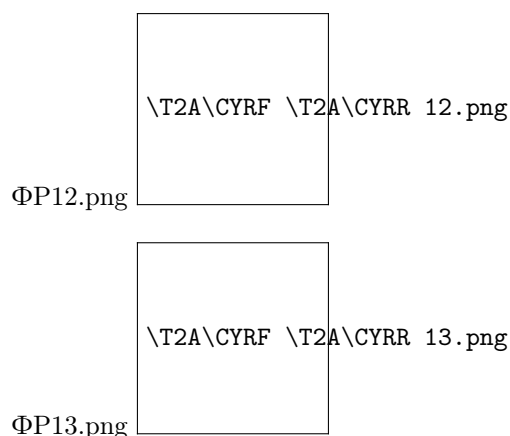
Использование данной переменной возможно с действиями многих групп действий. Например, при работе с циклом «Для каждого» в параметры необходимо записать коллекцию, элементы которой последовательно обработаются в цикле. Пример:

## 4. Координаты точки

Переменная данного типа содержит координаты X и Y какой-либо точки на экране. При создании переменной можно заполнить значения координат X и Y вручную, либо автоматически посредством нажатия кнопки «PickArea» в открывающемся окне. Пример:

Переменная такого типа может пригодиться для работы с действиями группы «Имитация устройств» или в переменную такого типа можно записать результат действия «Получить координаты элемента» из группы действий «Windows».

## 5. Логический



Логическая переменная может содержать только два значения: True и False. Заполненный чек-бокс при создании переменной означает, что переменная будет содержать значение истины (True), в противном случае переменная примет значение «ложь» (False). Пример:

Использовать переменную с типом «Логический» можно при заполнении параметров действий из различных групп действий в Robin Studio, в том числе и при заполнении параметра «Результат». Пример использования переменной: При заполнении параметра «Условие» действия «Если» из группы «Базовые действия» допустимо использование переменной логического типа.

## 6. Область

Тип данных для работы с размерами и координатами прямоугольной области. При создании переменной этого типа необходимо указать следующие параметры:

- X левого верхнего угла;
- Y левого верхнего угла;
- Высота;
- Ширина.

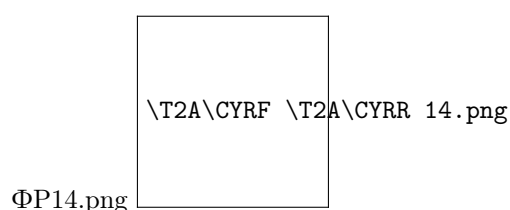
Заполнить координаты, ширину и высоту можно как в ручном режиме, так и в автоматическом режиме, посредством нажатия на кнопку «PickArea».

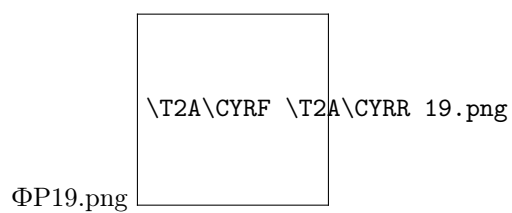
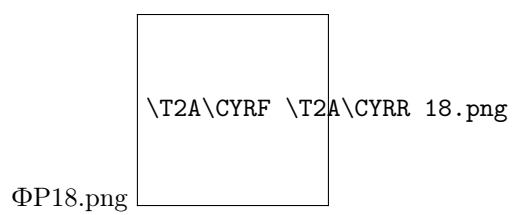
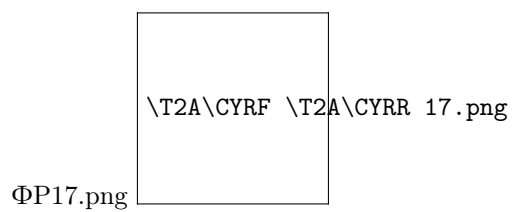
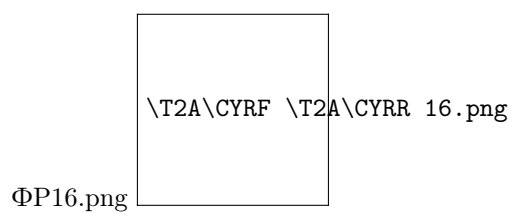
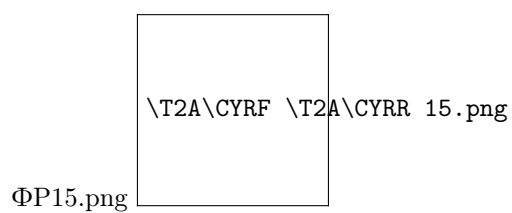
Пример:

Пример использования данной переменной: В переменную данного типа записывается результат действия «Получить размер и координаты элемента» из группы действий «Браузеры».

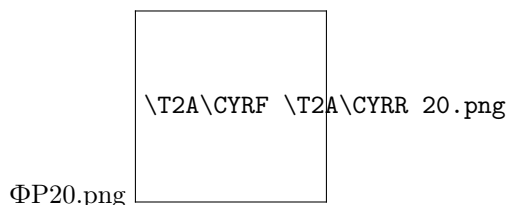
## 7. Объект

Объект может быть значением любого типа. При создании переменной, устанавливая значение, можно выбрать любой типа данных, который будет записан в переменную. Пример:



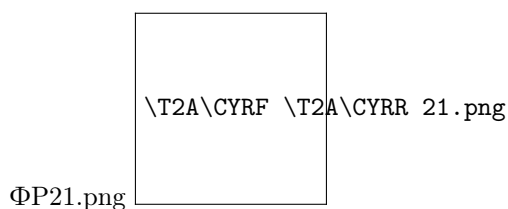


В переменную типа «Объект» можно записать результат действия другого типа. Пример: Результат действия «Сложение» типа «Число», но если записать результат в переменную типа «Объект», действие будет выполнено без ошибок.

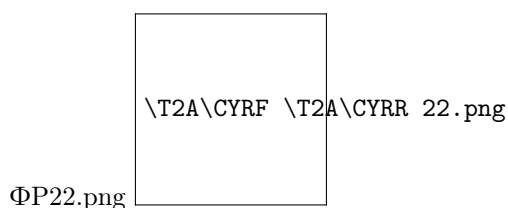


## 8. Пароль

Переменная этого типа содержит в себе информацию о пароле. После задания и сохранения переменной ее значение нельзя посмотреть, также оно не отображается в логе. Значение переменной этого типа также не отображается при создании:



Пример:



Пример использования данной переменной: При авторизации в системе роботу необходимо ввести пароль. Для этого, заполняя параметры действия «Отправить текст» из группы действий «Браузеры», в строке «Текст» указать ранее созданную переменную с типом данных «Пароль».

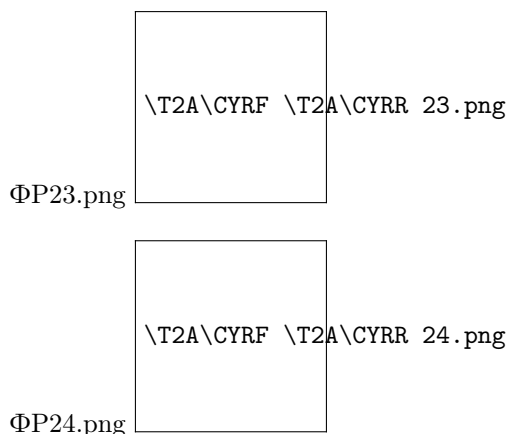
## 9. Путь к папке

Тип данных позволяющий выбрать путь к папке. При создании переменной с типом данных «Путь к папке» для заполнения значений открывается окно проводника, в котором можно выбрать папку, путь до которой будет сохранен в переменную. Пример:

Пример использования данной переменной: Переменную этого типа можно использовать для работы с действиями группы «Файловая система», например, при заполнении параметров действия «Получить содержимое папки». Пример:

## 10. Путь к файлу

Тип данных позволяющий выбрать путь к файлу. При создании переменной с типом данных «Путь к файлу» для заполнения значений открывается окно проводника, в котором можно выбрать файл, путь к которому будет сохранен в переменную.



Пример использования данной переменной: Переменную с типом данных «Путь к файлу» можно использовать для работы с действиями группы «Файловая система». Например, для заполнения параметров действия «Переместить файл». Результат так же может быть записан в переменную типа «Путь к файлу». Пример:

## 11. Словарь

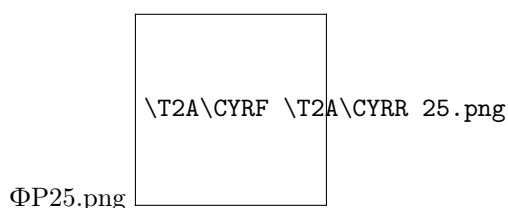
Переменная типа «Словарь» представляет собой массив данных, в отличие от коллекции, работающий с ключами и значениями, а не индексами. К каждому значению из словаря можно получить доступ по ключу. Значение может быть любого из предложенных в студии типов. При заполнении значения переменной возможно указать ключ, тип значения и само значение в появляющемся окне. Пример:

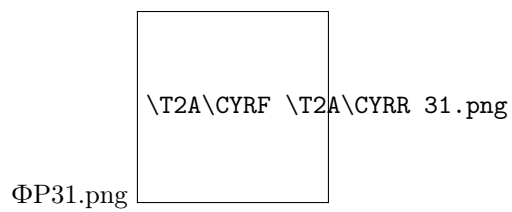
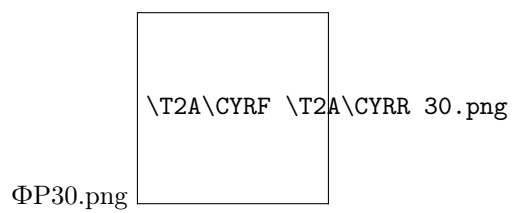
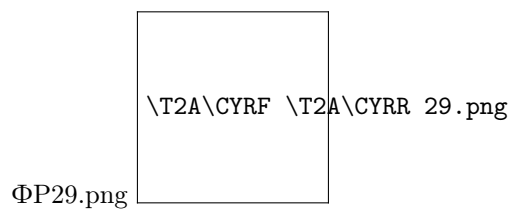
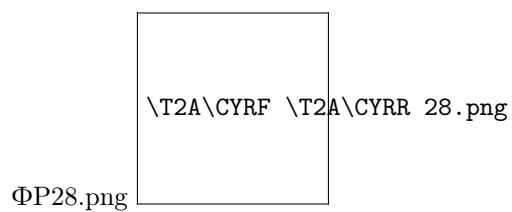
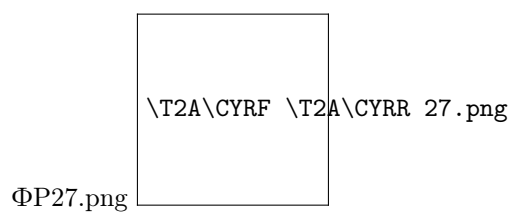
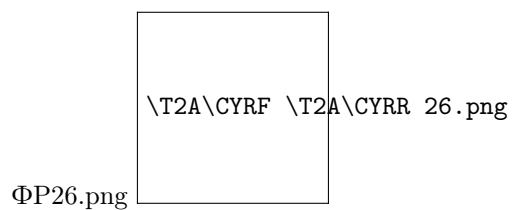
Использование данной переменной возможно с действиями многих групп действий, например, при заполнении параметров действия «Добавить запись» группы действия «Базы данных». Пример: Для заполнения параметра «Для вставки» использовать ранее созданную переменную с типом данных «Словарь».

## 12. Строка

Переменная с типом данных «Строка» содержит произвольную последовательность символьного алфавита. Переменные этого типа можно использовать с действиями почти каждой группы действий. При создании переменной с типом данных «Строка» в появляющемся окне можно вписать необходимый текст. Пример:

Пример использования переменной: Почти для каждого действия группы «Текст» требуется сохраненная в переменную или записанная вручную в соответствующие поля строковая информация. Пример: Для заполнения параметра «Исходный текст» действия «Проверить на вхождение» использовать созданную ранее переменную строкового типа.

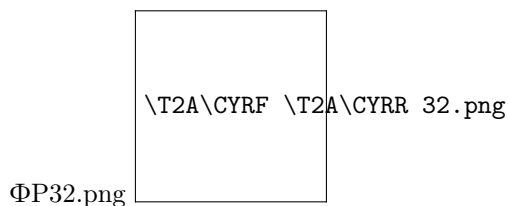




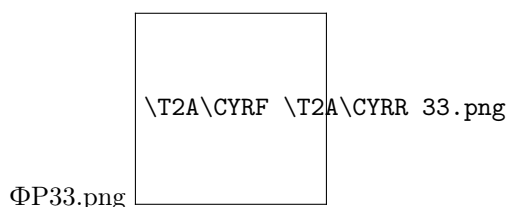


### 13. Таблица

Переменная этого типа предназначена для работы с данными в виде таблицы. Создание переменной:

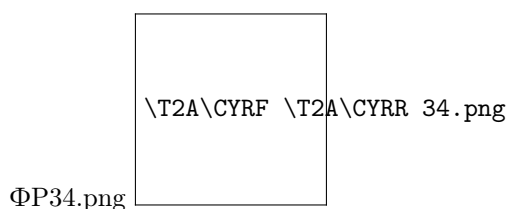


Пример применения переменной: Например, переменную данного типа можно использовать для работы с действиями группы действий «Таблицы» для заполнения параметров действия или в качестве результирующей переменной для действия «Получить сообщение» из группы действий «Почта».



### 14. Число

Переменная с типом данных «Число» хранит в себе числовые значения. При создании переменной необходимо ввести числовое значение в поле ввода значений. Пример:



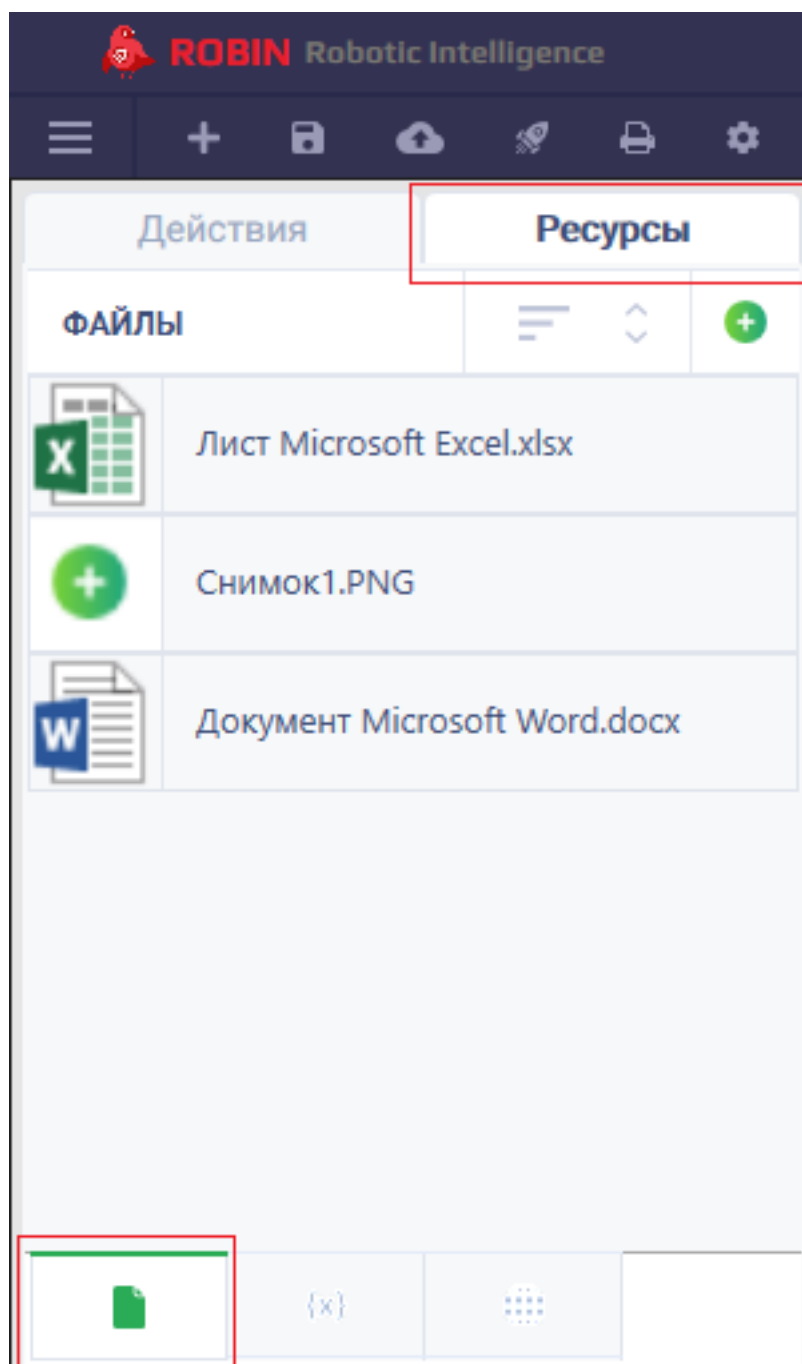
Пример использования переменной: Переменная типа число используется для действий различных групп, доступных в Robin Studio. Например, переменную численного типа можно использовать в качестве операнда и в качестве параметра «Результат» для действия «Сложение» группы действий «Операторы».

## 2.10.2 Файлы

Файлы в ROBIN Studio 2.0 – это внешние ресурсы, необходимые в работе некоторых действий, но которые можно хранить внутри робота. Однако, в отличие от переменных, файлы в ходе выполнения процесса не могут быть перезаписаны. Файлы, которые выступают в качестве ресурсов, могут содержать в себе различную служебную информацию и иметь любое расширение (.txt, .xlsx, .png, .dll и т.д.). Для создания и просмотра всех ресурсов файлового типа необходимо на левой панели открыть вкладку «Ресурсы» и нажать внизу на кнопку «Файлы».

\T2A\CYRF \T2A\CYRR 35.png

ФР35.png



При необходимости все файлы можно отсортировать по их названию (от А до Я или наоборот), нажав на кнопку:

Или по их расширению, нажав на кнопку:



Для отмены определенного вида сортировки нужно снова нажать на соответствующую кнопку.

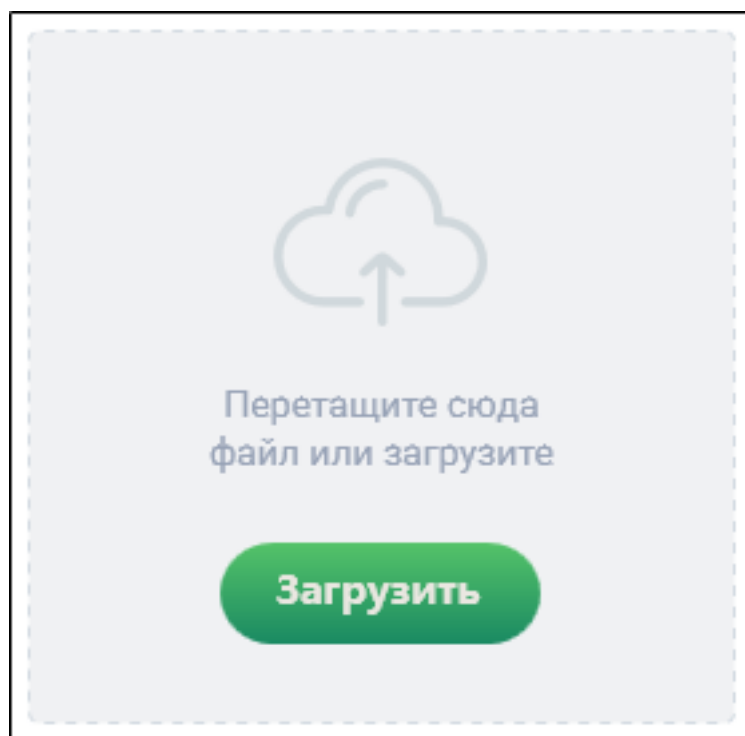
### Добавление файлов

Чтобы добавить новый файл, нужно нажать на кнопку плюса. На левой панели появится форма «Новый файл», на которой необходимо:

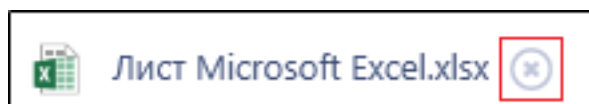
1. Заполнить поле «Название» именем создаваемого ресурса.
2. Нажать на кнопку «Загрузить» и выбрать в открывшемся окне Проводника нужный файл.

ИЛИ

Перетащить нужный файл в область загрузки:



При необходимости добавленный файл можно поменять на данном этапе, удалив старый нажатием на кнопку рядом с появившимся названием файла:



3. Нажать на кнопку «Создать».

Если нужно выйти с формы добавления файла без сохранения внесенных изменений, нажмите на кнопку «Отмена» или значок крестика вверх формы:



### Редактирование добавленных файлов

Все созданные ресурсы файлового типа, которые отображаются в общем списке на левой панели, имеют кнопки (появляются при наведении на конкретный ресурс):

1. При нажатии выбранный ресурс будет удален:



2. При нажатии откроется форма «Редактирование», на которой можно поменять имя ресурса или выбрать другой внешний файл:



3. При нажатии откроется окно Проводника, в котором можно выбрать директорию в системе компьютера, где будет сохранена копия файла из выбранного ресурса:

## 2.10.3 Win-элементы

Win-элементы – это ресурсы, в которых записаны уникальные идентификаторы, определяющие конкретный элемент (кнопка/поле/т.п) в окне Windows. Действия из группы «Windows» используют параметры из win-элемента для обнаружения этого элемента в окне и дальнейшей работы с ним.

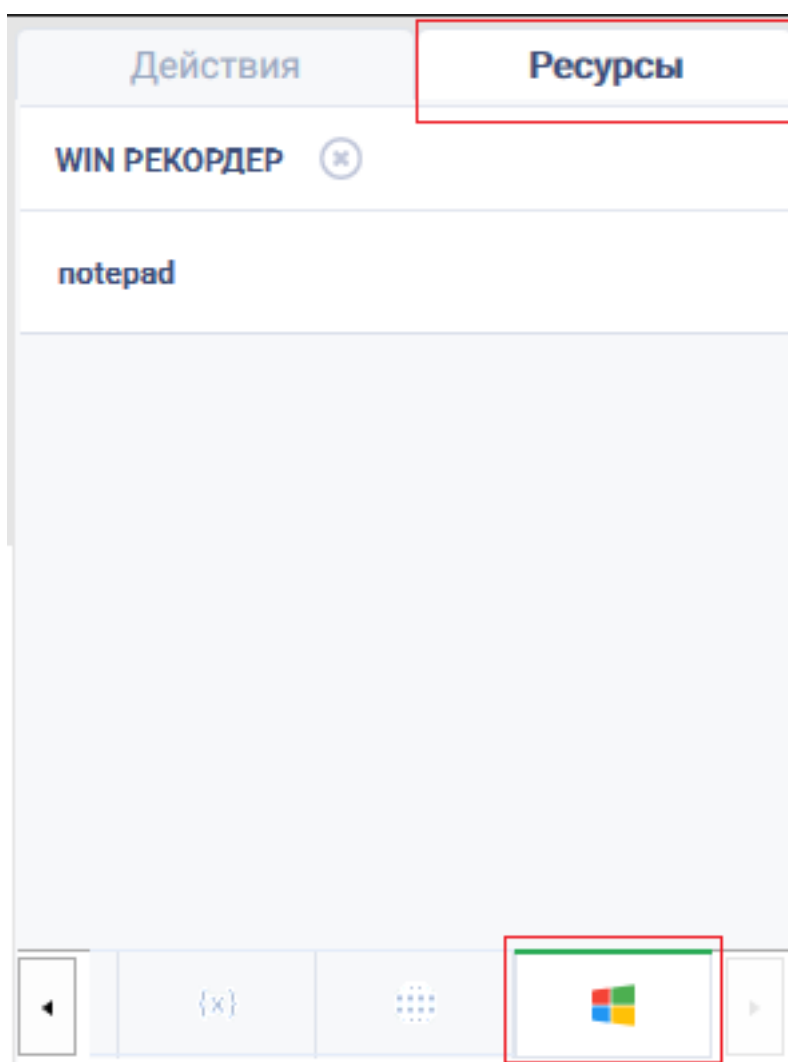
Для создания и просмотра всех win-элементов необходимо на левой панели открыть вкладку «Ресурсы» и нажать внизу на кнопку «Win-элементы».

В Robin Studio 2.0 также есть возможность автоматической записи сразу нескольких Win-элементов при помощи функции «Запись действий».

### Создание и редактирование блоков Win-элементов

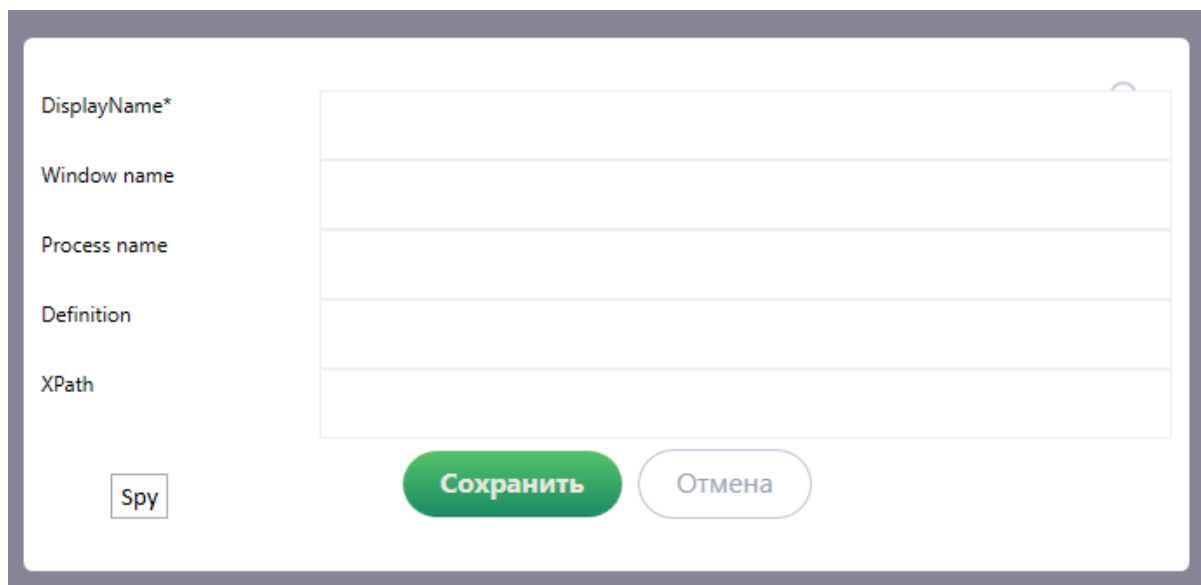
Все win-элементы автоматически распределяются по отдельным блокам в зависимости от названия процесса.

Чтобы создать новый win-элемент, нужно нажать на кнопку плюса около надписи «Win рекордер». Появятся поля, которые должны быть обязательно заполнены:



«Process name» - название процесса, в одном из окон которого расположены необходимые элементы.  
 «DisplayName» - название окна, в котором расположены элементы. Имя окна и имя процесса могут совпадать.

На распределение элементов по блокам влияет только имя процесса, которое будет указано в параметрах элемента. При этом имя окна, указанное в параметрах элемента, может не совпадать с тем, которое указано в параметрах всего блока.



После заполнения полей необходимо нажать на кнопку «Сохранить».

Если нужно выйти с формы добавления win-элемента без сохранения внесенных изменений, нажмите на кнопку «Отмена» или на кнопку:



У каждого блока при наведении на него отображаются 3 кнопки:

1. Добавить - открывает окно для ввода параметров нового элемента.



2. Редактировать - раскрывает поля «Имя процесса» и «Имя окна» для их редактирования.

3. Развернуть - разворачивает блок в виде списка элементов, которые в него входят.

При переименовании блока (изменении названия процесса), название процесса автоматически меняется и во всех элементах, входящих в этот блок.

Однако при изменении названия процесса в параметрах элемента, этот элемент будет перенесен в блок с соответствующим названием. Если же блока с таким названием еще не было, он будет создан автоматически.

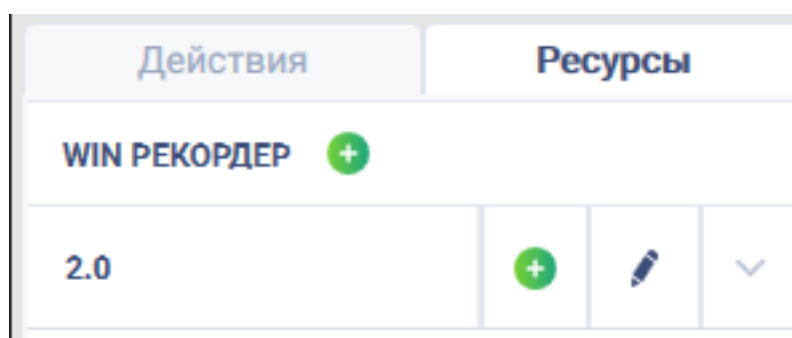


## Добавление и редактирование Win-элементов

Чтобы добавить в необходимый блок новый элемент, необходимо нажать на кнопку плюса в этом блоке.

Открывается окно с параметрами элемента:

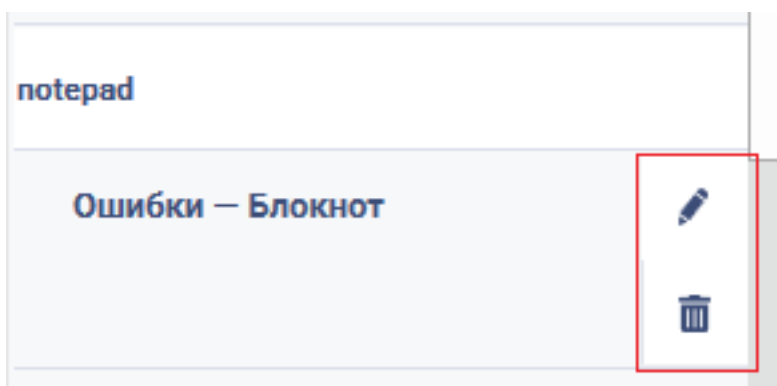
Параметр	Тип	Назначение
Window name	Поле	Имя окна, в котором робот будет определять необходимый элемент. При использовании Рекордера заполняется автоматически.
Process name	Поле	Автоматически заполнено названием блока (названием процесса), на котором нажали кнопку «Добавить». При использовании шпиона автоматически заполняется названием процесса, которое определил шпион. При использовании Рекордера или шпиона заполняется автоматически.
Display Name	Поле	Название элемента – именно оно отображается в общем списке элементов. При использовании шпиона или Рекордера заполняется автоматически именем окна, в котором был определен этот элемент.
Definition	Поле	В данном поле указываются уникальные идентификаторы элемента. При использовании шпиона или Рекордера заполняется автоматически. Отображаются только те идентификаторы которые пользователь посчитал нужным оставить, а не все, которые могут быть у элемента.
XPath	Поле	Путь до элемента, определяемый порядком следования элементов внутри друг друга.
Spy	Кнопка	При нажатии открывается окно шпиона для автоматической записи идентификаторов элементов.
Сохранить	Кнопка	При нажатии все внесенные изменения сохраняются, а окно параметров закрывается.
Отмена	Кнопка	При нажатии окно параметров закрывается, а все внесенные изменения, очищаются.



DisplayName*	
Window name	
Process name	
Definition	
XPath	

Spy
Сохранить
Отмена

У каждого элемента при наведении на него есть две кнопки:



1. Редактировать - при нажатии открывается окно с параметрами элемента. Все поля доступны для редактирования.



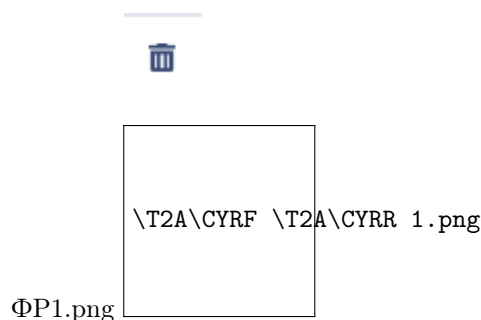
2. Удалить - при нажатии удаляет выбранный элемент.

### Использование Win-spy

Для использования рекордера Windows, необходимо нажать на кнопку SPY, для захвата элемента необходимо в появившемся окне нажать на кнопку [Захватить] в правой части окна, а затем нажать на клавишу [CTRL] и нажать ЛКМ.

В разделе «Дерево элементов окна» содержится информация о названии элемента, в разделе «Скриншот экрана» - скриншот элемента.

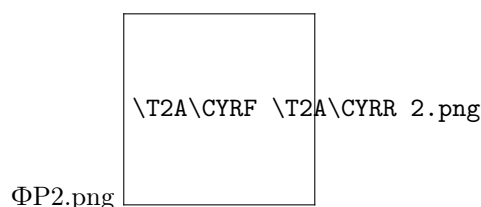




Примечание. В окне шпиона отображаются скриншот и идентификаторы только того элемента, который выбран в дереве элементов. Таким образом, можно просмотреть скриншоты и идентификаторы всех элементов, которые шпион смог определить в окне приложения, и сохранить нужный.

При нажатии на кнопку «Построить дерево» будет показано «дерево» (список элементов) выбранного приложения Windows.

При осуществлении ЛКМ по элементу дерева будет показываться скриншот элемента и его параметры.



В разделе «Список свойств» содержится список параметров окна и набор кнопок. При нажатии на кнопку [Захватить] будет произведен «захват» окна Windows, при нажатии на кнопку [Найти] будет осуществлена проверка того, чтобы выбранный в левой части интерфейса элемент можно найти на экране.

Красной рамкой выделяется найденный элемент.

При нажатии на кнопку [Выбрать уникальные] осуществляется фильтрация выбранных параметров до минимального набора для идентификации элемента.

В поле «Порядок», в котором можно вручную указать порядковый номер (относительно других элементов в окне приложения) выбранного элемента. В этом случае поле не будет заполнено автоматически.

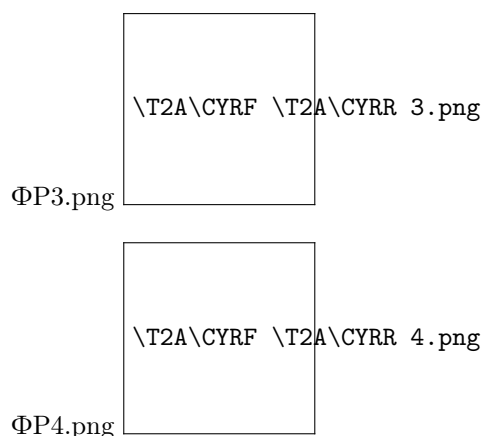
В окне «Статус» отображаются количество найденных элементов на экране.

Для сохранения выбранного элемента необходимо нажать на кнопку [Сохранить]. Выбранные элементы сохраняются в формате JSON.

## 2.10.4 Web-элементы

Web-recorder - инструмент для записи действий пользователя в браузере.

Recorder записывает манипуляции пользователя в браузере и автоматически производит добавление определенных действий на флору в Studio.



### Порядок работы

1. Web-recorder производит запись действий пользователя, и отправляет их в Robin Studio с помощью технологии Web Sockets;
2. В Robin Studio создаётся ресурс с типом «web-элемент»;
3. На основании параметров из web-элемента вызывается соответствующее действие и располагается на рабочем поле;
4. Параметры действия заполняются параметрами из web-элемента (по умолчанию «Default attribute»);
5. При запуске действия ПО обращается к параметрам веб-элемента.

### Внимание! Web-элементы поддерживают только браузер Chrome.

Все добавленные действия автоматически предзаполняются входными параметрами, полученными из браузера. В случае если параметр не найден действием, автоматически выбирается следующий параметр из веб-элемента.

### Список параметров получаемых Recorder

При добавлении ресурса, путем нажатия на кнопку плюса, будет открыто окно «Добавление нового ресурса».

Поля для заполнения:

Параметр	Значение параметра
DisplayName	Отображаемое имя элемента
Page title	Имя страницы
Url	Адрес страницы
ElementId	Идентификатор элемента
ElementName	Имя тега
XPath	Запрос к элементу
Full XPath	Полный запрос к элементу
Selector	Путь к элементу в html
Class	Перечисление классов элемента
TagName	Тип тега
LinkText	Адрес страницы, на который ссылается выбранный элемент
JavaScript	Код для поиска элемента на странице
IFrame	Имя отдельного HTML-документ на странице
Default Attribute	Атрибут который используется в параметрах действия по умолчанию. Возможен выбор элемента.



2.10. Панель ресурсов

✕

## ДОБАВЛЕНИЕ НОВОГО РЕСУРСА

DisplayName\*

Page title

Url

ElementId

ElementName

XPath

Selector

Class

TagName

LinkText

JavaScript

IFrame

Default Attribute

CssSelector

▼

Spy

Сохранить

Отмена

При нажатии на кнопку SPY имеется возможность забрать свойства другого элемента.

После добавления блока web-элементов пользователю становится доступен набор кнопок.

Функции кнопок:

При нажатии на кнопку SPY в появившемся окне пользователю необходимо ввести URL-страницы и нажать на кнопку [Начать].

После нажатия на кнопку [Начать] в правом нижнем углу экрана АРМа появляется окно, в котором отобразится информация о DisplayName, Xpath и Selector.

После закрытия браузера окно WebSpy заполняется соответствующими параметрами.

Для приостановки работы web-элемента необходимо нажать на кнопку «Пауза» , для завершения работы – на кнопку «Стоп». Для сохранения созданного web-элемента необходимо нажать на кнопку «Сохранить», для отказа от сохранения – на кнопку «Отмена».

#### **Список действий поддерживаемых в Recorder**

1. открыть вкладку;
2. перейти по URL;
3. отправить нажатие клавиш;
4. расширенный клик;
5. получить текст;
6. получить таблицу;
7. получить значение;
8. получить значение чек-бокса;
9. получить значение списка;
10. получить значение радио-кнопки;
11. переключиться на вкладку;
12. переключиться на фрейм.

### **2.10.5 Отслеживаемые переменные**

В ROBIN Studio 2.0 можно осуществлять отслеживание переменных, результаты которых необходимо видеть в результате отладки робота. Также отслеживать можно не только переменные, но и результаты действий. Если поставить переменную или результат действия на отслеживание, то их значение не будет видно во время выполнения, так как значения отслеживаемых переменных видны только когда робот стоит на паузе.

Вкладка «Отслеживаемые переменные» находится на панели «Ресурсы».

Существует несколько способов для добавления переменных и результатов действий на вкладку «Отслеживаемые переменные».

ДОБАВЛЕНИЕ НОВОГО РЕСУРСА

DisplayName\*

Page title

Url

ElementId

ElementName

XPath

Selector

Class

TagName

LinkText

JavaScript

IFrame

Default Attribute

CssSelector

Сpy

Сохранить

Отмена

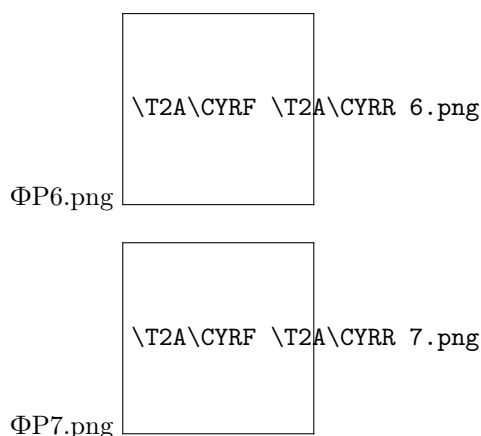
Начать со страницы

Начать

Yandex

\T2A\CYRF \T2A\CYRR 5.png

ФР5.png



### 1. Добавление уже созданной переменной на вкладку «Отслеживаемые переменные»

1.1 Для выбора переменных с целью дальнейшего их отслеживания, необходимо кликнуть на пиктограмму плюса, после чего отобразится список с переменными.

1.2 Для добавления нужной переменной на вкладку «Отслеживаемые переменные» следует из выпадающего списка нажать на выбранную переменную.

После она появится на вкладке «Отслеживаемые переменные». Также можно воспользоваться строкой «Поиск» для поиска нужной переменной.

### 2. Добавление переменной на вкладку «Отслеживаемые переменные» с помощью ее создания.

2.1 В панели «Ресурсы» необходимо перейти на вкладку «Переменные».

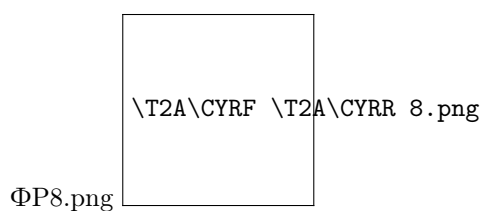
2.2 Далее нажать на пиктограмму плюса. Впоследствии произойдёт переход на форму, где представлены поля для создания новой переменной. После заполнения обязательных полей нужно установить параметр «Отслеживать при debug».

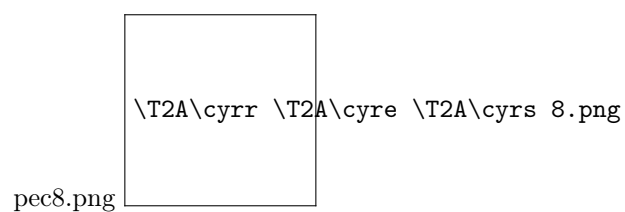
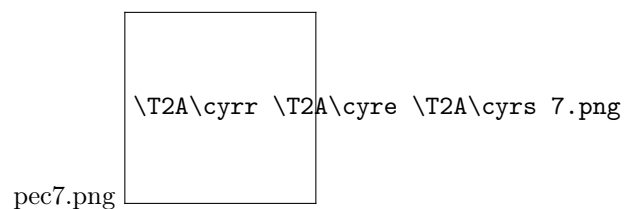
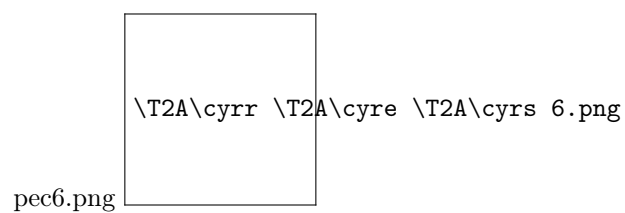
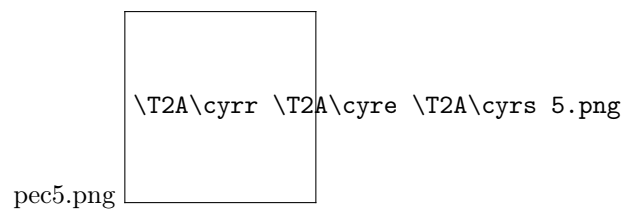
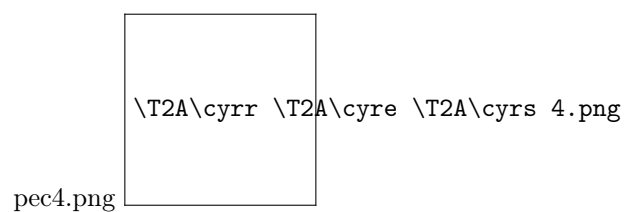
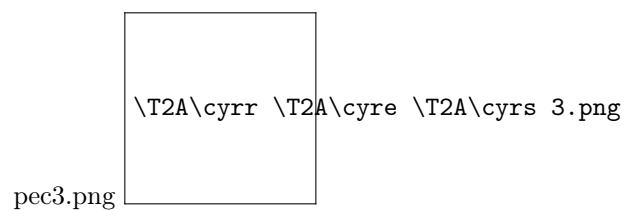
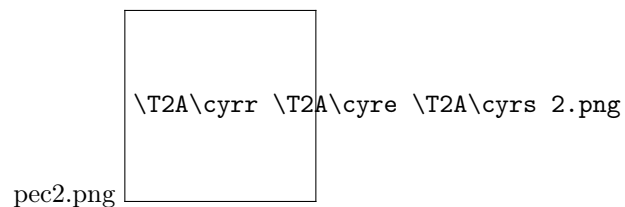
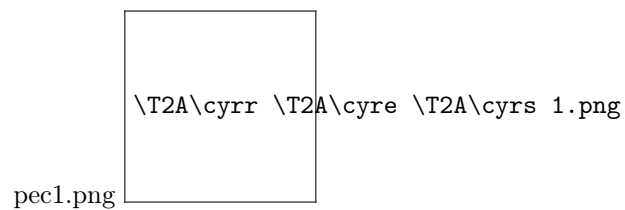
2.3 Кликнуть по кнопке «Создать».

2.4 Далее на вкладке «Отслеживаемые переменные», отобразиться созданная переменная.

2.5 Так же в списке созданных переменных, значок «Отслеживать при debug» станет активным.

### 3. Добавление переменной на вкладку «Отслеживаемые переменные» с помощью установки параметра «Отслеживать при debug» в списке созданных переменных,

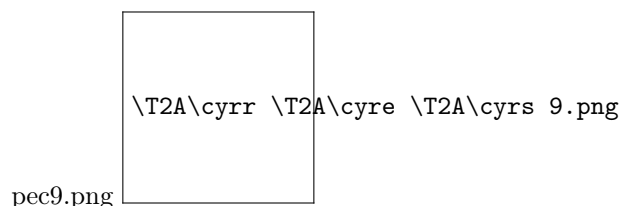




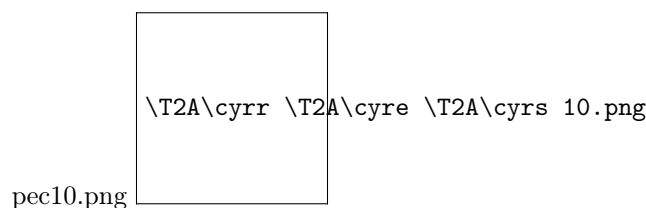


находящиеся на вкладке «Переменные».

3.1 В ресурсах на вкладке «Переменные» представлен список созданных переменных. Нужно кликнуть на значок «Отслеживать при debug».

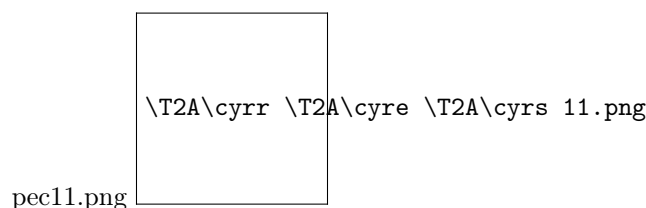


3.2 Данная переменная отобразится на вкладке «Отслеживаемые переменные».



4. Добавление результата действия на вкладку «Отслеживаемые переменные».

4.1 Открыть свойства действий и у результата действия сделать значок «Отслеживать при debug» активным.



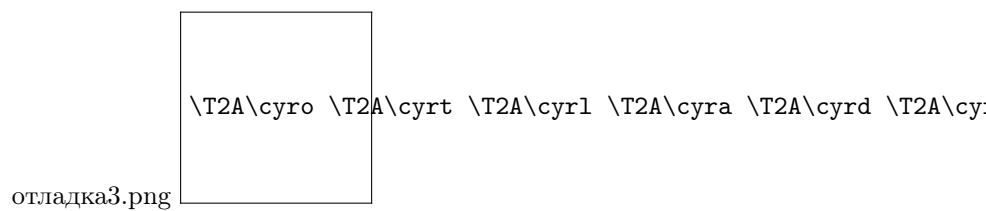
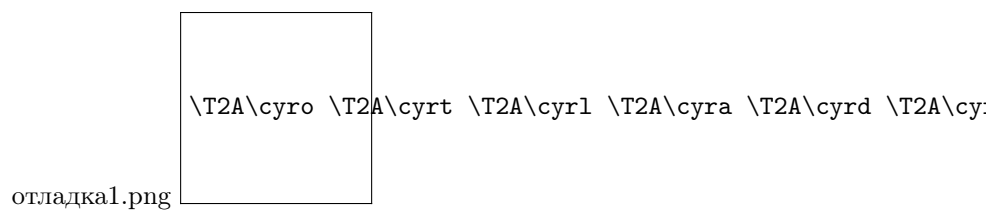
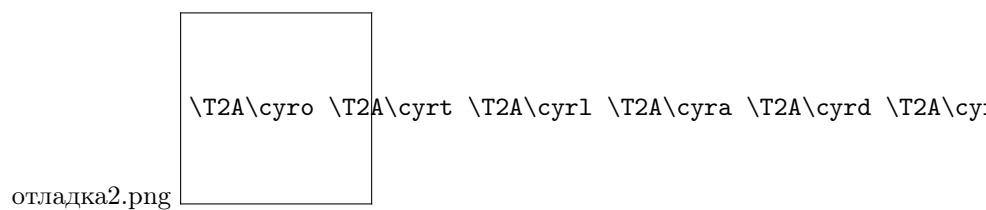
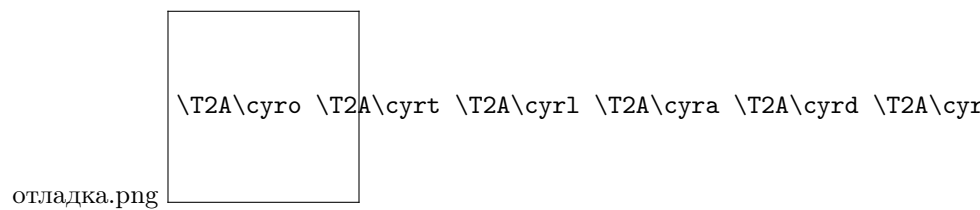
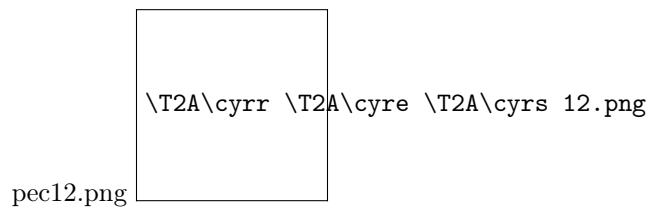
4.2 Данный результат действия отобразится на вкладке «Отслеживаемые переменные».

## 2.11 Панель управления роботом в режиме отладки (debug-режим)

Для проверки работы робота и корректировки действий необходим запуск схемы робота в Robin Studio в режиме отладки. Панель по управлению роботом в режиме отладки располагается в верхней части рабочего пространства:

Кнопки взаимодействия со схемой робота на панели управления:

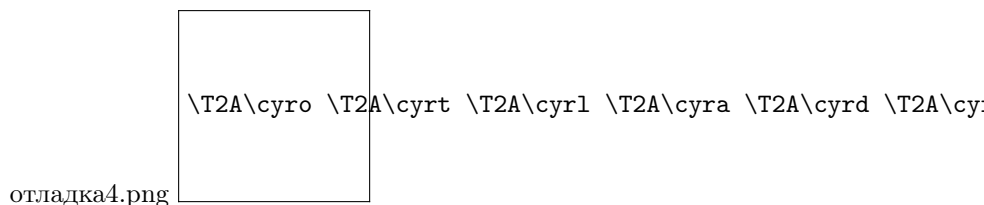
Для запуска робота в режиме отладки необходимо выбрать нужный уровень логирования, для этого необходимо нажать на кнопку, расположенную на скриншоте ниже и выбрать нужный уровень логирования, и нажать «Сохранить».



При нажатии кнопки «Отмена», уровень логирования не сохранится, и произойдет возврат в рабочее пространство ROBIN Studio.

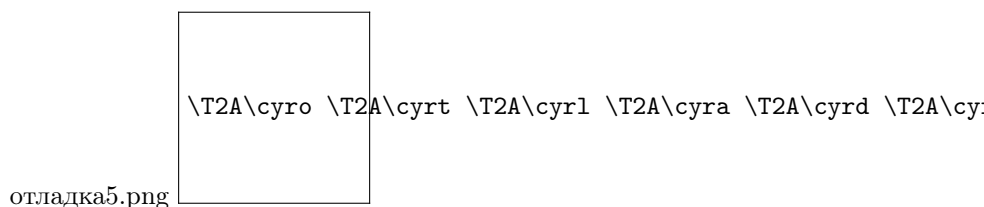
Далее схема запускается в Studio после нажатия кнопки «Старт».

При работе робота панель по управлению будет окрашиваться в зеленый цвет, при паузе - в желтый цвет. Если схема не была запущена, либо во время работы была нажата кнопка «Стоп», панель будет окрашена в серый цвет:



После запуска схемы, начинается записываться лог уровня отладки.

Лог действий находится в нижней части ROBIN Studio и выглядит следующим образом:



После завершения работы робота, пользователю доступен лог действий робота, при повторном запуске схемы лог действий в ROBIN Studio обнуляется.

## 2.12 Конструктор выражений

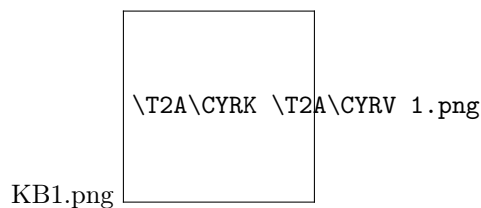
### 2.12.1 Определения

Понятие	Содержание
Выражение	Комбинация операторов и операндов
Оператор	Символ для обозначения операции. В Робин Студии различают следующие операции: арифметические, логические, сравнения, а также Робин-действия, как функции от других параметров.
Функция	Функция и операция - взаимозаменяемые понятия в контексте конструктора, поскольку унарные или бинарные операции всегда можно представить как функции от одной или двух переменных.
Операнд	Объект данных, к которому применяется оператор. В Робин Студии это может быть: константа, ресурс робота, результат выполнения одного из действий робота, результат другого выражения

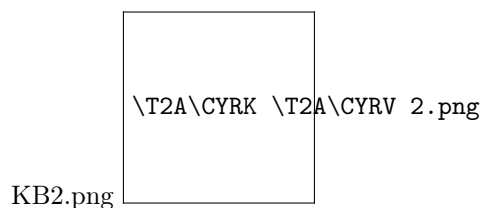
## 2.12.2 Назначение конструктора выражений

Значение параметра действия может быть задано тремя способами:

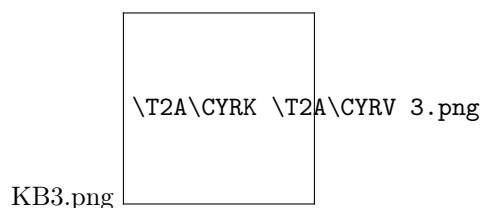
- непосредственно путем присвоения значения параметру - кнопка



- с помощью конструктора выражений - кнопка



- путем использования ресурсов робота или результатов ранее выполненных действий - кнопка



Конструктор выражений предназначен для создания выражения, результат вычисления которого будет использован как значение присваиваемое параметру действия. Конструктор выражений целесообразно применять тогда, когда значение параметра действия является вычисляемым, т.е. вычисляется по некоторой формуле.

## 2.12.3 Операторы и операции

### Арифметические операции

### Логические операции

Оператор	Операция	Вариант 1	Вариант 2
Eqv	Равно	==	EQV
Not	Отрицание	!	NOT
Not Eqv	Не равно	!=	NEQV
Or	ИЛИ		OR
Xor	Исключительное ИЛИ	^	XOR
И	И	&&	AND

КонструкторВыражений/KB50.png

WorkWithRobinStudio/StudioMain/\T2A\CYRK \T2A\cy

### Операции сравнения

Оператор	Операция	Вариант
<	меньше	
<=	меньше или равно	
==	равно	
>	больше	
>=	больше или равно	
между	в интервале	><

### Действия

Любое из действий (доступных в панели «Действия» студии) имеющее не более одного результата, может быть использовано в редакторе выражений как функция.

Во втором столбце вкладки отображается список групп действий, в третьем – список действий, в четвертом – версия реализации действия.

действия.png

\T2A\cyrd \T2A\cyre \T2A\cyrishrt \T2A\cyrs \T2A\cyrt \T2

### 2.12.4 Операнды

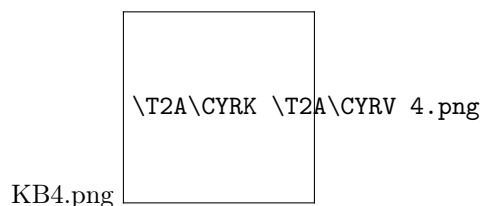
Операнд	Описание	Комментарий
Константа	Задается в конструкторе с помощью базового действия «Присвоить значение»	
Ресурс робота	В качестве значения параметра действия или операции может быть использован ресурс робота, заранее созданный в виде ресурсной переменной	Тип ресурса должен совпадать с типом параметра действия
Результат действия	В качестве значения параметра может быть использован также результат ранее выполненного действия	Тип результата действия должен совпадать с типом параметра действия
Выражения	В качестве значения параметра может быть использовано выражение	Результат выражения должен совпадать с типом параметра действия

## 2.12.5 Рекомендации

Конструктор выражений является достаточно мощным инструментом и чтобы не запутаться в его использовании целесообразно придерживаться следующих рекомендаций.

### Сложное - сложному, простое - простому

Другими словами, для выполнения простых присвоений используйте простые инструменты, а сложные инструменты - только для сложных выражений. Например, операция присвоения значения параметру является тривиальной и может быть выполнена с помощью кнопки инструмента «Параметры» панели «Свойства действия».



Тем не менее, для сравнения мы рассмотрим далее в первом примере как можно присвоить значение параметру действия с помощью конструктора выражений.

### Пример построения сложного выражения в конструкторе выражений

Допустим мы хотим использовать в качестве значения параметра действия значение переменной  $z$ , полученной в результате следующих вычислений:  $x = f1(a,b)$ ;  $y = f2(c,d)$ ;  $z = f3(x,y)$ , где  $a, b, c, d$  - параметры функций,  $f1, f2, f3$  - функции (операции арифметические, логические, сравнения или действия),  $x, y, z$  - вычисляемые значения.

В конструкторе процесс построения выражения будет выглядеть так:

$$z = f3(f1(a,b), f2(c,d))$$

Т.е. так, как мы привыкли писать выражения в алгебре. В конструкторе мы должны будем последовательно выполнять следующие действия:

1. Выбрать из конструктора функцию (операцию)  $f3$ ;
2. В качестве первого аргумента функции  $f3$  выбрать функцию  $f1$ ;
3. В качестве второго аргумента функции  $f3$  выбрать функцию  $f2$ ;
4. У функции  $f1$  задать значения первого и второго аргумента - соответственно значения параметров «a» и «b»;
5. У функции  $f2$  задать значения первого и второго аргумента - соответственно значения параметров «c» и «d».

### Используйте результаты ранее выполненных действий

В выражении Вы можете ссылаться не только на ресурсные переменные, но и на результаты ранее выполненных действий и даже выражений.

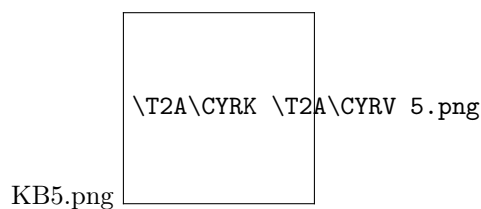
## 2.12.6 Порядок работы с конструктором

1. Вызвать конструктор выражений;
2. Построить выражение с использованием операций или действий;

3. Задать значения операндов. В качестве операндов могут быть использованы: константные значения, ресурсные переменные, результаты действий, результаты других выражений, другие операции или действия;
4. Сохранить выражение.

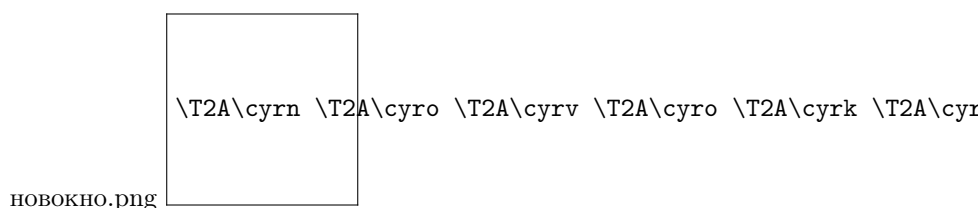
### Вызов конструктора

Для вызова конструктора выражений необходимо в поле параметра (панель «Свойства действия»), которому будет присвоен результат выражения, нажать на кнопку с тремя точками , а затем на кнопку - вызов конструктора выражений:



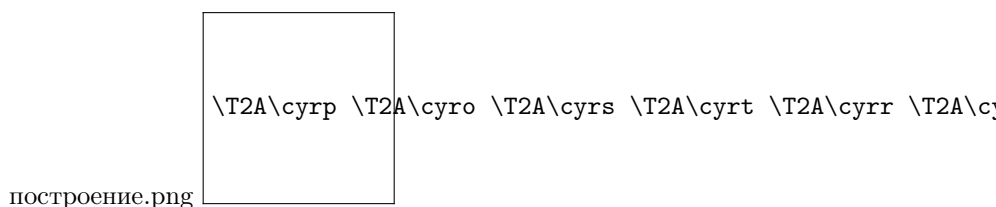
Появится новое окно с вкладками:

- Действия;
- Операции;
- Результаты;
- Ресурсы.



### Построение выражения

1. Первой выбирается та операция (или действие), результат выполнения которой будет являться результатом всего выражения
2. После двойного клика ЛКМ на версию операции (или действия) в левом верхнем углу конструктора должен появиться шаблон операции с названием операции и незаполненными операндами. Пример шаблона для операции сложения:



3. Далее заполняются операнды операции (или действия).

Чтобы заполнить операнд его поле предварительно нужно выделить (сделать активным). Для этого необходимо ЛКМ нажать справа от поля операнда. В результате поле будет выделено голубыми полосками слева и справа от него.

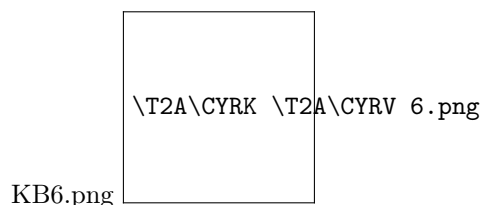
В качестве значений операнда могут быть использованы: константы задаваемые вручную, ресурсные переменные, результаты работы ранее выполненных действий или выражений, другие операции

или действия. Тут следует понимать, что если внутри поля операнда была использована новая операция или действие, то внутри поля этого операнда появится шаблон новой операции или действия, который тоже надо будет заполнить.

- Шаг 3 повторяется до тех пор пока все операнды всех операций (или действий) не будут заданы и выражение не получит законченный вид.

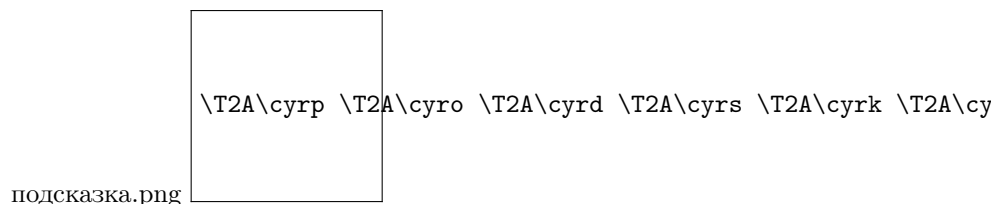
### Подсказки

Для просмотра подсказок для некоторых действий, необходимо нажать кнопку:



При этом предварительно нужно выбрать действие и нажать ЛКМ на его версию.

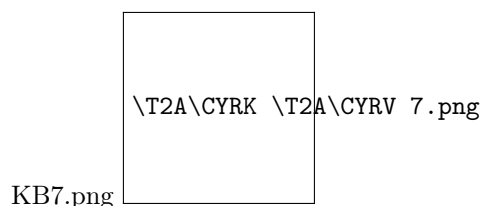
Пример подсказки:



### Использование результатов действий в выражении

Для выбора результата какого-либо действия робота, необходимо перейти на вкладку «Результаты действий».

Пример:



### Использование ресурсов в выражении

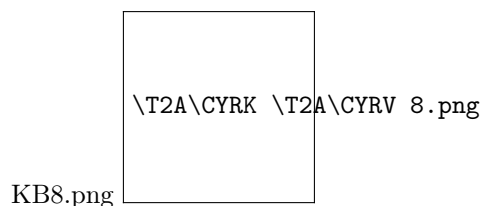
Для просмотра и выбора ресурсов робота необходимо перейти на вкладку «Ресурсы» - должны появиться группы ресурсов: Variables, Files, WebElements, WinElements. Если выбрать какую-либо из этих групп, то появится список ресурсов этой группы, которые можно выбрать для использования в выражении.

Пример:

### Сохранение выражения

После построения выражения нужно нажать на кнопку «Сохранить».

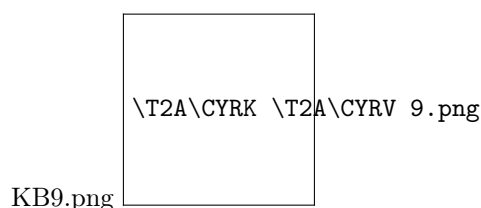




После завершения работы с конструктором в поле «Значение» параметра действия появится обозначение:

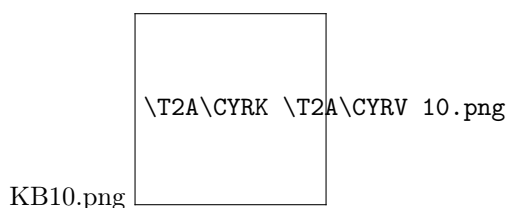
{Имя действия}.Результат вычисления выражения, где {Имя действия} - идентификатор действия вычисления выражения в блок схеме алгоритма.

Пример:



На схеме работа перед действием, для параметра которого было создано выражение, появится действие типа «Выражение» (представляющее собой группу выражений - операций выражения). В названии группы будет указано имя действия и его порядковый номер в схеме работа.

Пример:



### 2.12.7 Примеры работы с конструктором

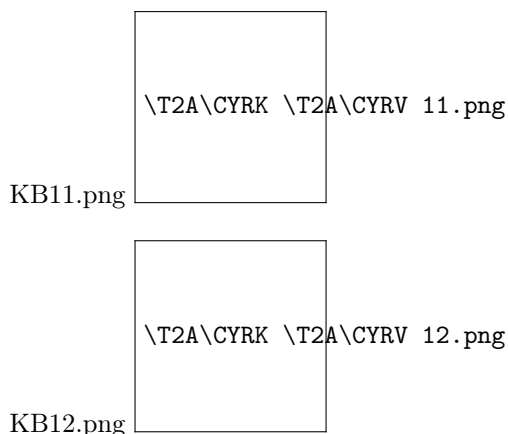
Для демонстрации возможностей конструктора выражений создадим сценарий, который создает коллекцию и наполняет ее элементами. Значение каждого нового элемента коллекции вычисляется с помощью выражения.

#### Исходные данные

В качестве исходных данных будем использовать следующие ресурсные переменные:

Переменная	Тип	Значение
v1	Число	5
v2	Число	10
date_format	Строка	dd.MM.yyyy
C	Коллекция	Результат действия

Для начала создадим первое действие для работа: «Создать коллекцию» из группы «Коллекции»: и сохраним коллекцию в предварительно созданную ресурсную переменную «C»:



### Пример 1. Присвоить параметру действия значение ресурсной переменной v1

Операция присвоения значения параметру является самой тривиальной и ее проще выполнять через панель «Параметры» в свойствах действия с помощью кнопки, чем с помощью конструктора выражений.

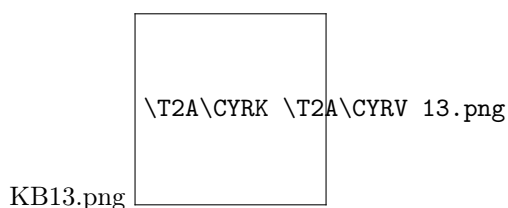
Тем не менее мы рассмотрим этот тривиальный случай в качестве первого знакомства с работой конструктора.

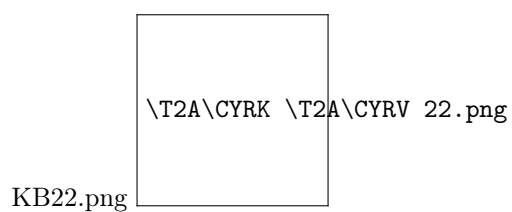
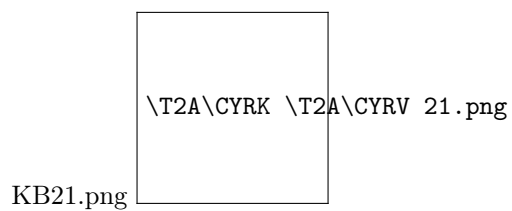
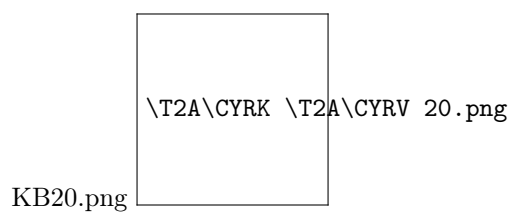
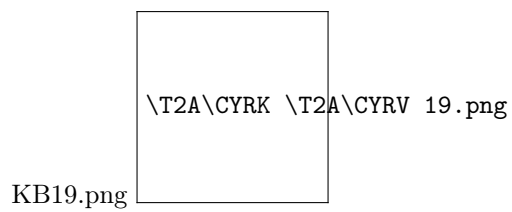
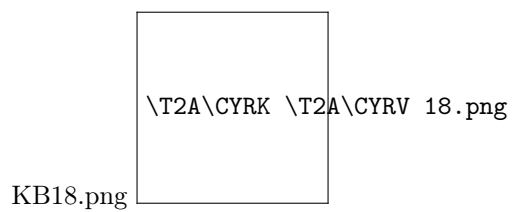
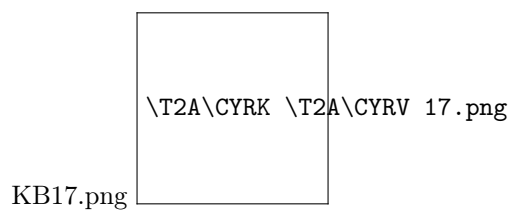
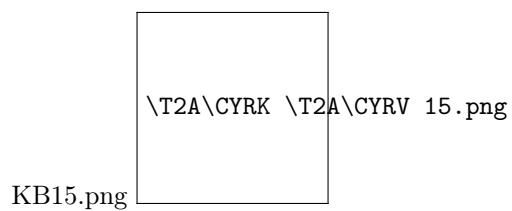
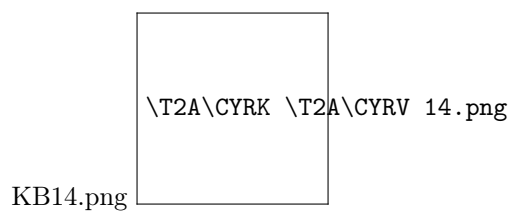
Последовательность шагов:

1. Добавляем новый элемент к коллекции с помощью действия «Добавить элемент»:
2. У действия «Добавить элемент 1» задаем параметр: «Коллекция». Для этого нажимаем на кнопку с тремя точками, а затем на кнопку, указанную на скрине ниже и из списка действий робота выбираем ресурсную переменную «С»:
3. У действия «Добавить элемент 1» задаем параметр: «Значение». Для этого нажимаем на кнопку с тремя точками, а затем на кнопку, указанную на скрине ниже - вызов конструктора выражений:
4. В открывшемся окне последовательно выбираем: «Действия студии/Базовые действия/Присвоить значение» и ЛКМ дважды нажимает на реализацию действия:
5. В левом верхнем углу появится дополнительное поле для задания значения параметра:
6. У этого поля последовательно нажимаем на кнопку с тремя точками, затем на кнопку, указанную на скрине ниже и из выпадающего списка выбираем ресурсную переменную v1.
7. Нажимаем на кнопку «Сохранить».
8. Сохраним результат добавления 1го элемента в ресурсную переменную С. Для этого у результата действия нажимаем на кнопку, указанную на скрине ниже и из выпадающего списка выбираем ресурсную переменную С:

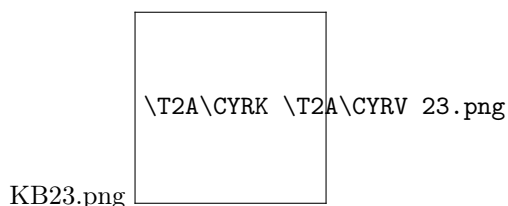
Мы будем выполнять это действие всякий раз, когда изменяем нашу коллекцию. Это потом позволит нам ссылаться на эту переменную в процессе работы с коллекцией.

9. Обращаем внимание, что на схеме робота появился новый артефакт «Выражение\_Добавить элемент 1\_Значение»:





10. Запускаем робота и проверяем результат:

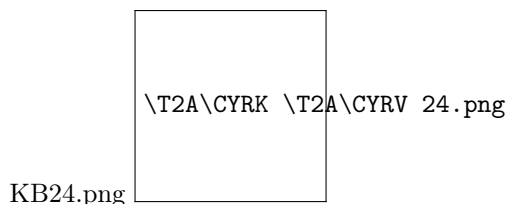


Видим, что в коллекции появился первый элемент и его значение равно 5 - такое же как и у переменной  $v_1$

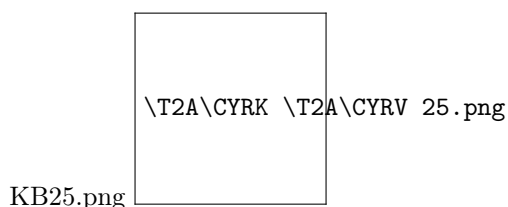
## Пример 2. Использование арифметической операции «+»

Последовательность шагов:

1. Добавляем новый элемент к коллекции с помощью действия «Добавить элемент».
2. У действия «Добавить элемент 2» задаем параметр: «Коллекция». Для этого нажимаем на кнопку с тремя точками, а затем кнопку, указанную на скрине ниже и из выпадающего списка выбираем ресурсную переменную  $C$ .

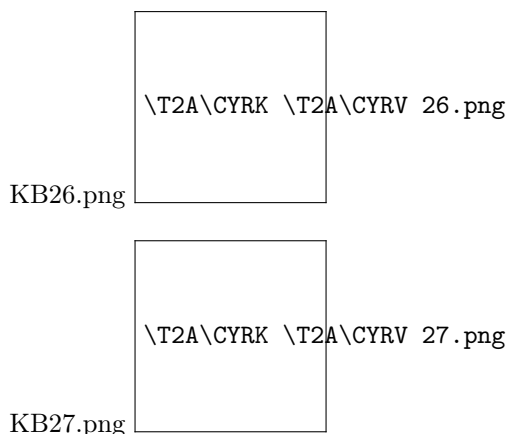


3. У действия «Добавить элемент 2» задаем параметр: «Значение». Для этого нажимаем на кнопку с тремя точками, а затем на кнопку - вызов конструктора выражений.
4. В открывшемся окне последовательно выбираем: «Операции/Арифметические/+» и ЛКМ дважды нажимаем на реализацию действия:



5. В левом верхнем углу появятся дополнительные поля для задания значений операндов. В качестве значений операндов выбираем ресурсные переменные  $v_1$  и  $v_2$ :
6. После ввода значений операндов нажимаем на кнопку «Сохранить». Все. Вторым элементом добавлен.
7. Не забываем сохранять результат добавления 2го элемента в ресурсную переменную  $C$ . Чтобы потом можно было ссылаться на эту переменную в процессе работы с коллекцией. Для этого у результата действия нажимаем на кнопку, указанную на скрине ниже и из выпадающего списка выбираем ресурсную переменную  $C$ :
8. На схеме робота появился новый артефакт «Выражение\_Добавить элемент 2\_Значение».
9. Запускаем робота и проверяем результат:

Видим, что в коллекции появился 2й элемент и его значение равно 15 - такое же как и у суммы переменных  $v_1$  и  $v_2$ .



### Пример 3. Использование действия форматирования даты

Последовательность шагов:

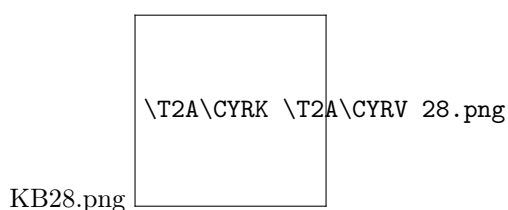
1. Добавляем новый элемент к коллекции с помощью действия «Добавить элемент».
2. У действия «Добавить элемент 3» задаем параметр: «Коллекция». Для этого нажимаем на кнопку с тремя точками, а затем на кнопку, указанную на скрине ниже и из списка действий робота выбираем «Добавить элемент 2».
3. У действия «Добавить элемент 3» задаем параметр: «Значение». Для этого нажимаем на кнопку с тремя точками а затем на кнопку - вызов конструктора выражений.
4. В открывшемся окне последовательно выбираем: «Действия/Конвертация дат/Преобразовать дату в строку» и ЛКМ дважды нажимаем на реализацию действия:
5. В левом верхнем углу появятся дополнительные поля для задания значений операндов:
6. В первом поле через кнопку с тремя точками и кнопку «календарик» задаем значение даты
7. Во втором поле через кнопку с тремя точками , а затем на кнопку, указанную на скрине ниже, выбрать ресурсную переменную date\_format:
8. После ввода значений операндов нажимаем на кнопку «Сохранить». Все. Третий элемент добавлен.
9. Не забываем сохранять результат добавления 3го элемента в ресурсную переменную С.
10. На схеме робота появился новый артефакт «Выражение\_Добавить элемент 3\_Значение».
11. Запускаем робота и проверяем результат:

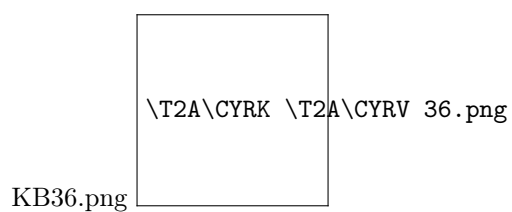
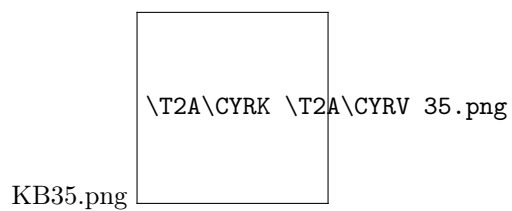
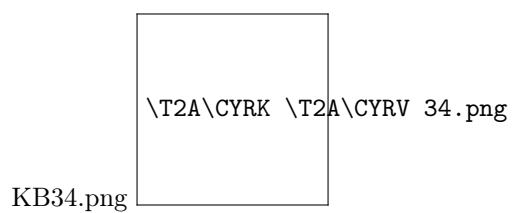
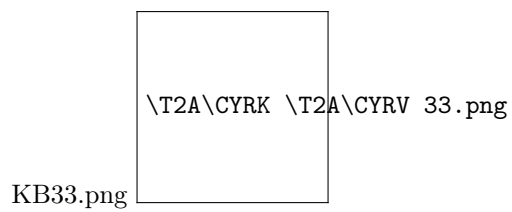
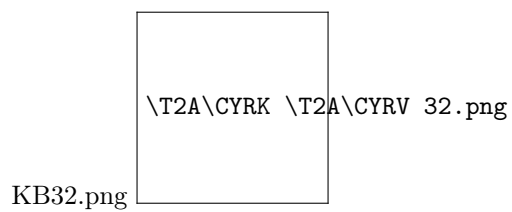
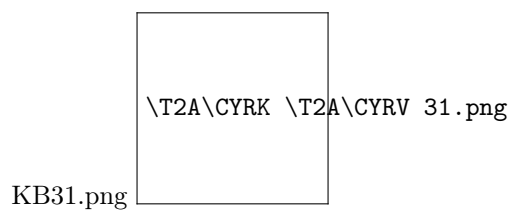
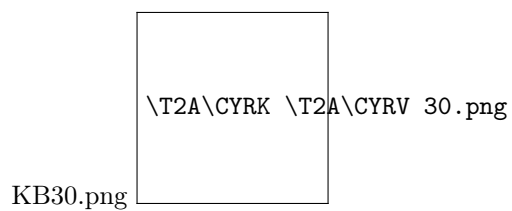
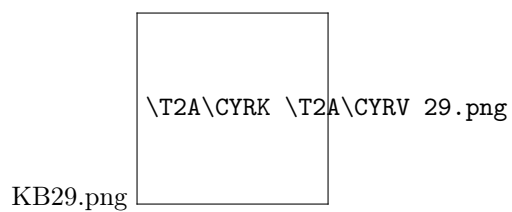
Видим, что в коллекции появился 3й элемент и его значение равно «08.08.2021», что соответствует формату, заданному переменной date\_format.

### Пример 4. Использование результатов

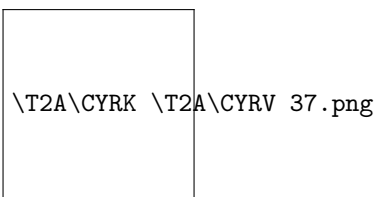
Задача: Добавить в коллекцию С 4й элемент, а значение элемента задать как сумму значений 1го и 2го элемента коллекции.

Это уже сложное выражение и прежде чем реализовывать его создадим план его реализации:





KB37.png



$$f1(f2(c,i1),f2(c,i2))$$

где

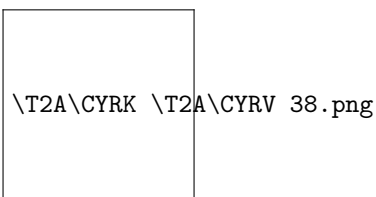
- $f1 = +$  - операция сложения;
- $f2$  - действие «Получить значение по индексу» из группы «Коллекции»;
- $c$  - коллекция на шаге «Добавить элемент 2» или позднее;
- $i1=0$  - индекс 1го элемента коллекции (значение элемента равно 5);
- $i2=1$  - индекс 2го элемента коллекции (значение элемента равно 22).

В итоге в 4м элементе коллекции должно оказаться значение равное 27.

Выполняем последовательность шагов по реализации плана построения выражения:

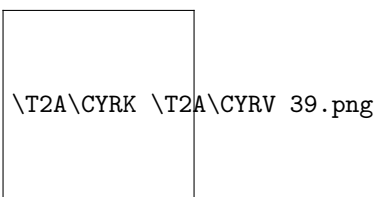
1. Добавляем новый элемент к коллекции с помощью действия «Добавить элемент»
2. У действия «Добавить элемент 4» задаем параметр: «Коллекция». Для этого нажимаем на кнопку с тремя точками, а затем на кнопку, указанную на скрине ниже, и из списка действий робота выбираем «Добавить элемент 3».

KB38.png

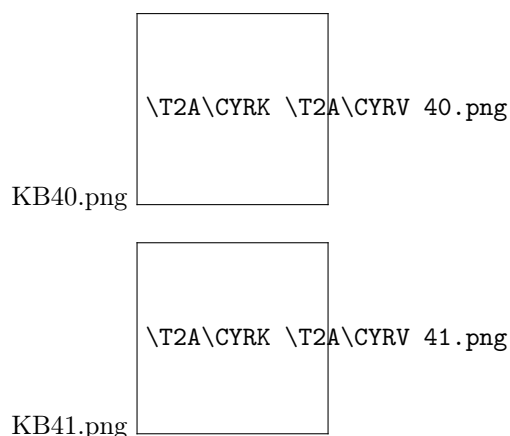


3. У действия «Добавить элемент 4» задаем параметр: «Значение». Для этого нажимаем на кнопку с тремя точками, а затем на кнопку - вызов конструктора выражений.
4. В открывшемся окне последовательно выбираем: «Операции/Арифметические/+» и ЛКМ дважды нажимаем на реализацию действия:

KB39.png



5. В левом верхнем углу появятся дополнительные поля для задания значений операндов:
6. В поле первого операнда операции «+» мы должны извлечь из коллекции первый элемент коллекции. Для этого:
  - выделяем поле первого операнда, разместив курсор мыши либо слева либо справа от этого поля и кликнув один раз ЛКМ - поле будет выделено голубыми полосками слева и справа от него;
  - выбираем действие «Получить значение по индексу» из группы «Коллекции»;
  - дважды кликаем ЛКМ на реализацию действия и получаем в поле первого операнда панель для задания параметров действия «Получить значение по индексу 1»;



- в поле первого операнда действия «Получить значение по индексу 1» задаем коллекцию через кнопку с тремя точками , а затем кнопку, указанную на скрине, и из списка действий робота выбираем ресурсную переменную С;
- в поле второго операнда действия «Получить значение по индексу 1» задаем индекс первого элемента - 0;
- получаем готовый первый операнд операции «+»:

7. Повторяем точно такую же последовательность шагов и для второго операнда операции «+»:

Только при этом задаем индекс у коллекции равным 1:

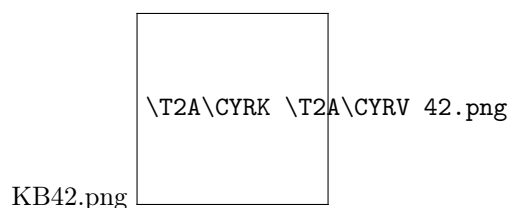
8. После ввода значений операндов операции «+» нажимаем на кнопку «Сохранить» и сохраняем результат в ресурсную переменную С. Все. Четвертый элемент добавлен.
9. На схеме робота появился новый артефакт «Выражение\_Добавить элемент 4\_Значение».
10. Запускаем робота и проверяем результат:

Видим, что в коллекции появился 4й элемент и его значение равно 20 - сумме 1го и 2го элемента коллекции, заданными ранее.

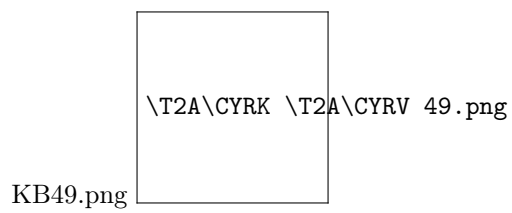
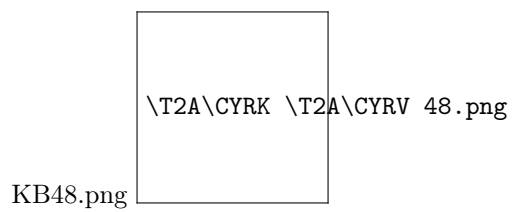
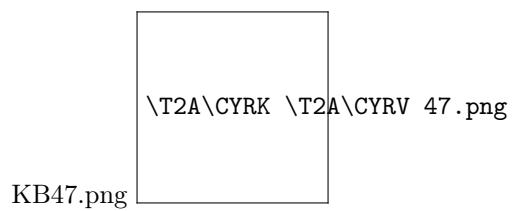
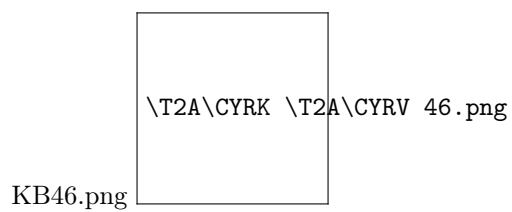
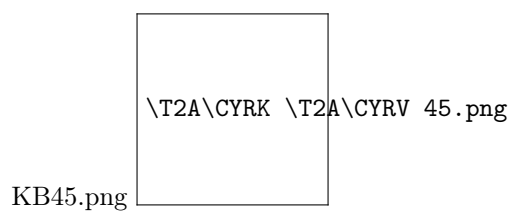
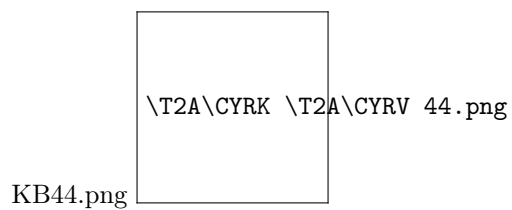
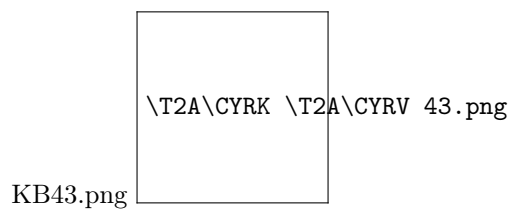
## 2.13 Пути инсталляции, размещения конфигов и логов приложений

Лог установки лежит в папке %LocalAppData%\Temp под названием «Setup Log YEAR-MONTH-DAY #NUMBER.txt» Студия и приложения системы исполнения имеют два типа поставки - «standalone» (не нужен сервер и все работает на локальном компе юзера) и «RMS» (предполагается наличие оркестратора на предприятии). В каждом из вариантов поставки, инсталлятор предлагает два варианта установки - «только для текущего юзера» и «для всех».

В таблице ниже приведены пути размещения исполнимых файлов, файлов логов и файлов конфигураций, для всех возможных вариантов дистрибуции и инсталляции приложений.





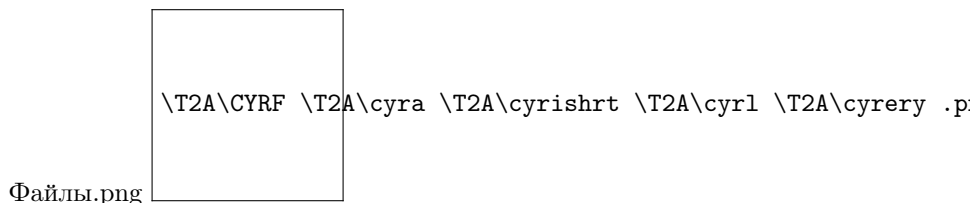


Компонент		Описание	Standalone		Enterprise, Cloud	
			For User	For All	For User	For All
EXE	Доступен для запуска юзером	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\Studio	%ProgramFiles%\Robinson Platform\{2.0.0}\Studio	%Localappdata%\Program Files\Robinson Platform\{2.0.0}\Studio	%ProgramFiles%\Robinson Platform\{2.0.0}\Studio	
LOGs	Всегда в temp. Не имеет ценности	%LocalAppData%\Temp\Robinson Platform\{2.0.0}\StudioLogs				
Config	Всегда в личном файловом пространстве	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\Studio				
EXE	При возможности не доступен для запуска юзером при варианте установки для всех, ставится на старт для учетки LOCAL SYSTEM	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\Agent	%ProgramFiles%\Robinson Platform\{2.0.0}\Agent	%SystemRoot%\System32\config\system	%SystemRoot%\System32\config\system	
LOGs	Не в temp. Снимается кибаной	%AppData%\Robinson Platform\{2.0.0}\Agent	%SystemRoot%\System32\config\system	%SystemRoot%\System32\config\system	%SystemRoot%\System32\config\system	
Config	При возможности не доступен для просмотра пользователем	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\Agent	%ProgramFiles%\Robinson Platform\{2.0.0}\Agent	%APPDATA%\Robinson Platform\{2.0.0}\Agent	%SystemRoot%\System32\config\system	
Root	Директория, поддиректории которой содержат пакеты и являются источниками (sources) для восстановления пакетов.	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\Agent	%ProgramFiles%\Robinson Platform\{2.0.0}\Agent			
Root	Директория, в которой создаются поддиректории для каждого запуска робота. В поддиректорию восстанавливаются пакеты необходимые для работы конкретного экземпляра робота. После завершения работы экземпляра робота, поддиректория с восстановленными пакетами удаляется. Другие пользователи системы не должны иметь доступ к поддиректории созданной для текущего пользователя.	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\Agent	%ProgramFiles%\Robinson Platform\{2.0.0}\Agent	%APPDATA%\Robinson Platform\{2.0.0}\Agent	%SystemRoot%\System32\config\system	
Agent	Файлы с информацией о портах, которая вычитывается студией, раннерами.	%LocalAppData%\Robinson Agent	%ProgramFiles%\Robinson Agent			
EXE	Доступен для запуска юзером	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\RobotPlayer	%ProgramFiles%\Robinson Platform\{2.0.0}\RobotPlayer	%Localappdata%\Program Files\Robinson Platform\{2.0.0}\RobotPlayer	%ProgramFiles%\Robinson Platform\{2.0.0}\RobotPlayer	
Config	Всегда в личном файловом пространстве	%LocalAppData%\Program Files\Robinson Platform\{2.0.0}\RobotPlayer				
LOGs	Всегда в temp. Не имеет	%LocalAppData%\Temp\Robinson Platform\{2.0.0}\RobotPlayerLogs				

### 2.13.1 Расположение файлов для нового агента

Новый агент получает все пути относительно своего exe, соответственно все файлы должны быть в «пешей доступности».

Внутри файлы и папки расположены следующим образом:



### 2.13.2 Расположение файлов на Linux

Таблица расположения исполняемых файлов, логов и конфигурационных файлов компонентов:

Компонент	Расположение
Studio bin	/opt/RobinPlatform/{2.0.0}/Studio
Studio Logs	/var/log/RobinPlatform/{2.0.0}/Studio
Studio Configs	/etc/RobinPlatform/{2.0.0}/Studio
Agent bin	/opt/RobinPlatform/{2.0.0}/Agent
Agent Logs	/var/log/RobinPlatform/{2.0.0}/Agent
Agent Configs	/etc/RobinPlatform/{2.0.0}/Agent
Agent Packages	/opt/RobinPlatform/{2.0.0}/Agent/Packages
Player bin	/opt/RobinPlatform/{2.0.0}/Player
Player Logs	/var/log/RobinPlatform/{2.0.0}/Player
Player Configs	/etc/RobinPlatform/{2.0.0}/Player
Robot Logs	/var/log/RobinPlatform/{2.0.0}/Robot
Runs	/var/opt/RobinPlatform/2.0.0/Runs

## 2.14 Работа с действиями

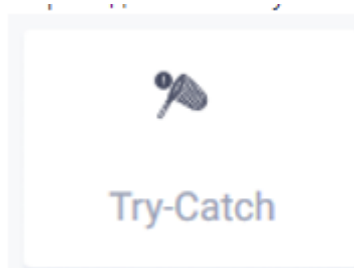
В данном разделе приведено подробное описание действий, наиболее сложных для понимания, и приведены кейсы в качестве примеров.

### 2.14.1 Try-Catch (Группа Базовые действия)

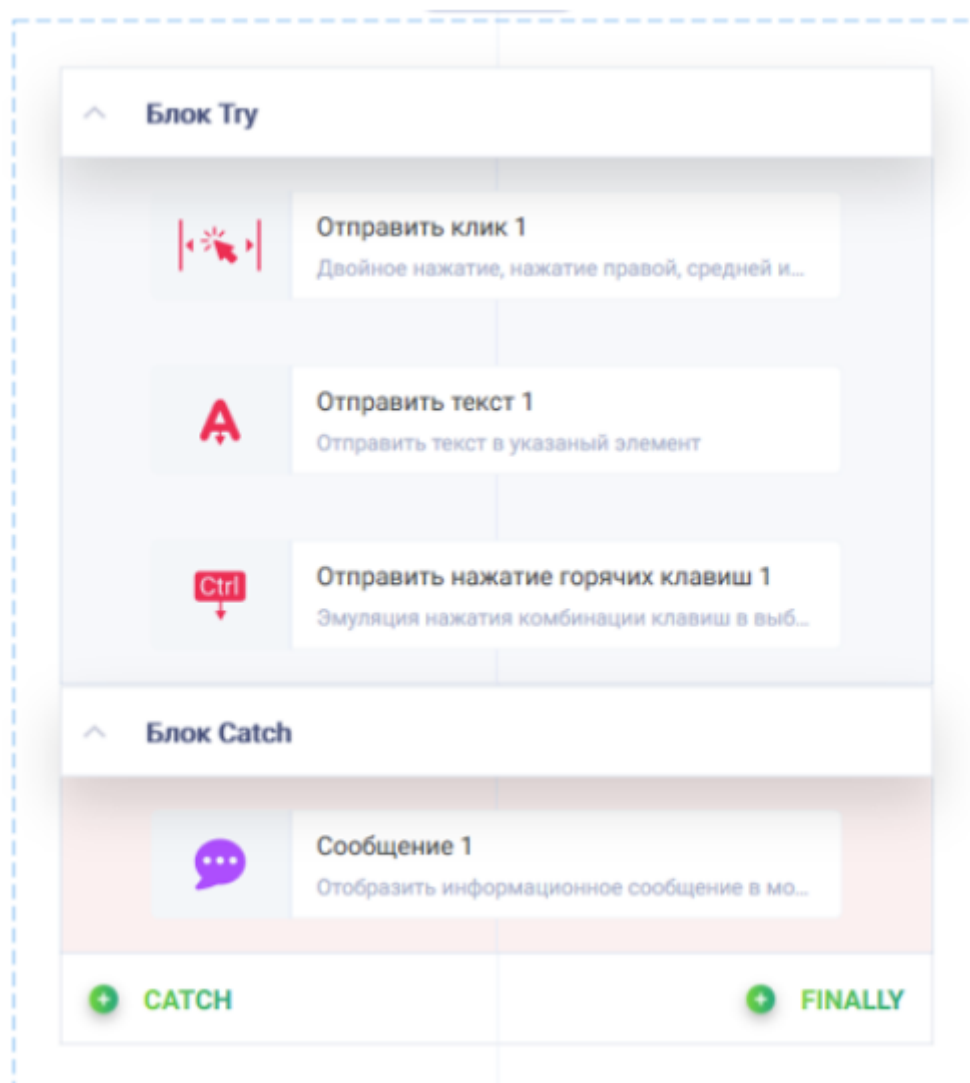
**Try-Catch** - действие группы «Базовые действия». Действие предназначено для обработки исключений в выбранных действиях, т.е. если при выполнении действия из блока «Try» произошла ошибка, указанная в параметрах действия «Try-Catch», то робот перейдет к выполнению действий, указанных в блоке «Catch».

#### Порядок работы с Try-Catch

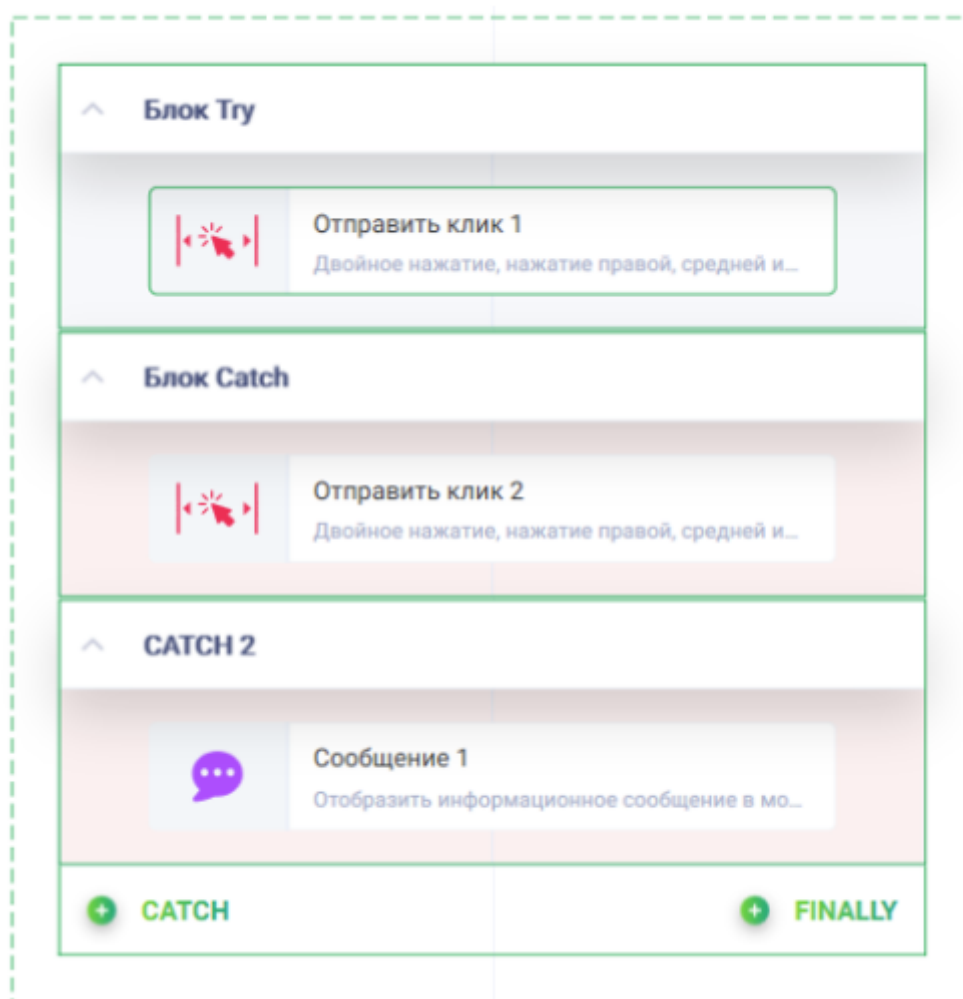
1. Выбрать действие «Try-Catch» из группы действий «Базовые действия» и перетащить на рабочую область.



2. В блок «Try» поместить действия, для которых будет выполнено исключение. В блок «Catch» поместить действия, которые будут выполнены, если возникнет исключение, т.е. какая-либо ситуация, когда действие выполнено некорректно, иными словами, возникнет ошибка. Если блок «Catch» не заполнен, то робот продолжит работу сценария, проигнорировав возникшие ошибки. Пример заполнения:



Блоков «Catch» может быть несколько, тогда для каждого блока будет создан параметр «Исключение для CATCH [Номер блока]». Это нужно, когда для каждой возможной ошибки есть свой вариант действия. Пример: Действие «Try-Catch» в сценарии:



Заполненные параметры действия «Try-Catch»:

ПАРАМЕТРЫ ^		
Исключения	0 : Robin.Exception.ElementNotFound	...
Исключения для CATCH 2	0 : Robin.Exception.Unknown	...

Таким образом, если веб-элемент не будет найден, то робот выполнит действие из блока «Catch», а если возникнет неизвестная ошибка, то робот выполнит действие из блока «CATCH 2».

3. Заполнить параметры действия «Try-Catch». Параметры заполняются в формате коллекции, представляющей собой список исключений, для обработки которых предназначена соответствующая последовательность Catch. Доступно несколько типов ошибок. Для выбора типа ошибки необходимо нажать на выбранный тип, а затем на стрелку рядом с ним.

Список типов ошибок

Доступные		Выбранные
Любая ошибка	+	
Неизвестная ошибка		
Веб-элемент не найден		
Веб-элемент вне доступа		
Параметр не найден		

Сохранить
Отмена

После перенесения всех необходимых типов ошибок, нажать кнопку «Сохранить».

Пример заполненных параметров:

ПАРАМЕТРЫ ^		
* Исключения	0 : Robin.Exception.Any 1 : Robin.Exception.Unknown	...

Пример заполненных параметров для нескольких блоков «Catch».

ПАРАМЕТРЫ ^		
Исключения	0 : Robin.Exception.ElementNotFound	...
Исключения для CAT...	0 : Robin.Exception.ElementNotReachable	...

## Блок Finally

Блок Finally является не обязательным блоком. Действия, указанные в этом блоке, выполнятся вне зависимости от того, возникла ошибка или нет. Если в блоке Finally отсутствуют действия, то робот продолжит свою работу.

Пример заполнения блока Finally:

Выше были рассмотрены общие моменты работы с действием «Try-Catch». Рассмотрим кейс «Погода», где продемонстрирована работа этого действия.

## Кейс «Погода»

**Задача:** Роботу необходимо зайти в поисковую систему, сделать запрос «Погода» и получить сегодняшнюю температуру, а затем сохранить ее в excel-файл. Задача простая, но в ходе ее выполнения могут возникнуть ошибки.

### Решение:

1. Поместим все действия сценария в блок «Try»,
2. В блок «Catch» действие «Сообщение», в котором робот сообщит, что возникла ошибка при работе с браузером.
3. В блок «CATCH 2» поместим действие «Сообщение», которое сообщит, что ошибка произошла в действии группы «Excel».
4. В блоке Finally будет информационное сообщение о том, что робот закончил работу.

Пример заполнения блока «Try»:

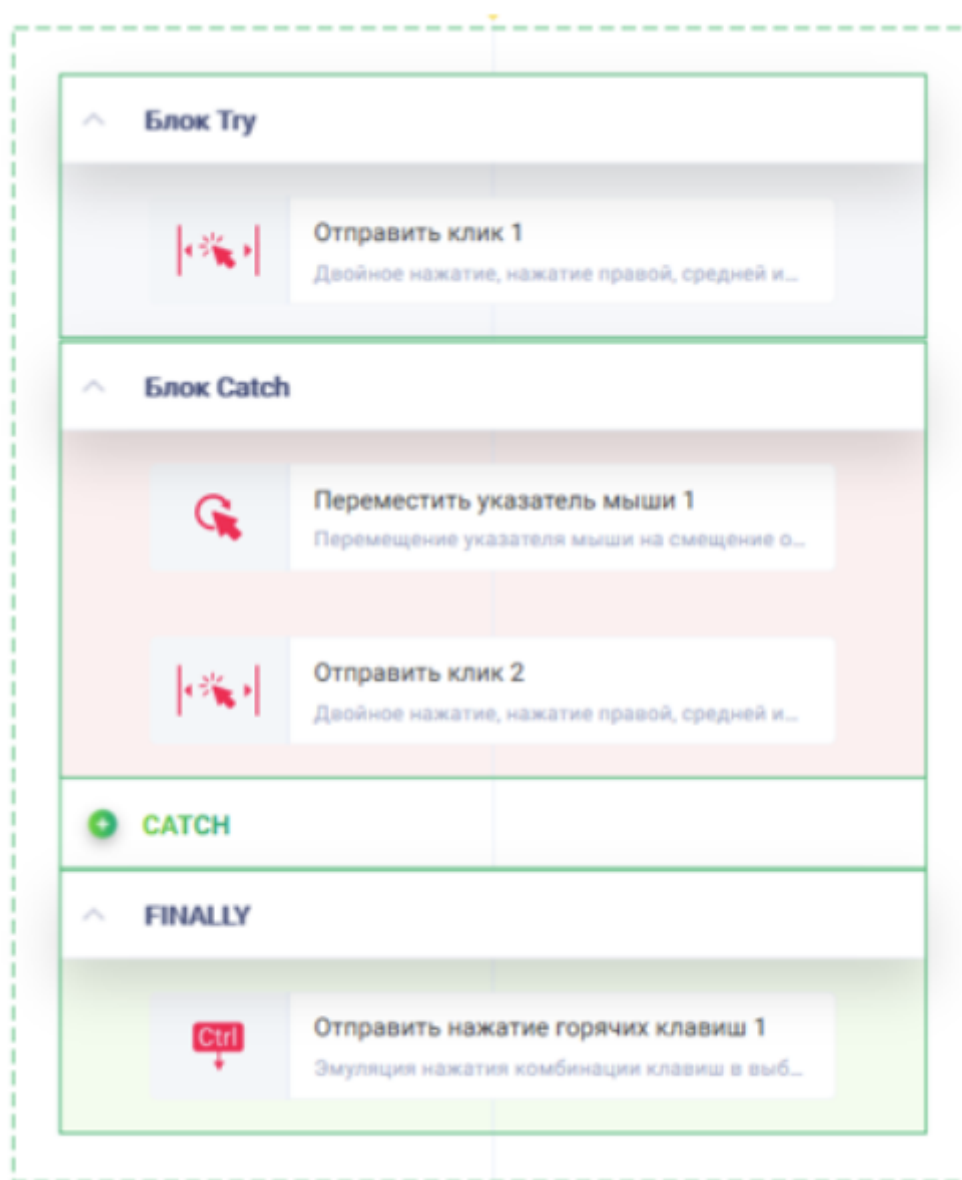
Пример заполнения блока «Catch»:

5. Заполним параметры: Выберем типы ошибок, которые могут возникнуть во время работы с действиями группы «Браузеры» для блока «Catch».

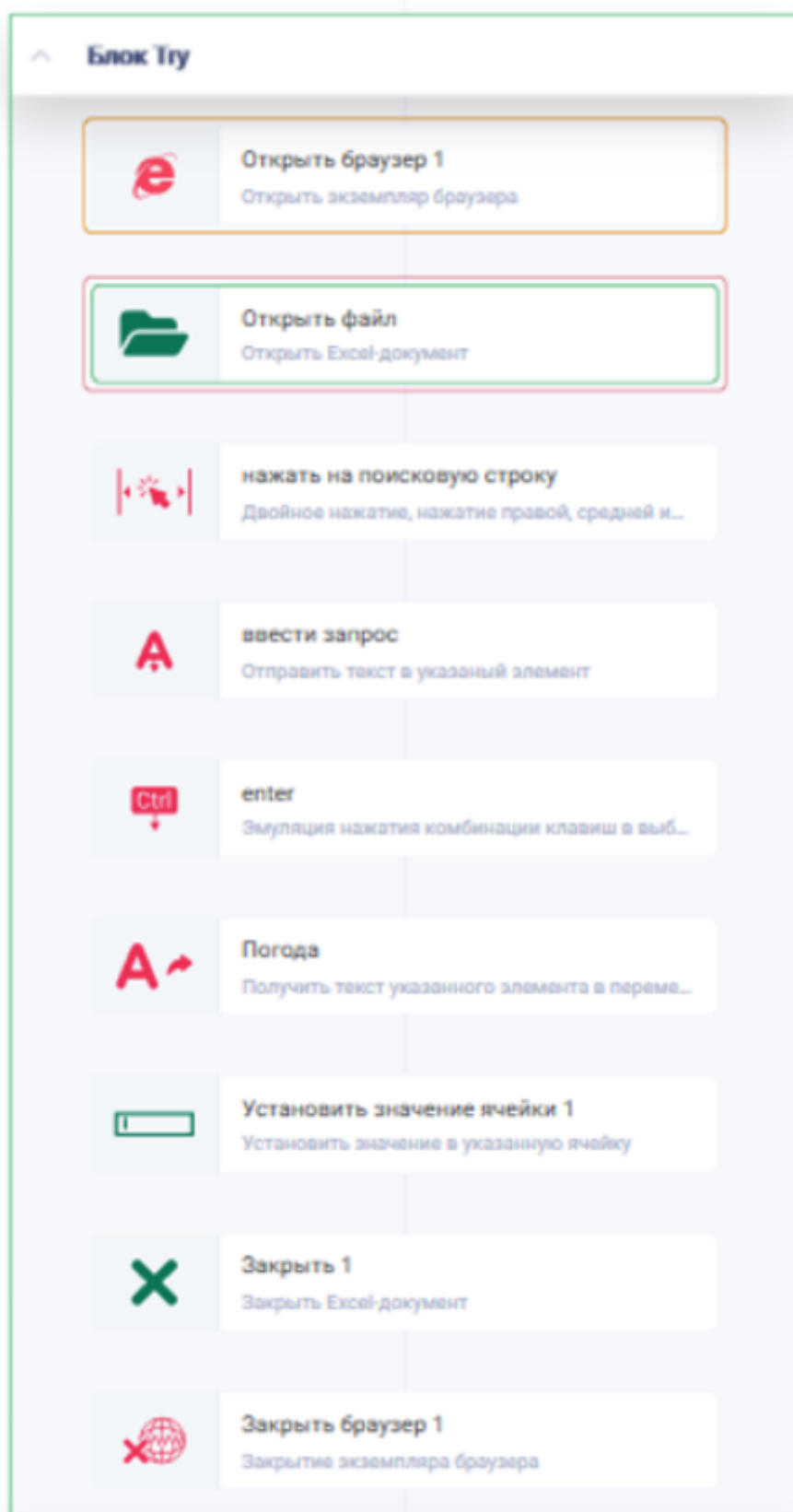
И типы ошибок, которые могут возникнуть в результате работы с действиями группы «Excel» из предложенных для блока «CATCH 2».

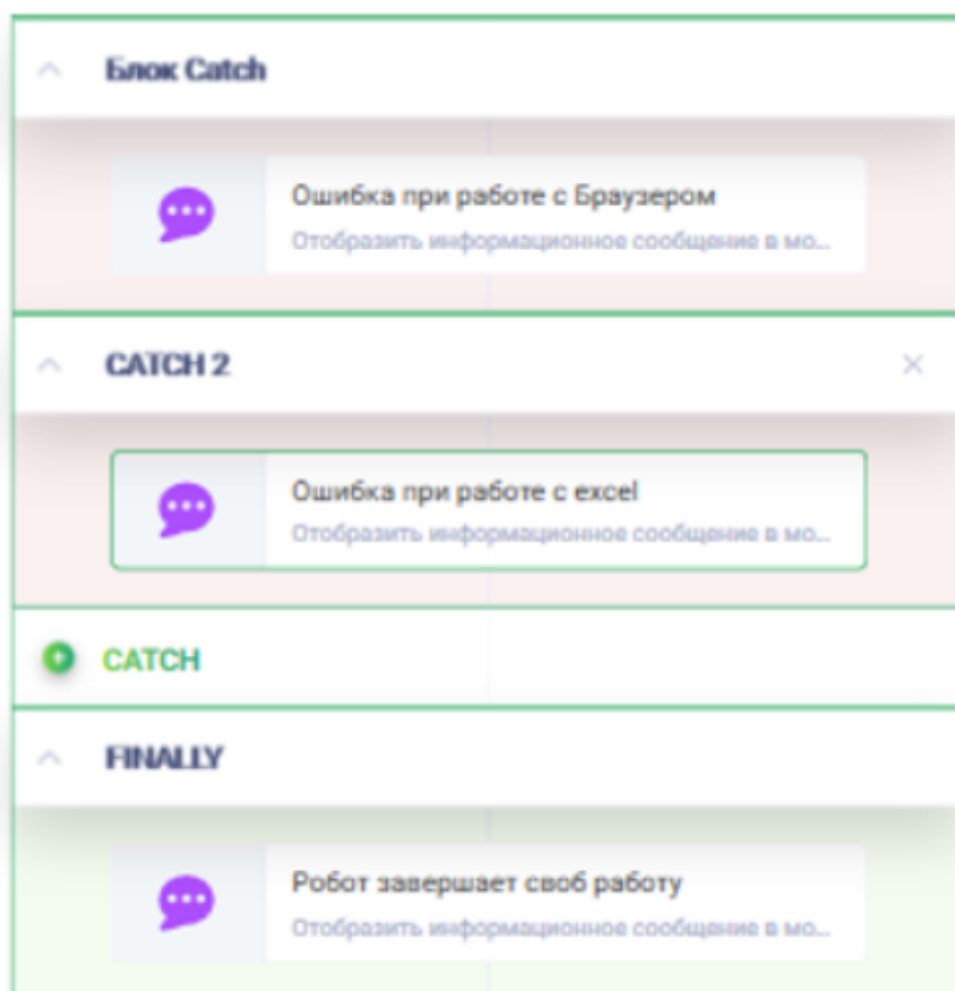
Результат:

Затем на экране появилось сообщение из блока «Finally».









Список типов ошибок

Доступные	Выбранные
Любая ошибка	Браузер не запустился
Неизвестная ошибка	Веб-элемент не найден
Браузер не запустился	Веб-элемент вне доступа
Ошибка в базе данных	
Параметр не найден	
Файл недоступен	
Файл не найден	
Веб-элемент не найден	
Веб-элемент вне доступа	

Сохранить

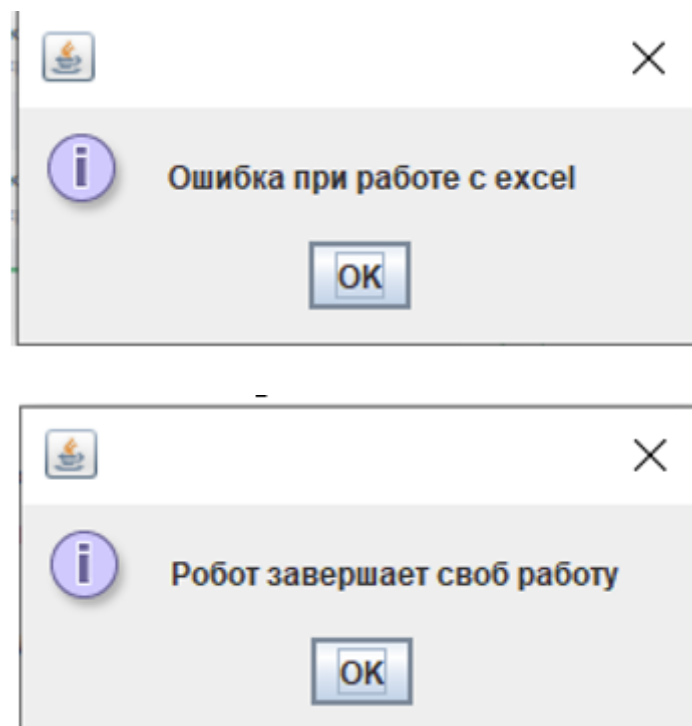
Отмена

Список типов ошибок

Доступные	Выбранные
Любая ошибка	Файл недоступен
Неизвестная ошибка	Файл не найден
Браузер не запустился	
Ошибка в базе данных	
Параметр не найден	
Файл недоступен	
Файл не найден	
Веб-элемент не найден	
Веб-элемент вне доступа	

Сохранить

Отмена



При работе робота произошла ошибка, анализируя лог в студии, можно увидеть, как робот остановил работу с действиями в блоке «Try», перешел в блок «CATCH 2» и вывел на экран информационное сообщение.

Из-за наличия блока «Try-Cath» робот завершил свою работу без ошибок, несмотря на то, что при открытии файла возникла ошибка.

### 2.14.2 Распознавание документов (OCR)

В данном разделе пошагово расписана работа с действиями группы «Распознавание текста»:

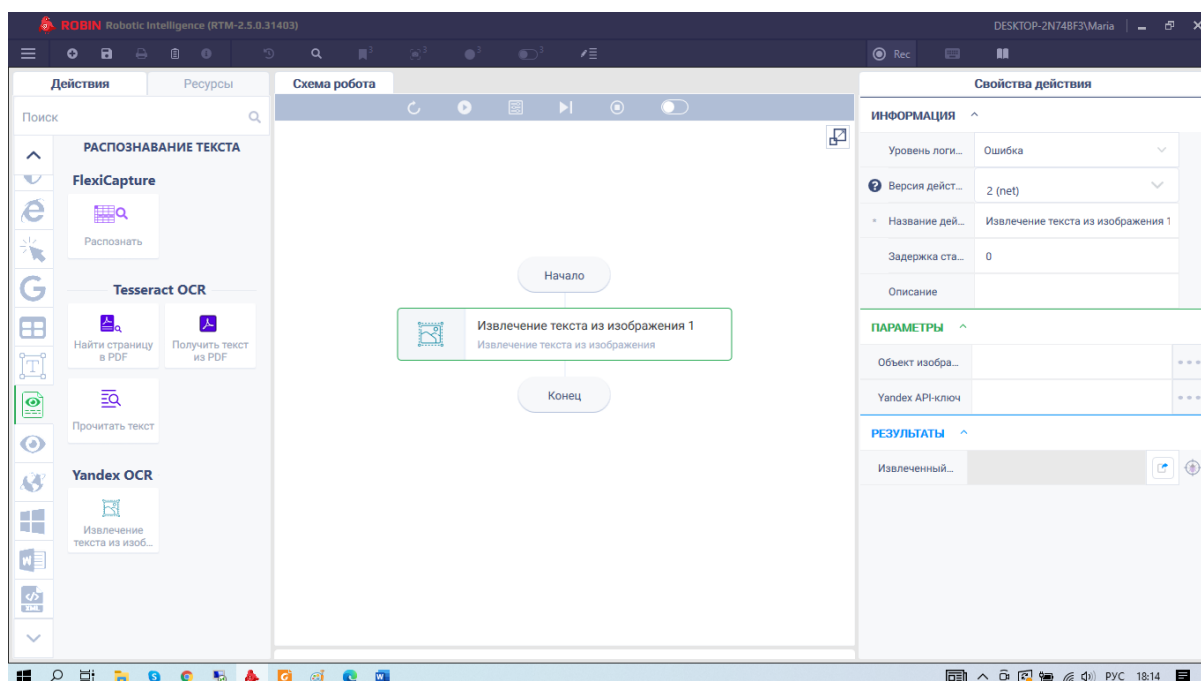
- Подгруппа Yandex OCR - Извлечение текста из изображения;
- Подгруппа «Tesseract OCR» - Найти страницу в PDF;
- Подгруппа «Tesseract OCR» - Получить текст из PDF;
- Подгруппа «Tesseract OCR» - Прочитать текст;
- Подгруппа «FlexiCapture» - Распознать.

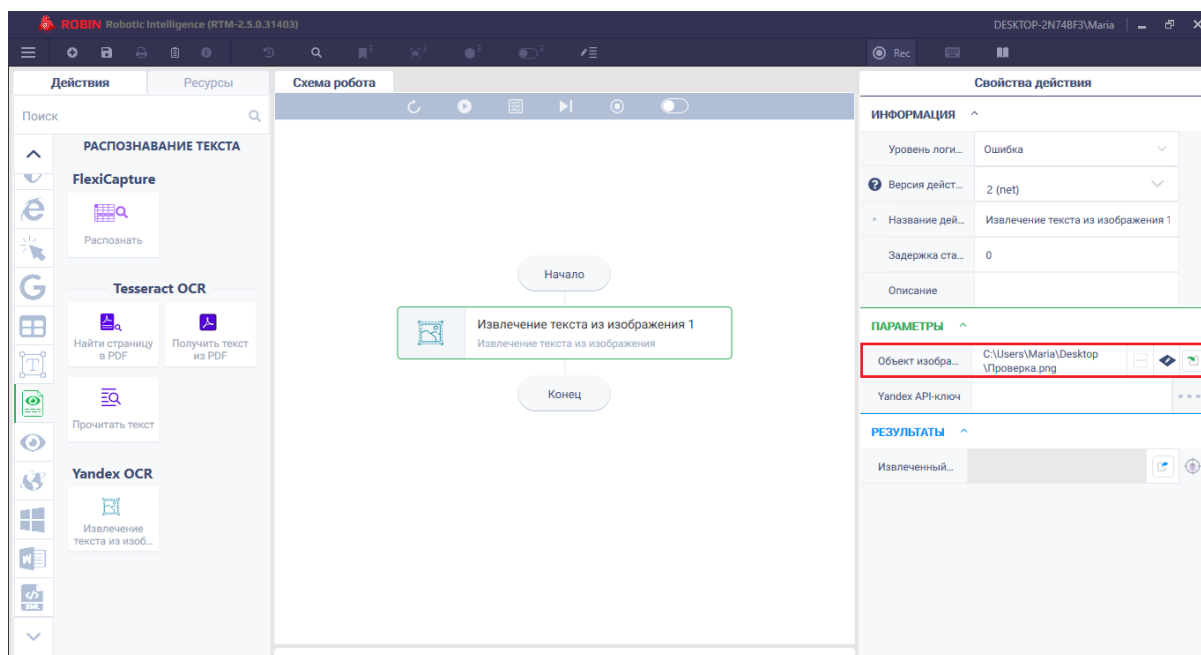
#### Подгруппа Yandex OCR - Извлечение текста из изображения

Для настройки действия «Извлечение текста из изображения» необходимо выполнить следующие шаги:

1. Перенести действие «Yandex OCR» на рабочую область студии.
2. Записать путь до нужного изображения в разделе «Параметры» в поле «Объект изображения».

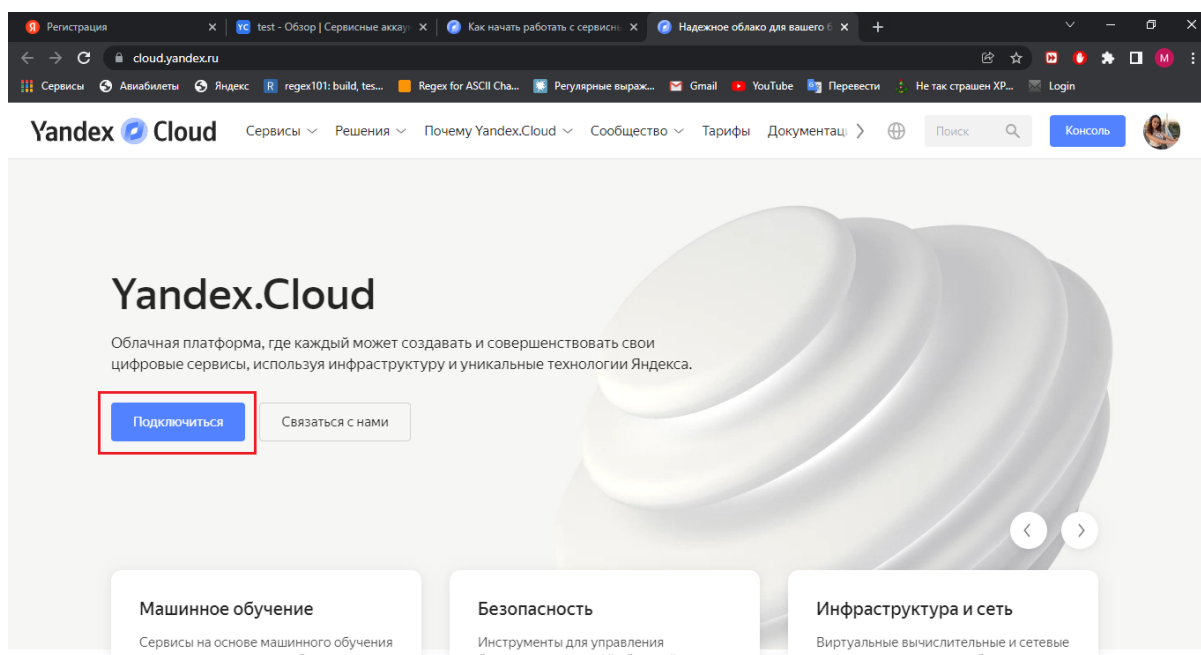
18:33:41.436	Подготовка к запуску робота
18:33:45.546	Выполнение робота началось
18:33:46.066	Try-Catch 1 действие началось
18:33:46.163	вызов подсценария 1 на движке java
18:33:51.999	Открыть браузер 1 действие началось
❗ 18:33:52.422	Открыть браузер 1 Точная версия драйвера не найдена. Требуется драйвер для 'CHROME' / '96'. Укажите его в конфиг файле.
18:33:58.310	Открыть браузер 1 действие завершилось
18:33:58.600	вызов подсценария 1 на движке java
18:33:58.889	Открыть файл действие началось
❌ 18:33:59.037	Открыть файл bf5b77b4-2512-4c00-84f1-efcd66647db9 возникло Robin.Exception.FileNotFoundException в Robin.Excel.Open: File 'C:\Users\persi\OneDrive\Рабочий стол\Лист Microsoft Excel.xlsx' is locked by something
18:33:59.068	вызов подсценария 5 на движке java
18:33:59.645	Ошибка при работе с excel действие началось
18:34:30.815	Ошибка при работе с excel действие завершилось
18:34:30.970	вызов подсценария 5 на движке java
18:34:30.986	вызов подсценария 6 на движке java





3. Записать ключ авторизации API Яндекс.Облако в поле «Yandex API-ключ». Для получения этого ключа нужно выполнить следующие действия:

- Перейти на сайт Yandex.Cloud по ссылке <https://cloud.yandex.ru/>
- Кликнуть по кнопке «Подключиться».

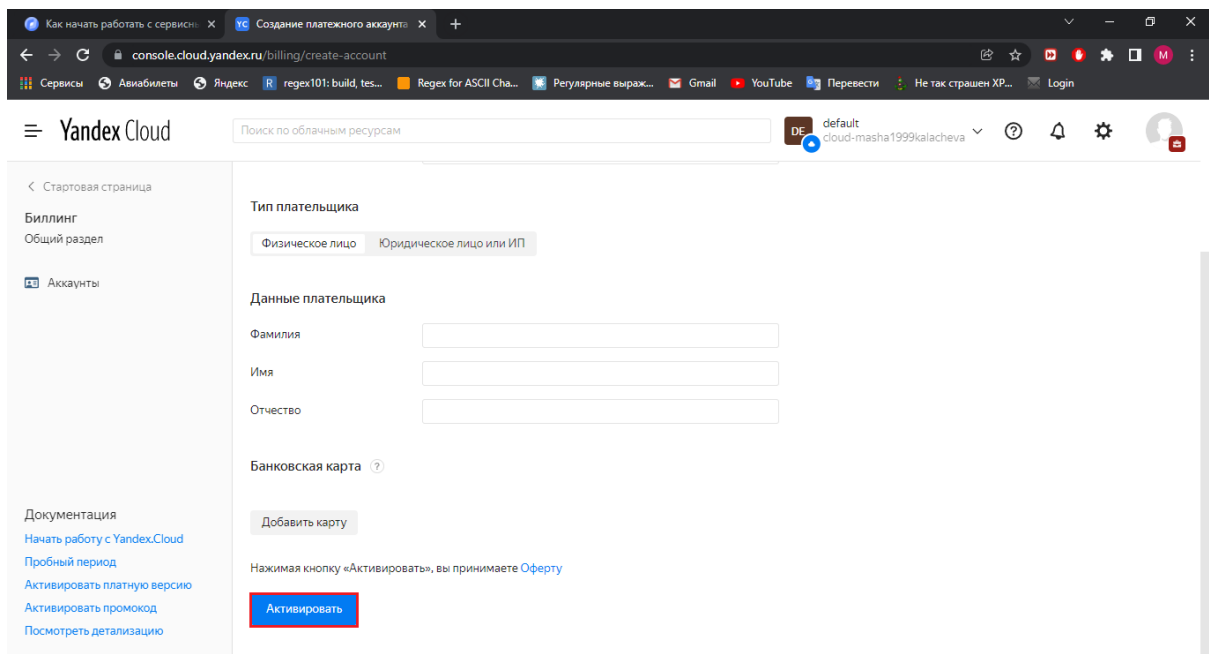
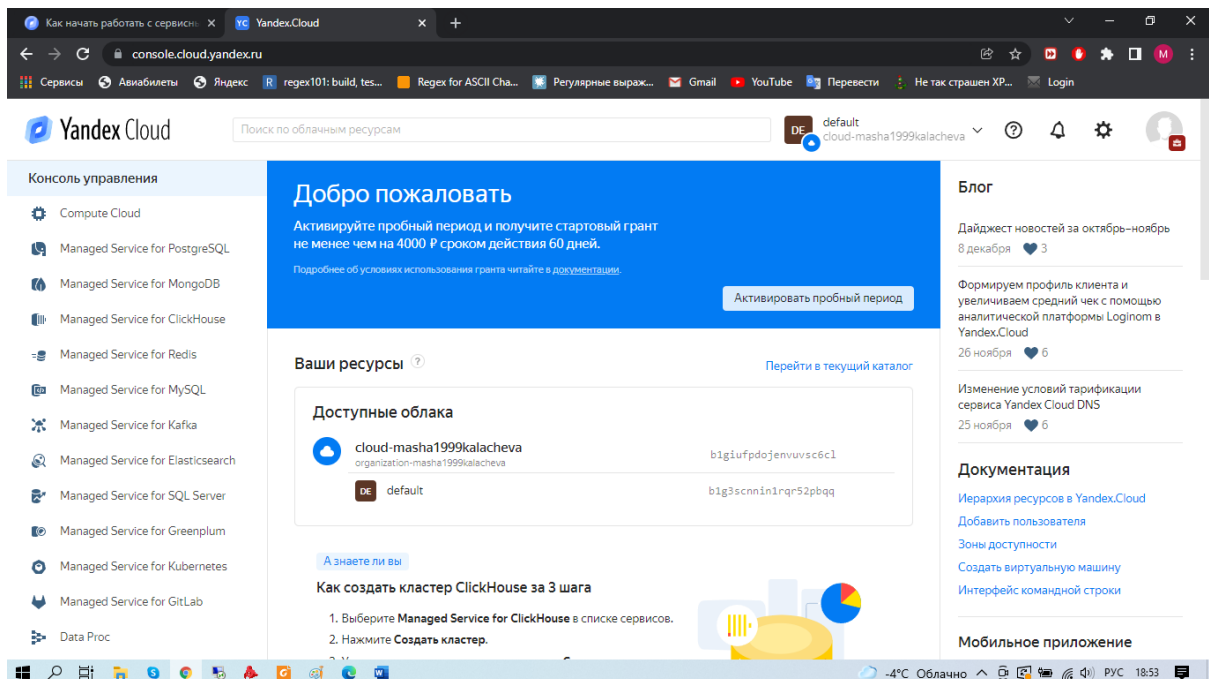


- Кликнуть по кнопке «Войти в аккаунт на Яндексе».
- Выполнить вход, заполнив необходимые поля и нажав на кнопку «Войти». Или зарегистрироваться и нажать на кнопку «Создать ID».
- Авторизоваться, заполнив все поля.
- Кликнуть по кнопке «Создать».
- Кликнуть по кнопке «Активировать пробный период».

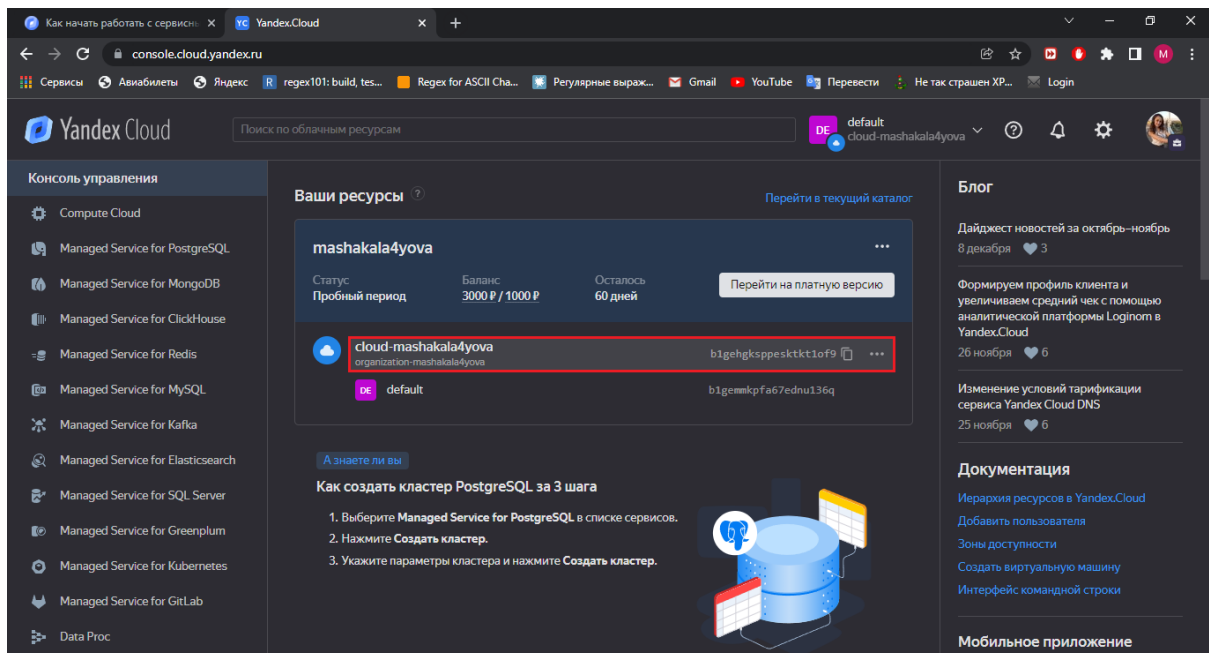




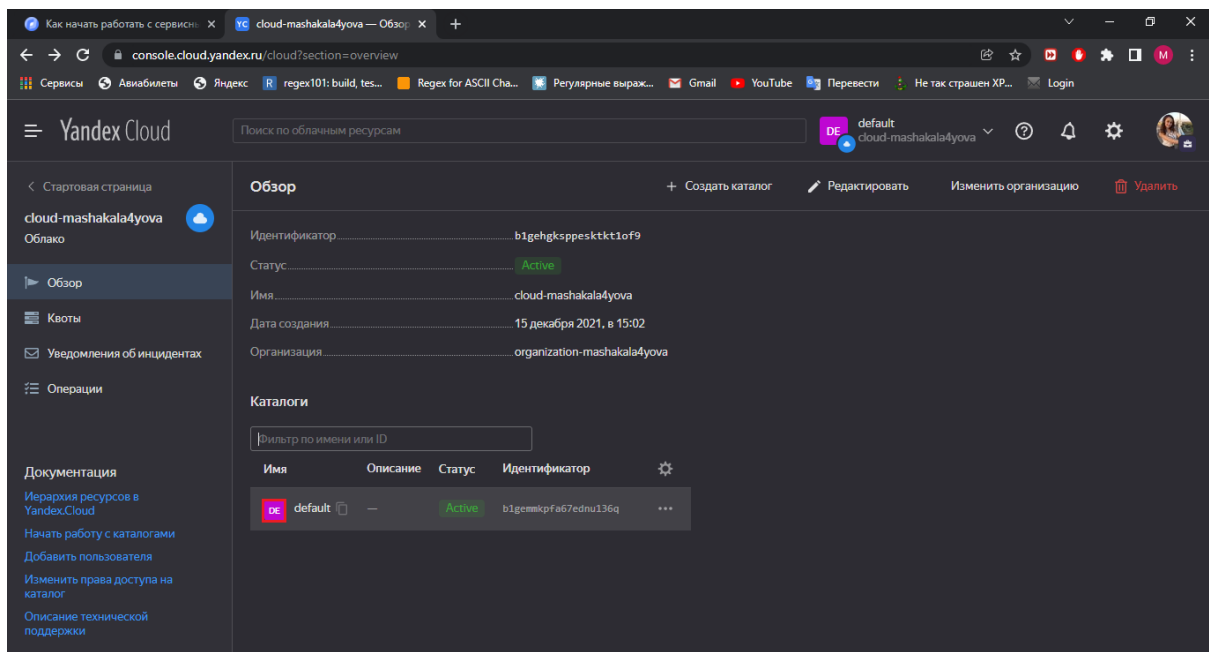




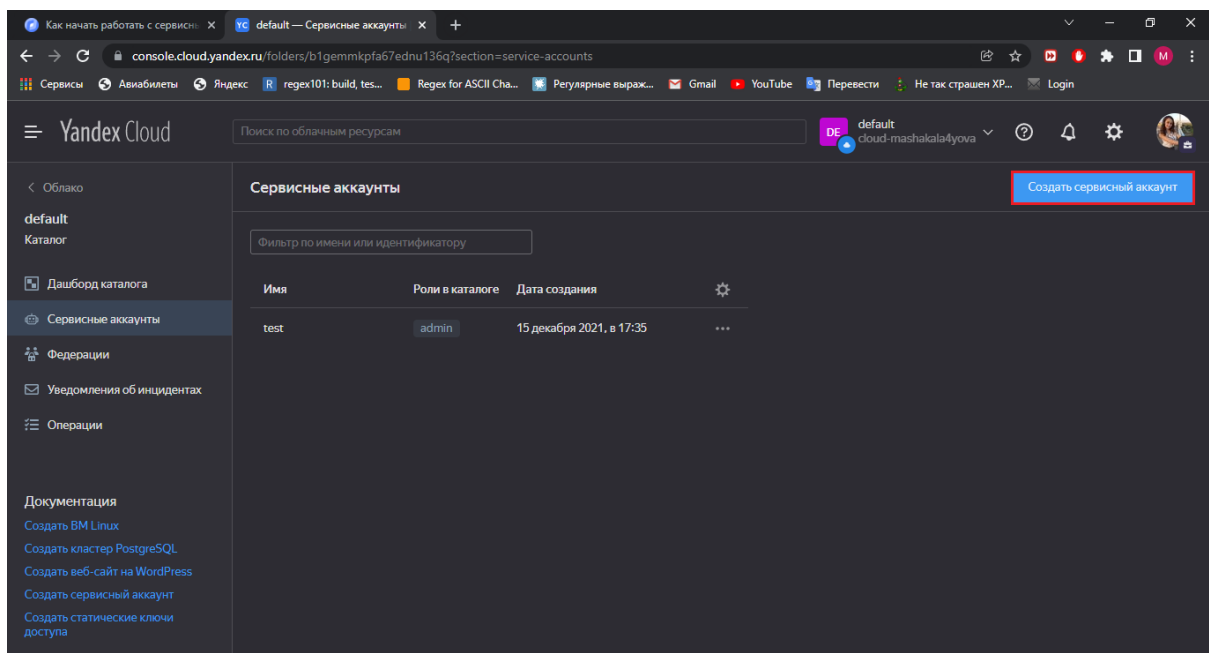
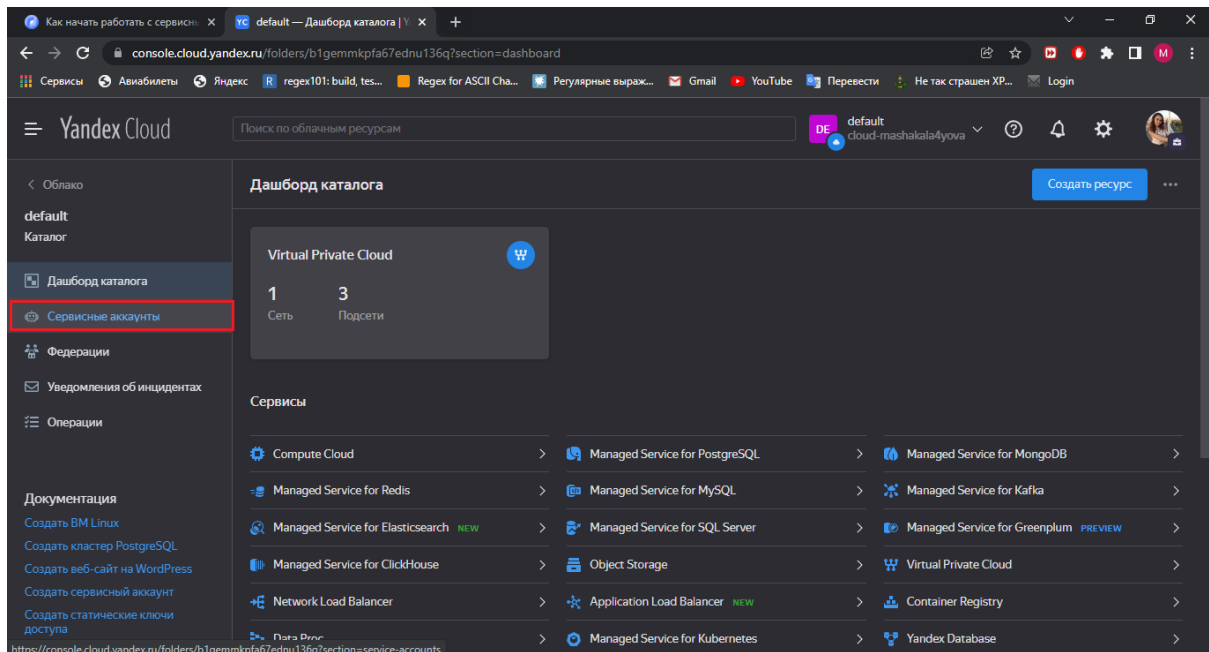
- Заполнить все поля и кликнуть по кнопке «Активировать».
- Нажать на строку с именем каталога, в котором нужно создать сервисный аккаунт.

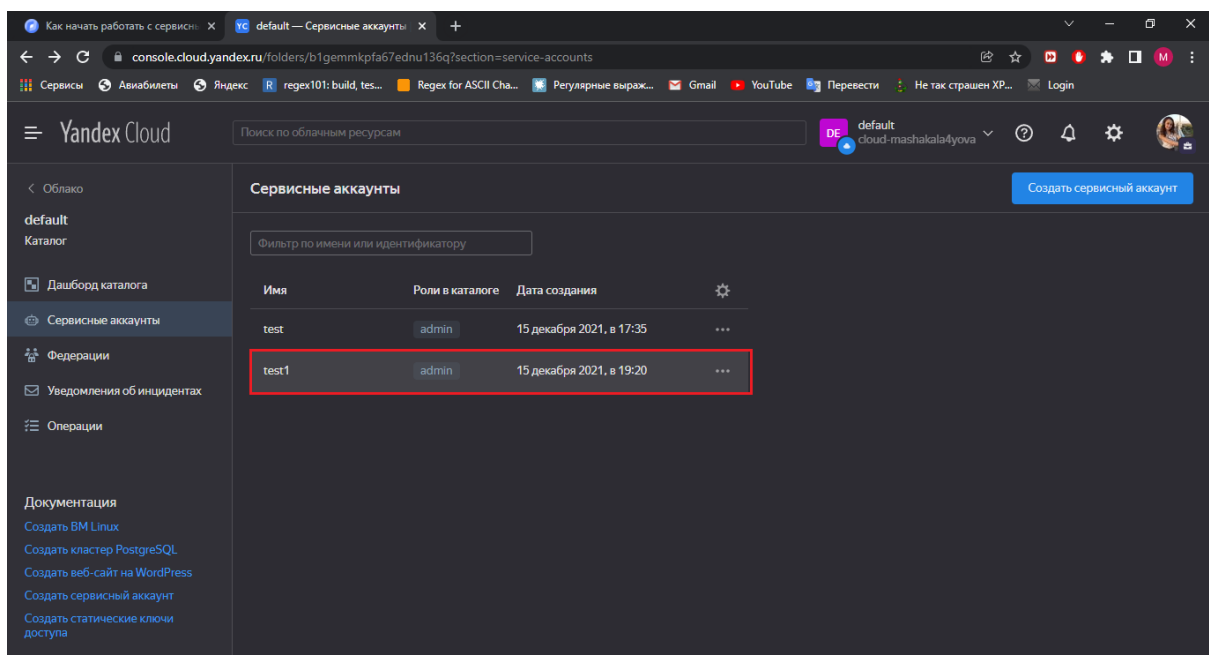
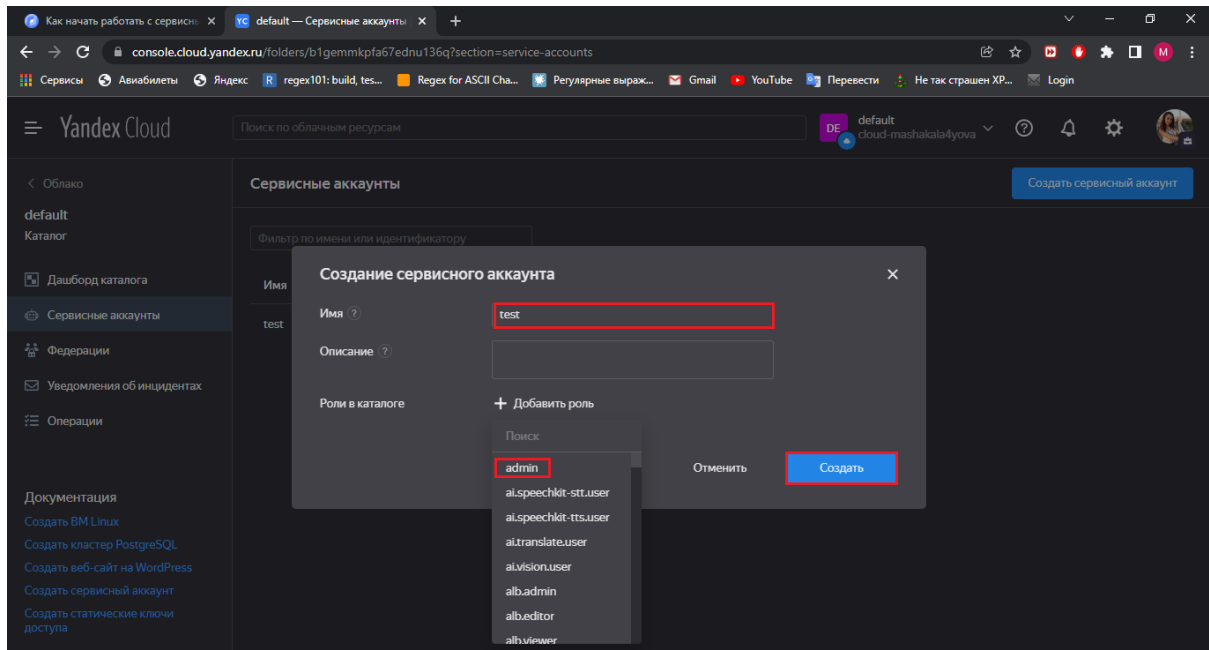


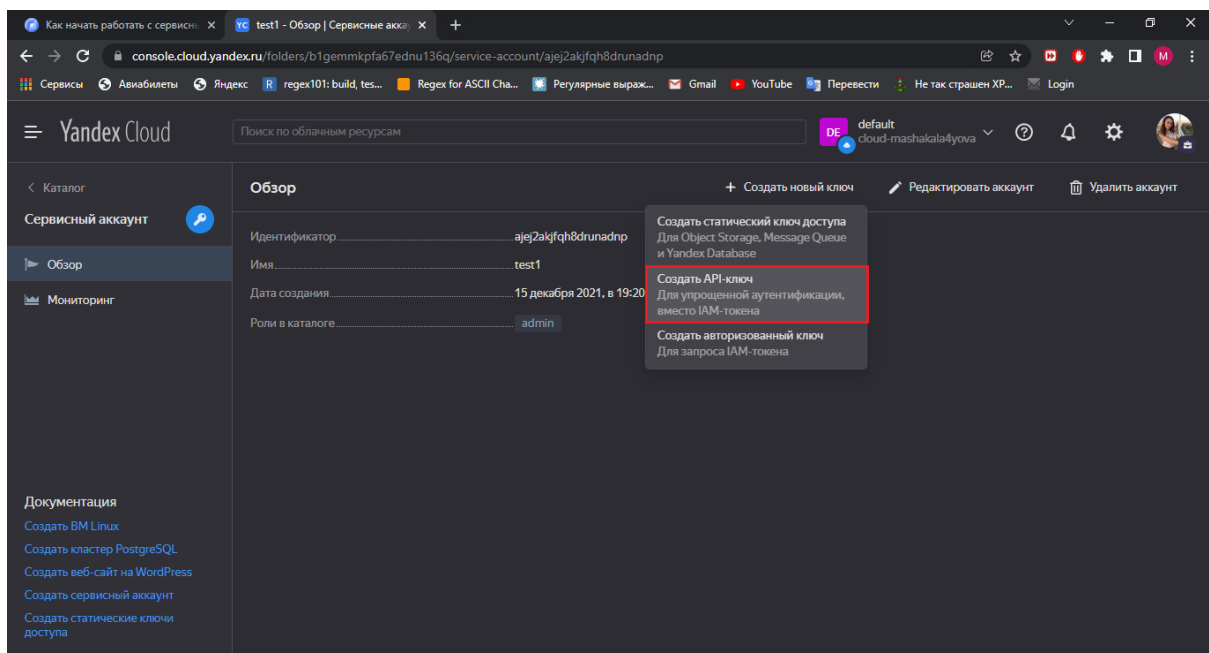
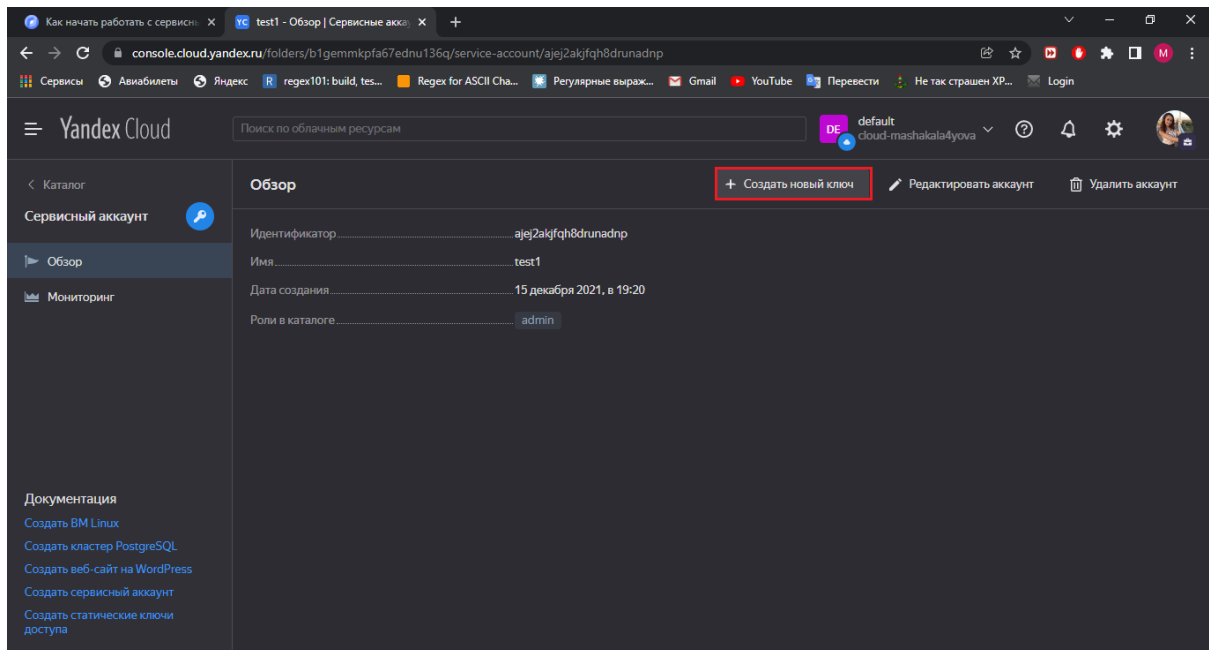
- Кликнуть по имени.



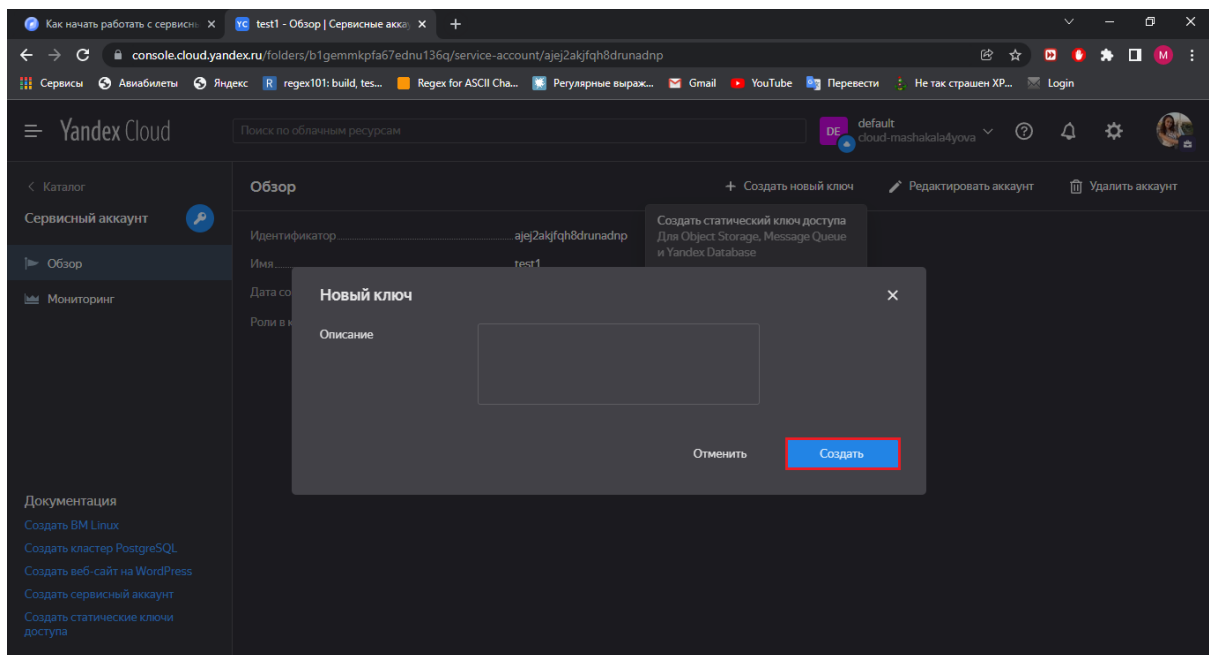
- Выбрать вкладку «Сервисные аккаунты».
- Нажать кнопку «Создать сервисный аккаунт».
- Ввести имя сервисного аккаунта, нажать на кнопку «Добавить роль» и выбрать роль, например admin. Кликнуть по кнопке «Создать».
- Кликнуть по созданному аккаунту.
- Кликнуть по кнопке «Создать новый ключ».
- Выбрать из списка «Создать API-ключ».



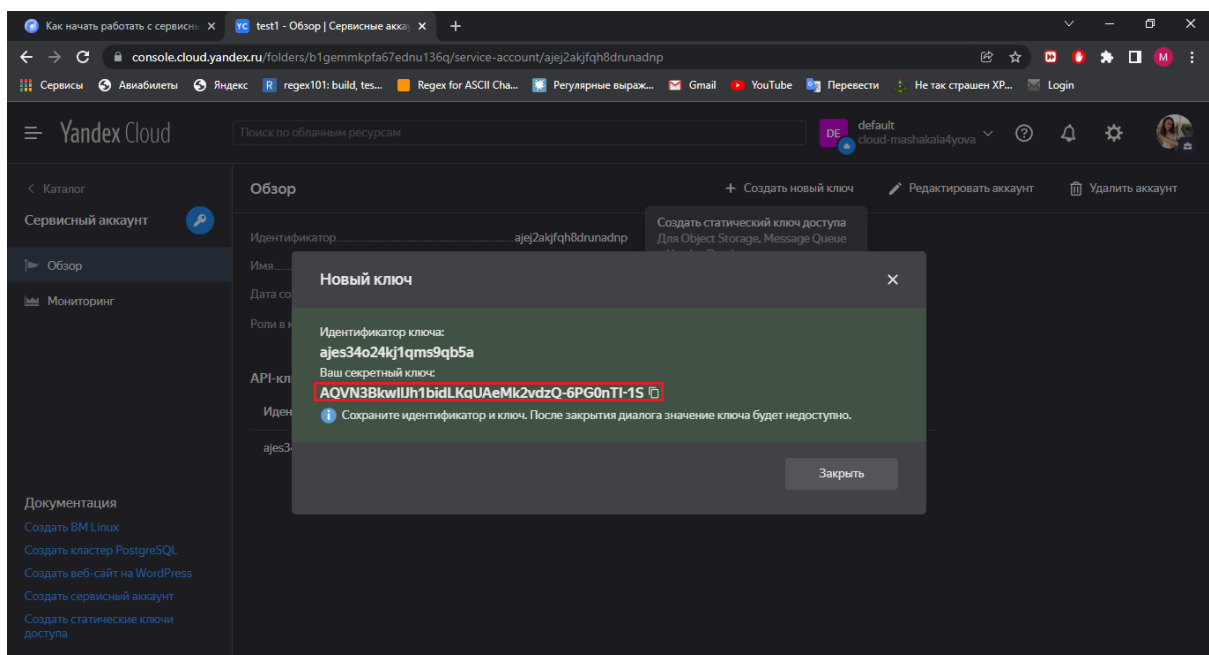




- Кликнуть по кнопке «Создать».



- Копировать секретный ключ.

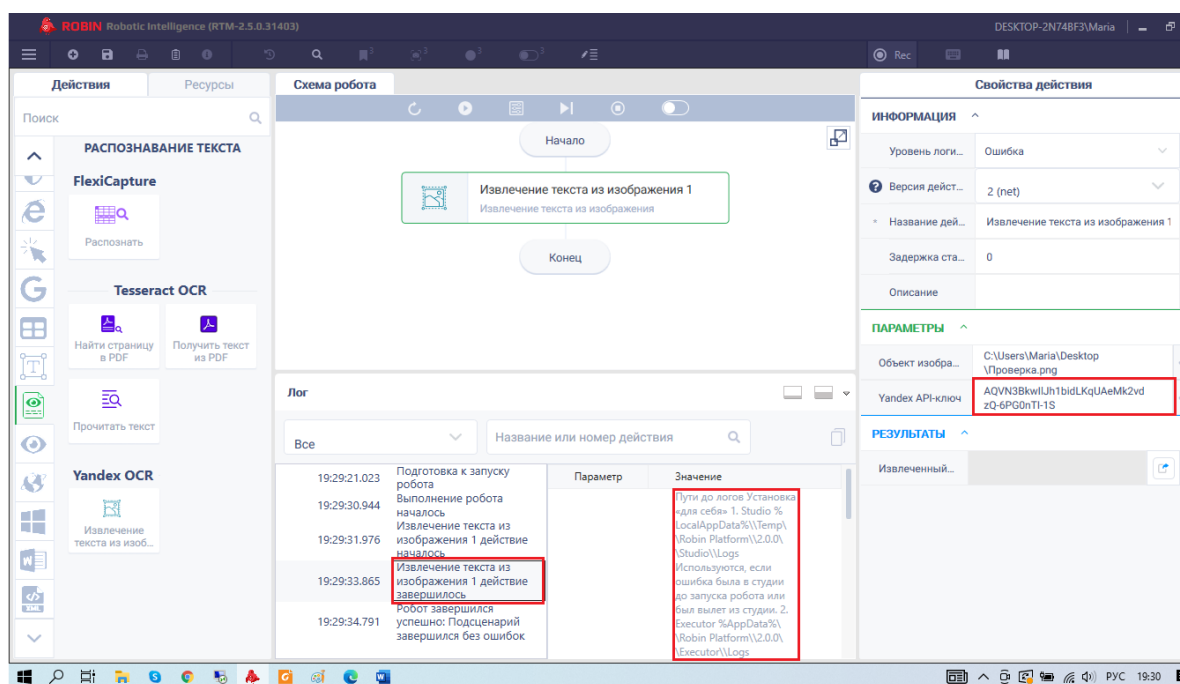


- Вставить скопированный ключ в параметр «Yandex API-ключ» и запустить робота.

## Подгруппа «Tesseract OCR» - Найти страницу в PDF

Для настройки действия «Найти страницу в PDF» необходимо выполнить следующие шаги:

1. Перенести действие «Найти страницу в PDF» на рабочую область.
2. Заполнить параметры действия:



1. Параметр «Путь к файлу PDF» - указать путь до PDF-файла, в котором будет выполнен поиск страницы;
2. Параметр «Текст» - указать текст, который должна содержать искомая страница;
3. Параметр «Ожидаемые языки в PDF файле» - выбрать из выпадающего списка язык, который ожидается на искомой странице.

Доступные языки:

- Русский язык;
- Английский язык;
- Русский и Английский язык;

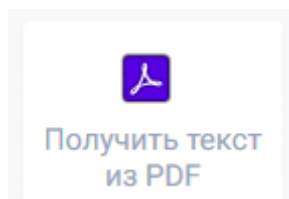
Пример заполненных параметров:

ПАРАМЕТРЫ	
* Путь к PDF файлу	C:\Users\persi\Downloads\Certificate.pdf
? Текст	Робот
* Ожидаемые языки текста в PDF файле	Русский язык

3. Результатом действия является коллекция, в которой хранятся номера страниц, на которых был найден искомый текст.

## Подгруппа «Tesseract OCR» - Получить текст из PDF

1. Перенести действие «Получить текст из PDF» на рабочую область.



2. Заполнить параметры действия:

1. Параметр «Путь к файлу PDF» - указать путь до PDF-файла, в котором будет выполнен поиск страницы;
2. Параметр «Язык текста» - выбрать из выпадающего списка язык, который ожидается на искомой странице.

Доступные языки:

- Русский язык;
- Английский язык;
- Русский и Английский язык;

Параметр «Номер страницы» - номер страницы файла, с которой будет считываться текст. Пример заполненных параметров:

ПАРАМЕТРЫ ^		
* Путь к файлу PDF	C:\Users\persi\Downloads\Certificate.pdf	...
* Язык текста	Русский язык	▼
* Номер страницы	15	...

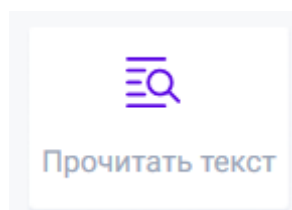
3. Результат действия – текст с указанной страницы PDF-файла. Если страница отсутствует в документе, будет возвращено пустое значение.

## Подгруппа «Tesseract OCR» - Прочитать текст

Для настройки действия «Прочитать текст», предназначенного для распознавания и чтения текста с изображения, необходимо выполнить следующие шаги:

1. Перенести действие «Прочитать текст» на рабочую область.
2. Заполнить параметры действия:
  1. Параметр «Изображение» - ссылка на изображение, с которой будет прочитан текст;





2. Параметр «Ожидаемые языки» - выбрать из выпадающего списка язык, который ожидается на искомой странице.

Доступные языки:

- Русский язык;
- Английский язык;
- Русский и Английский язык;

3. Параметр «Формат контента» - выбрать из выпадающего списка ожидаемый формат текста.

Доступные форматы:

- Строка;
- Блок;
- Страница.

Пример заполнения параметров:

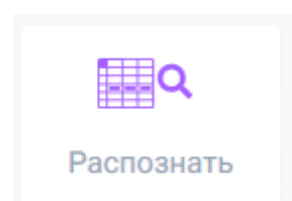
ПАРАМЕТРЫ ^		
* Изображение	C:\Users\persi\OneDrive\Documents \Yp5MSBm8xEg.jpg	...
* Ожидаемые языки...	Русский язык	▼
* Формат контента	Строка	▼

3. Результат действия – строка, содержащая текст, полученный из изображения.

### Подгруппа «FlexiCapture» - Распознать

Для настройки действия «Распознать» необходимо выполнить следующие шаги:

1. Перенести на рабочую область действие «Распознать».



2. Заполнить параметры действия:

1. Параметр «Графический файл» - указать путь до графического файла, из которого необходимо извлечь текст;
2. Параметр «Url к веб-серверу FlexiCapture» - указать ссылку для авторизации на веб-сервере FlexiCapture;
3. Параметр «Логин» - логин для авторизации на веб - сервере FlexiCapture;
4. Параметр «Пароль» - пароль для авторизации на веб - сервере FlexiCapture;
5. Параметр «ID роли подключения» - указать идентификатор роли оператора FlexiCapture;
6. Параметр «ID типа станции» - указать идентификатор типа станции FlexiCapture;
7. Параметр «Имя или Guid проекта» - указать имя или идентификатор проекта FlexiCapture;
8. Параметр «ID типа пакета» - указать идентификатор типа пакета FlexiCapture;
9. Параметр «ID пользователя или группы» - указать идентификатор владельца пакета FlexiCapture;
10. Параметр «ID стадии экспорта» - указать идентификатор стадии экспорта FlexiCapture;
11. Параметр «Формат вложения» - выбрать из выпадающего списка расширение файла, который будет возвращен с сервера.

Доступны следующие типы:

- xlsx;
  - xls;
  - xml;
  - json;
  - csv;
  - txt;
  - dbf.
12. Параметр «Папка для вложения» - указать путь до папки, куда будет сохранено вложение полученное с сервера.

Пример заполнения полей:

3. Результат действия – словарь с результатом распознавания в формате ключ-значение и путь до файла с результатом распознавания.

### 2.14.3 REST и SOAP запросы

В данном описании рассмотрены способы заполнения параметров действий REST и SOAP для понимания работы с запросами при помощи Robin Robot, включая общую информацию по данным запросам.

Также краткое описание к каждому действию есть в студии, оно расположено на самом значке действия во всплывающем значке вопроса, рассмотреть его можно нажав на знак вопроса, а затем перейдя по подсказке «Подробнее».

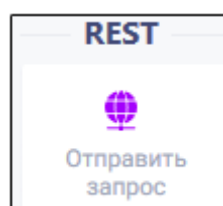
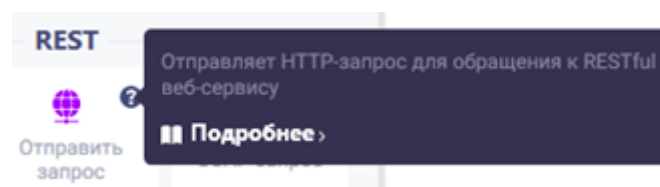
#### Отправить запрос / Send request

Группа действий: Интеграции

Подгруппа: REST

**ПАРАМЕТРЫ**

Графический файл	C:\Users\persi\OneDrive\Documents\Yp5MSBm8xEg.jpg	...
Url к веб-сервису FI...	http://172.28.4.19/FlexiCapture12/Server/FCAuth/API/v1/WSDL	...
Логин	LOGIN	...
Пароль	*****	...
ID роли подключен...	5	...
ID типа станции	4	...
Имя или Guid прое...	4ad76228-517f-4496-aa6c-eef36d3de35a	...
ID типа пакета	5	...
ID пользователя и...	4	...
ID стадии экспорта	7	...
Формат вложения	txt	▼
Папка для вложен...	C:\Users\persi\OneDrive\Рабочий стол\Документы	...



Действие «Отправить запрос» – предназначено для отправки HTTP-запроса для обращения к RESTful веб-сервису.

REST-запрос – это архитектурный стиль создания web-сервиса, когда на имеющиеся URL можно отправлять запросы для получения информации.

В таблице ниже приведено описание параметров действия.

Параметры действия	Пример заполнения	Описание параметров
<b>URL ресурса</b>	<div> <div>URL REST-сервиса</div> <div>https://reqres.in/api/users</div> <div>...</div> </div>	<p>Строка, в которую вписан URL, на который отправляется запрос.</p> <p>Пример запроса: <a href="https://reqres.in/api/users?v=jTQtyllwHsg&amp;name=124546">https://reqres.in/api/users?v=jTQtyllwHsg&amp;name=124546</a></p> <ul style="list-style-type: none"> <li>• Протокол должен быть указан!</li> </ul> <p><b>HTTPS</b> можно использовать с методами: GET, POST, PUT, DELETE, HEAD, OPTIONS.</p> <p><b>HTTP</b> можно использовать только с методами: GET, HEAD.</p> <p>Если протокол не будет указан, то по умолчанию будет использован HTTPS!</p> <p>Если протокол указан, то код не меняет его значение.</p> <ul style="list-style-type: none"> <li>• <a href="https://reqres.in/">//reqres.in/</a> - адрес уникального ресурса в интернете.</li> <li>• В URL ставится «/api/» перед запросом.</li> <li>• Перед перечислением параметров, ставится символ «?».</li> <li>• Параметры между собой разделяются «&amp;».</li> </ul>
<b>Метод</b>	<div> <div>Метод</div> <div>GET</div> <div>^</div> </div> <div> <div>Тело запроса</div> <div>GET</div> <div>POST</div> <div>PUT</div> <div>DELETE</div> <div>HEAD</div> <div>OPTIONS</div> </div> <div> <div>Параметры запр...</div> <div>DELETE</div> </div> <div> <div>Параметры фор...</div> <div>HEAD</div> </div>	<p>Определяет тип создаваемого запроса.</p> <p>По умолчанию ставится метод GET.</p> <p>GET— метод для чтения данных с сайта. Передача параметров может осуществляться в URL.</p> <p>POST — метод для отправки данных на сайт. Чаще всего с помощью метода POST передаются формы.</p> <p>PUT – заменяет все текущие представления ресурса данными запроса.</p> <p>DELETE – удаляет указанный ресурс.</p> <p>HEAD - запрашивает ресурс так же, как и метод GET, но без тела ответа.</p> <p>OPTIONS - используется для описания параметров соединения с ресурсом.</p>

Описание параметров: тело запроса, параметры запроса, параметры формы, авторизация, заголовки запроса.

Описание параметра «прикрепить файл».

## Запрос / Request

Группа действий: Интеграции

Подгруппа: SOAP

SOAP (HTTP-запрос для обращения к веб-сервису (SOAP)).

Тело запроса	<div>Тело запроса</div> <pre>{   "email": "eve.holt@reqres.in",   "password": "cityslicka" }</pre>	<p>Позволяет отправить JSON запрос в виде строки.</p> <p>Тело отправляемого запроса для метода POST.</p>
	<pre>{   "email": "eve.holt@reqres.in",   "password": "cityslicka" }</pre>	
Параметры запроса	<div>Параметры запроса</div> <pre>gtins : 04600000000000 includeSubaccount : true</pre>	<p>Данный способ используется только для метода GET. Параметры можно передавать при помощи JSON, XML, и т.д.</p>
	<div> <div>Ключи</div> <div>gtins</div> <div>includeSubaccount</div> </div> <div> <div>Тип</div> <div>Строка</div> <div>Значение</div> <div>true</div> </div>	
Параметры формы	<div>Параметры формы</div> <pre>"uit" : "string"</pre>	<p>Данный способ используется только для метода POST. Здесь параметры задаются в HTML формах.</p>
	<div> <div>Ключи</div> <div>"uit"</div> </div> <div> <div>Тип</div> <div>Строка</div> <div>Значение</div> <div>"string"</div> </div>	
Авторизация	<div>Авторизация</div> <pre>Authorization : Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...</pre>	<p>Логин и пароль для базовой аутентификации или JSON Web Token (JWT) для авторизации.</p> <p>Заголовок авторизации будет автоматически сгенерирован при отправке запроса.</p> <p>Токен задается через словарь.</p>
	<div> <div>Ключи</div> <div>Authorization</div> </div> <div> <div>Тип</div> <div>Строка</div> <div>Значение</div> <div>Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...</div> </div>	
Заголовки запроса	<div>Заголовки запроса</div> <pre>content-type : application/json</pre>	<p>Набор заголовков отправляемого запроса и их значений.</p> <p>Каждый ключ – отдельный заголовок.</p> <p>Используется для метода POST.</p>

Прикрепить файл	<div>Путь к файлу для отправки методом POST</div> <div>K:\\bookModel.txt</div>	<p>Путь к файлу.</p> <p>К запросам прикрепляется файл, для метода POST.</p>
-----------------	--	---



SOAP – это простой протокол, который используется для обмена произвольными сообщениями в формате XML, а также для реализации удалённого вызова процедур.

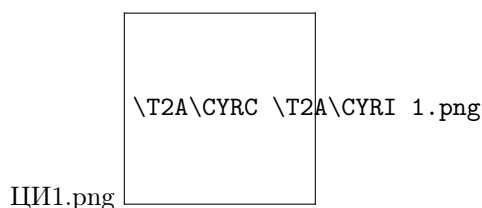
В таблице ниже приведено описание параметров действия.

Параметры действия	Пример заполнения	Описание параметров
URL-адрес:	<div> <div>Url-адрес</div> <div>https://api.direct.yandex.ru/live/v4/wSDL/</div> <div>...</div> </div>	Строка, в которую вписан URL, на который отправляется запрос.
Заголовки:	<div> <div>Заголовки</div> <div>SOAPAction : checkText</div> <div>...</div> <div> <div>Ключи</div> <div> <div>SOAPAction</div> <div>Тип</div> <div>Строка</div> <div>Значение</div> <div>checkText</div> </div> </div> </div>	Словарь, заполняется набором заголовков для запроса и их значениями.
XML:	<div> <div>XML</div> <div>Открыть файл XML</div> <div>Контекст XML-документа</div> <div>...</div> </div>	Отправляемый запрос, отправляется в виде XML файла.

## 2.14.4 Циклы

Циклы необходимы для повторения определенной последовательности действий нужное количество раз.

В ROBIN Studio 2.0 представлено четыре вида циклов в группе действий «Базовые действия»: «Для», «Для каждого», «Пока» и «Цикл с постусловием», а так же действия «Прервать цикл» и «Продолжить цикл» для остановки работы цикла и перехода к началу его выполнения.



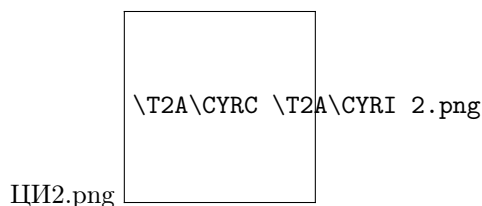
Рассмотрим порядок работы с каждым циклом.

### Цикл «Для»

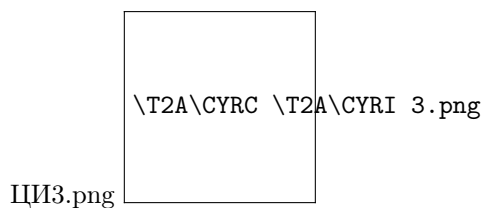
Данное действие предназначено для обеспечения заданной итерации по счетчику шагов, т.е. цикл выполнит действия внутри него заданное количество раз.

*Порядок работы с действием цикл «Для»:*

1. Выбрать действие «Для» из группы действий «Базовые действия» и перенести в сценарий;



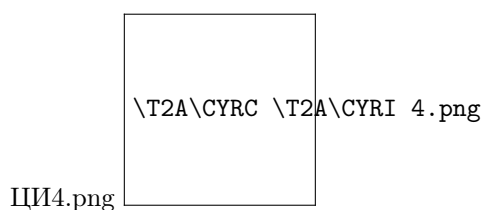
2. Заполнить тело цикла необходимыми действиями; Пример заполнения:



3. Заполнить параметры действия:

- Начальное значение (значение, с которого начнется отсчет);
- Конечное значение (значение до которого будет вестись отсчет.);
- Инкремент (число, на которое будет увеличиваться значение итератора);

Пример заполнения:



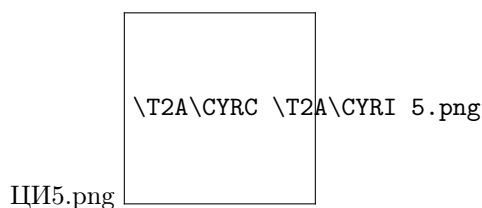
4. Результат работы цикла: состояние итератора, т.е. значение счетчика для использования в текущей итерации цикла.

### Цикл «Для каждого»

Данный цикл предназначен для последовательной работы с каждым элементом выбранной коллекции.

*Порядок работы с циклом «Для каждого»:*

1. Выбрать действие «Для каждого» из группы действий «Базовые действия» и перенести в сценарий;



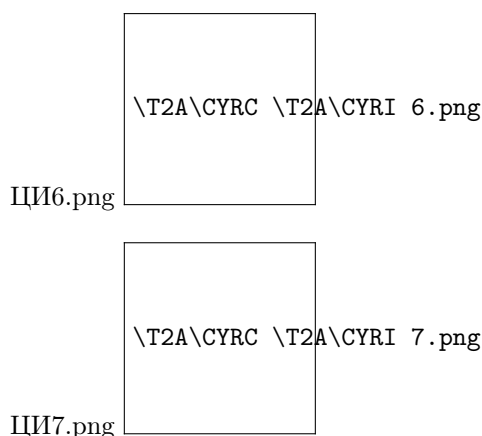
2. Заполнить тело цикла необходимыми действиями; Пример заполнения:

3. Заполнить параметр действия:

- Коллекция (список значений, которые будут обработаны в цикле)

Пример заполнения параметра:

4. Результат работы цикла: объект, в который будет помещен каждый элемент коллекции во время выполнения итерации.



### Сравнение циклов «Для» и «Для каждого»

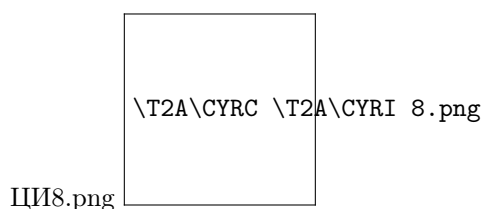
Цикл «Для» выполнит работу заданное количество раз. Итератор будет принимать значения от начального заданного значения до конечного с шагом, указанным в поле «Инкремент». Цикл «Для каждого» последовательно работает с элементами выбранной коллекции. Количество итераций равно количеству элементов в коллекции, выбранной на входе. Таким образом результат цикла «Для» это число, а «Для каждого» объект.

### Цикл «Пока»

Данное действие предназначено для создания цикла действий, который выполняется до тех пор, пока результат условия равен «True».

*Порядок работы с циклом «Пока»:*

1. Выбрать действие «Пока» из группы действий «Базовые действия» и перенести в сценарий.



2. Заполнить тело цикла (блок внутри цикла, где прописывается алгоритм для работы робота) необходимыми действиями; Пример заполнения:

3. Заполнить параметр действия:

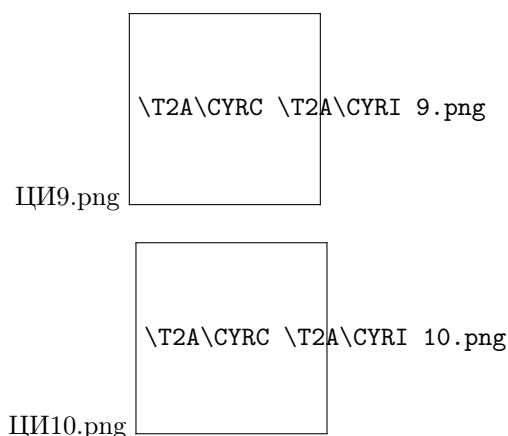
- Условие (критерий, по которому будет приниматься решение о продолжении или прекращении работы цикла).

Пример заполнения:

Если условие заполняется посредством инструмента построения выражений, то оно отобразиться в разделе «Условие» внутри цикла. Инструмент построения выражений:

При заполнении чек-бокса условие цикла считается равным True. Если чек-бокс не заполнен и не указано иное условие, то условие цикла считается равным False.





Пример цикла с условием, заполненным с помощью инструмента построения выражений:

### «Цикл с постусловием»

Данное действие предназначено для создания цикла действий, который выполняется до тех пор, пока результат условия равен «True»

Порядок работы с «Циклом с постусловием»:

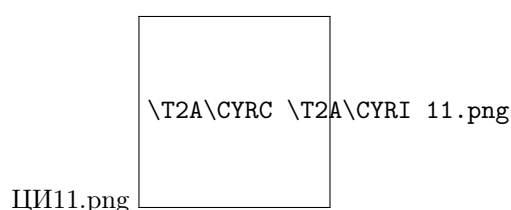
1. Выбрать действие «Цикл с постусловием» из группы действий «Базовые действия» и перенести в сценарий.
2. Заполнить тело цикла необходимыми действиями. Пример заполнения:
3. Заполнить параметр действия.
  - Условие

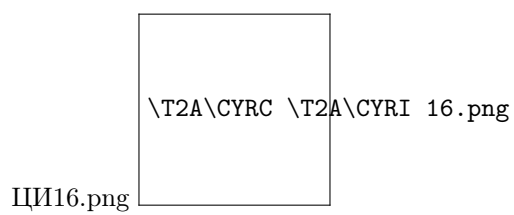
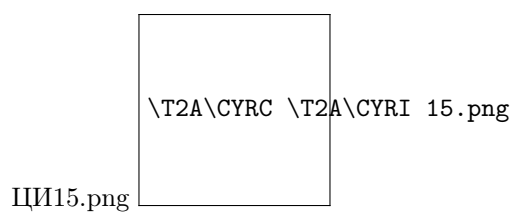
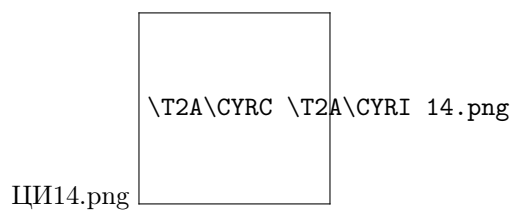
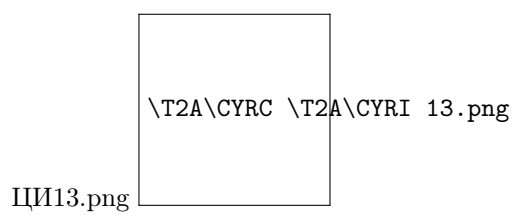
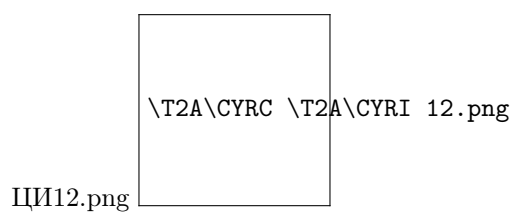
Пример заполнения:

Если условие заполняется посредством инструмента построения выражений, то оно отобразится в разделе «Условие» внутри цикла. Пример цикла с постусловием, где условие заполнено с помощью инструмента построения выражений:

### Сравнение циклов «Пока» и цикла с «Постусловием»

Оба цикла работают до тех пор, пока результат условия равен «True». Однако в цикле «Пока» проверка действия происходит перед началом выполнения цикла, т.е. цикл может отработать 0 раз и более. Проверка условия в действии «Цикл с постусловием» происходит после выполнения действий в теле цикла, т.е. цикл отработает минимум один раз.



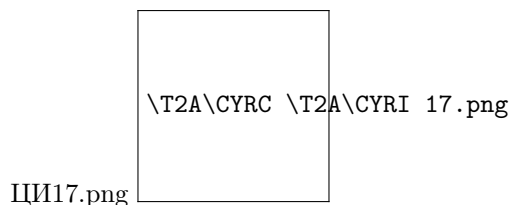


## Действия «Прервать цикл» и «Продолжить цикл»

Действия «Прервать цикла» и «Продолжить цикл» находятся в группе действий «Базовые действия» в подгруппе «Циклы».

*Действие «Прервать цикл»*

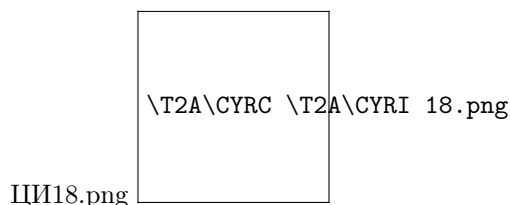
Выбрать действие «Прервать цикл» и перенести в сценарий.



На этом этапе сценария робот завершит работу с циклом и продолжит выполнять действий вне цикла.

*Действие «Продолжить цикл»*

Выбрать действие «Продолжить цикл» и перенести в сценарий.



На этом этапе сценария робот завершит работу с циклом и начнет выполнять действия в теле цикла сначала.

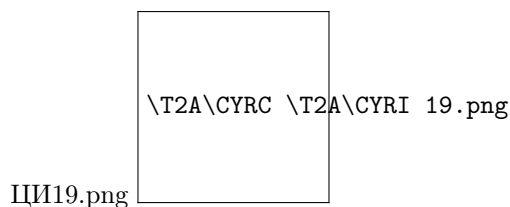
Для более точного понимания работы циклов в Robin Studio рассмотрим небольшие кейсы.

### Кейс «Нумерация строк». Цикл «Для»

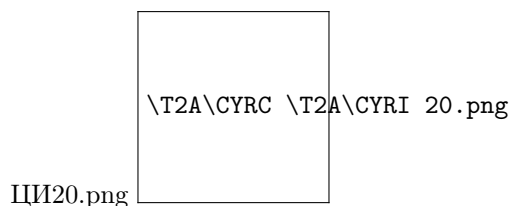
*Задача:* пронумеровать строки в excel-файле.

*Решение:*

1. Создать переменную «Номер». Тип переменной «Число».

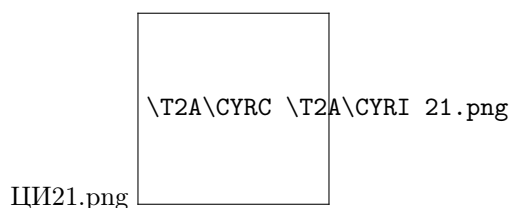


2. Перенести на рабочую область действие «Открыть» из группы действий «Excel».
3. Перенести на рабочую область действие «Для» (цикл) из группы действий «Базовые действия».
4. Перенести на рабочую область действия «Сохранить» из группы действий «Excel».
5. Перенести на рабочую область действия «Закрыть» из группы действий «Excel».



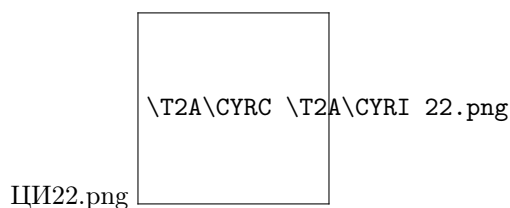
6. Заполнить параметры действий, перенесенных на рабочую область:

а. Действие «Открыть»: В строке «Путь к файлу» указать путь к excel-файлу в котором будет проведена работа.



б. Действие «Для»: «Начальное значение», «Конечное значение» и «Инкремент». В параметр «Результат» записать созданную ранее переменную «Номер».

Пример заполнения параметров:



с. Для действия «Сохранить».

В строке «Контекст» указать экземпляр открытого excel-файла.

д. Действие «Закрыть». В строке «Контекст» указать экземпляр открытого excel-файла.

7. В тело цикла поместить действия «Конвертировать данные» из группы «Базовые действия», «Соединить» из группы «Текст» и «Установить значение ячейки» из группы действий «Excel».

8. Заполнить параметры выбранных действий в теле цикла:

1. Действие «Конвертировать данные»

- В строке «Источник» выбрать переменную «Номер».
- В строке «Тип» выбрать значение «Строка» из выпадающего цикла.

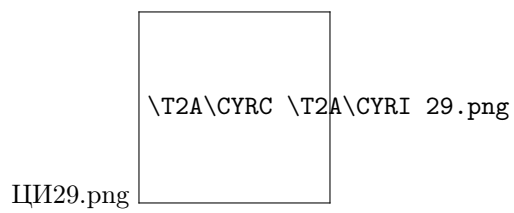
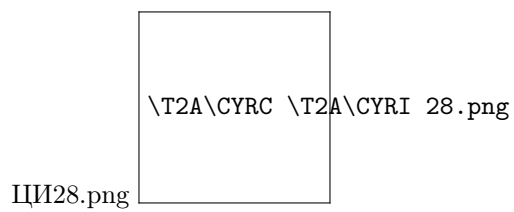
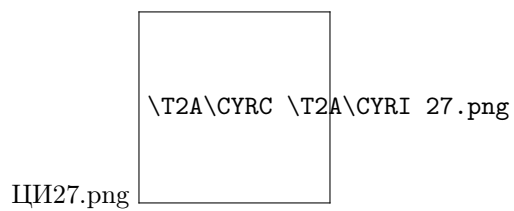
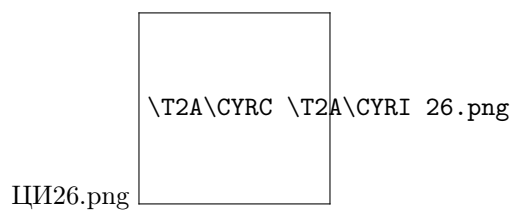
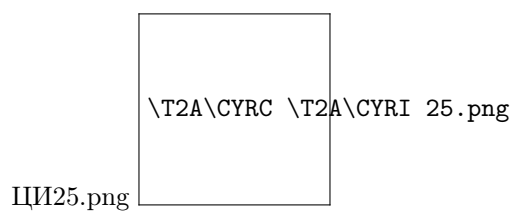
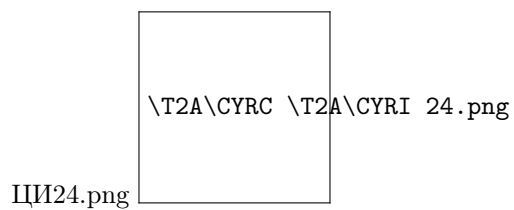
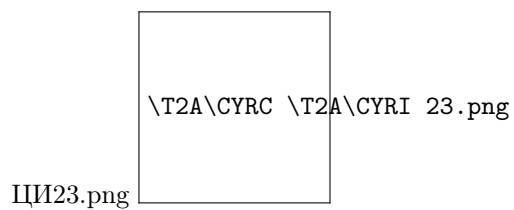
2. Действие «Соединить».

- «Текст 1» - название столбца, который будет пронумерован.
- «Текст 2» - результат действия «Конвертировать данные».

с. Действие «Установить значение ячейки».

- В строке «Контекст» указать экземпляр открытого excel-файла.
- В строке «Ячейка» указать результат действия «Соединить».
- В строке «Значение» указать переменную «Номер».

9. Запустить работа по кнопке «Запуск» в верхней панели.



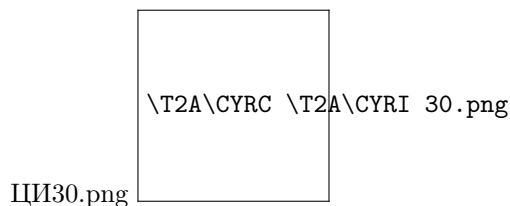
*Результат:* алгоритм прошел по циклу, ячейки в выбранном столбце excel-файла были пронумерованы.

### Кейс «Сортировка». Цикл «Для каждого»

*Задача:* В папке находятся файлы с расширением .xls(x) и .docx. Необходимо создать папки с названием расширения и распределить файлы по папкам.

*Решение:*

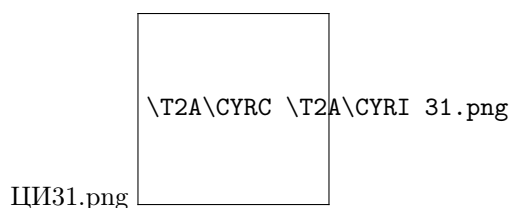
1. На рабочую область перенести действия «Получить содержимое папки» и «Для каждого».



2. Заполнить параметры действий.

а. Действие «Получить содержимое папки».

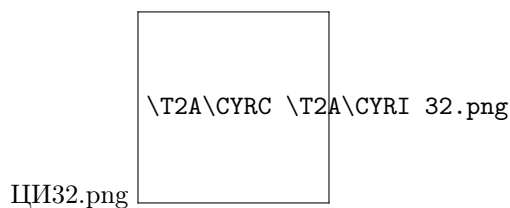
- В строке «Папка» указать путь до папки с файлами, которые необходимо рассортировать.
- В поле «Тип объектов» указать значение «только файлы».



2. Действие «Для каждого».

- В строке «Коллекция» указать результат выполнения действия «Получить содержимое папки»

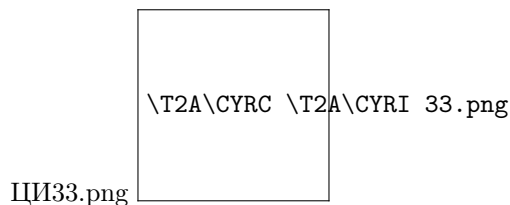
Пример:



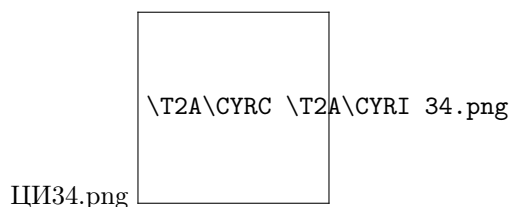
3. В тело цикла поместить следующие действия в указанном порядке:

1. «Получить информацию о файле»;
2. «Извлечь с позиции»;
3. «Найти папку»;
4. «Получить длину»;
5. «Если»
6. На ветвь «Истина» поставить действия:
  1. «Получить значение по индексу»;
  2. Конвертировать данные;

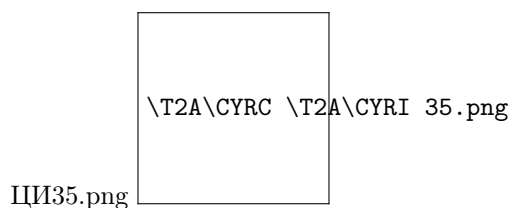
3. «Переместить файл»;
7. На ветвь «Ложь» установить действия:
  1. «Создать новую папку»;
  2. «Переместить файл»;



4. Заполнить параметры действий.
  1. Для действия «Получить информацию о файле»:
    - Параметр «Файл» - ссылка на объект, обрабатываемый в момент итерации цикла.
    - Параметр «Свойство» - «Тип».



2. Для действия «Извлечь с позиции»;
  - Параметр «Исходный текст» - результат действия «Получить информацию о файле».
  - Параметр «Позиция» - 1.
  - Параметр «Длина» - если данный параметр не заполнен, то текст извлекается с заданной позицией до конца.



с. Для действия «Найти папку».

- Параметр «Папка для поиска» - путь до папки, где храниться подпапка для сортировки файлов.
- Параметр «Шаблон поиска» - результат действия «Извлечь с позиции».

d. Для действия «Получить длину». Параметр «Коллекция или массив» - результат действия «Найти папку».

e. Для действия «Если». С помощью инструмента построения выражения в параметр условия записать логический оператор «Больше чем», где первый операнд – результат действия «Получить длину», а второй – 0.

6. Для действия «Получить значение по индексу».
  - Параметр «Коллекция» - результат действия «Найти папку».
  - Параметр «Индекс» - 0.



7. Для действия «Конвертировать данные».

- Параметр «Источник» - результат действия «Путь к папке».
- Параметр «Тип» - путь к папке.

8. Для действия «Переместить файл».

- Параметр «Файл» - ссылка на объект, обрабатываемый в момент итерации цикла.
- Параметр «Папка назначения» - результат действия «Конвертировать данные».

10. Для действия «Создать новую папку».

- Параметр «Родительская папка» - путь до папки, где , где храниться подпапка для сортировки файлов.
- Параметр «Имя» - результат действия «Извлечь с позиции».

11. Для действия «Переместить файл».

- Параметр «Файл» - ссылка на объект, обрабатываемый в момент итерации цикла.
- Параметр «Папка назначения» - результат действия «Создать новую папку».

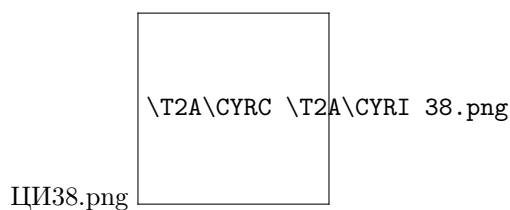
*Результат:* Алгоритм прошел по циклу, рассортировывая файлы по папкам.

### Кейс «Первое вхождение». Цикл «Пока»

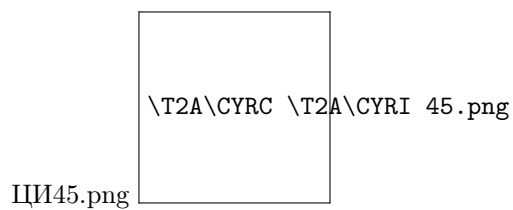
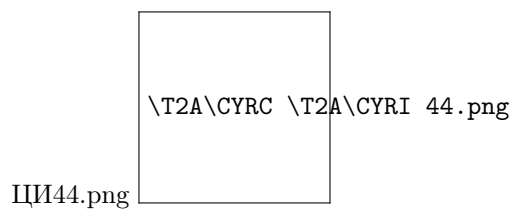
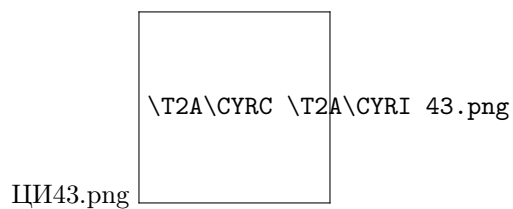
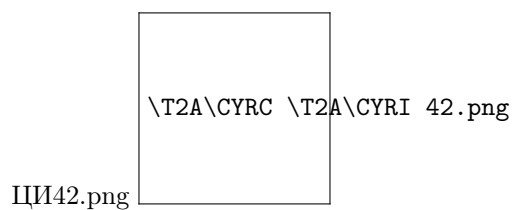
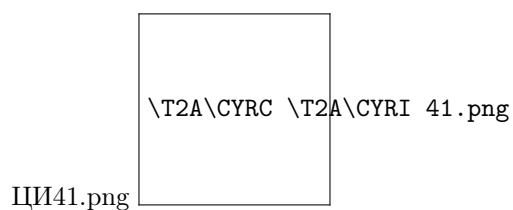
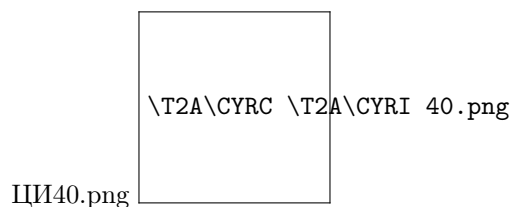
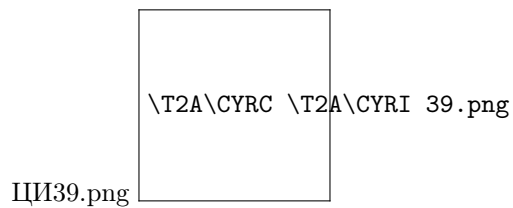
*Задача:* В excel-файле записаны дата и температура за это число. Необходимо найти первый день в списке, когда температура стала отрицательной.

*Решение:*

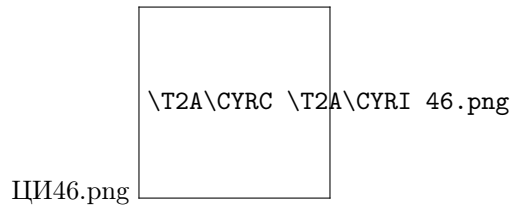
1. Создать переменные «Счетчик» и «Значение» тип переменных – число.
2. На рабочую область перенести действия «Открыть» и «Получить данные из столбца» (группа «Excel»), действие «Пока» (группа «Базовые действия»), «Сохранить» и «Закрыть» (группа «Excel»).
3. Заполнить параметры действий:



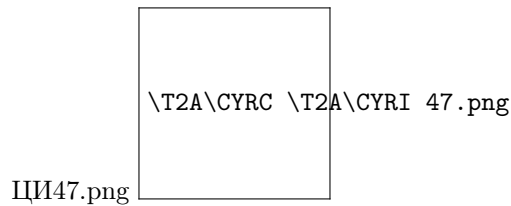




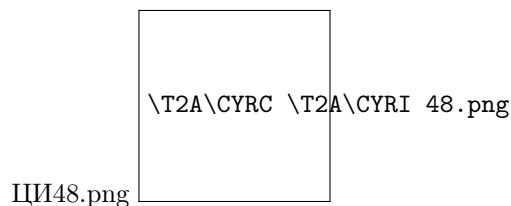
а. Действие «Открыть»: в строке «Путь к файлу» указать путь к excel-файлу с которым будет проведена работа.



б. Действие «Получить значение из столбца». Указать контекст и название столбца с температурными данными.



с. Действие «Пока». С помощью инструмента построения выражений записать условие «Переменная Значение больше 0». Пример:



д. Действие «Сохранить». В строке «Контекст» указать экземпляр открытого excel-файла.

е. Действие «Закрыть». В строке «Контекст» указать экземпляр открытого excel-файла.

4. В тело цикла добавить действие «Получить значение по индексу» (группа действий «Коллекции») и «Сложение» (группа действий «Операторы»).

5. Заполнить параметры этих действий:

а. Действие «Получить значение».

- В строке коллекция указать результат действия «Получить значения из столбца».
- В строке «Индекс» указать переменную «Счетчик».
- В параметре «Результат» записать переменную «Значение».

2. Действие «Сложение».

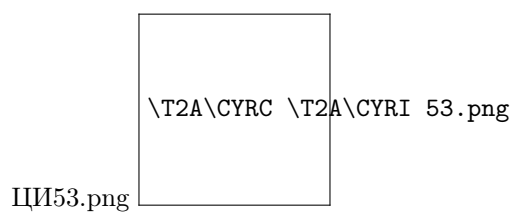
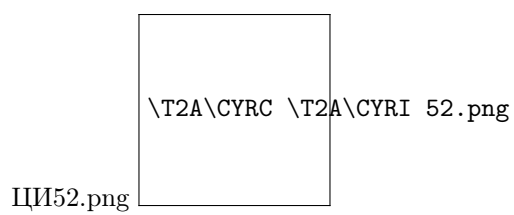
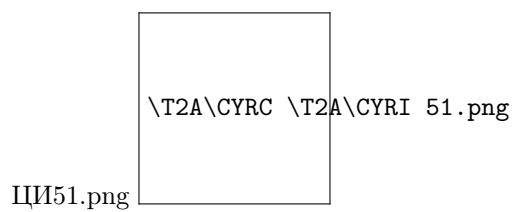
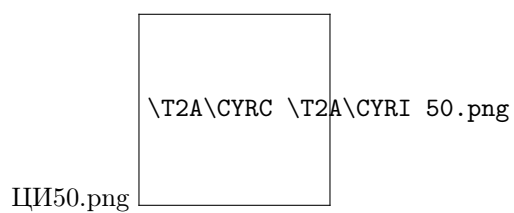
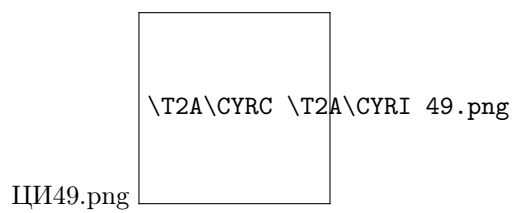
- Первый слагаемое – переменная «Счетчик».
- Второе слагаемое – 1.
- Результат – переменная «Счетчик».

*Результат:* алгоритм начнет перебирать значения коллекции в цикле и прекратит работу, как только встретит первое отрицательное число. Оно будет записано в переменную «Значение».

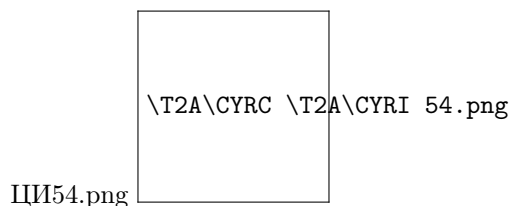
### Кейс «Поиск файла». «Цикл с постусловием»

*Задача:* В папке есть несколько подпапок, где лежит необходимый нам файл. Нужно просмотреть подпапки и выяснить в какой из них сохранен этот файл.

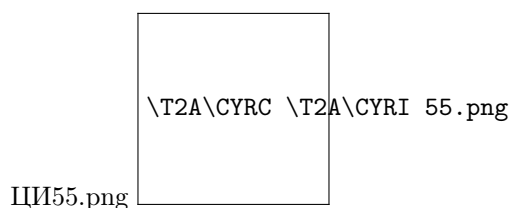
*Решение:*



1. Создать переменную «Счетчик», тип переменной – число.
2. Создать переменную «Значение», тип переменной – объект.



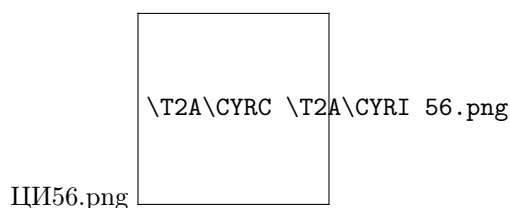
3. На рабочую область перенести действия «Получить содержимое папки» (группа действий «Файловая система»), «Цикл с постусловием» (группа действий «Базовые действия»).



4. Заполнить параметры действий:

а. Для действия «Получить содержимое папки»:

- В параметр «Папка» указать путь до папки с подпапками.
- В параметре «Тип объектов» выбрать из выпадающего списка значение «Только папки».



б. Для действия «Цикл с постусловием». В параметр условие с помощью инструмента построение выражений записать условие «Длина коллекции, полученной в результате действия «Найти файл» равно 0» Пример:

5. В тело цикла добавить действия «Получить значение по индексу» (группа действий «Коллекции»), «Конвертировать данные» (группа действий «Базовые действия»), «Найти файл» (группа действий «Файловая система») и «Сложение» (группа действий «Операторы»).

6. Заполнить параметры действий:

а. Для действия «Получить значение по индексу»:

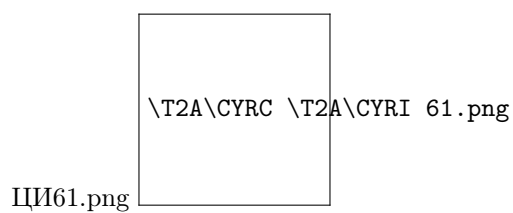
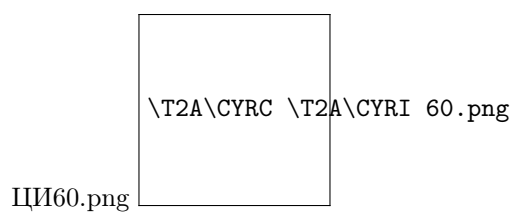
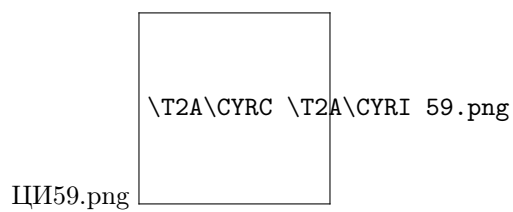
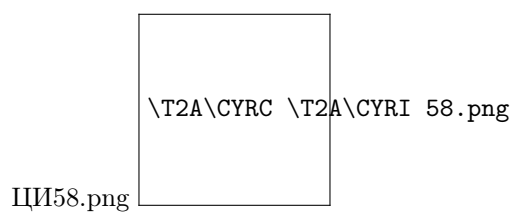
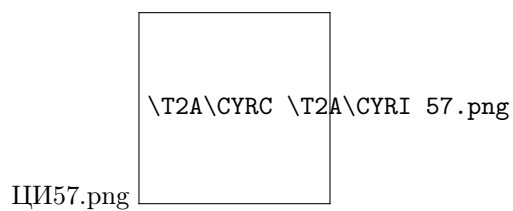
- Параметр «Коллекция» - результат действия «Получить содержимое папки».
- Параметр «Индекс» - переменная «Счетчик».
- Параметр «Результат» - переменная «Значение».

2. Для действия «Конвертировать данные»:

- Параметр «Источник» - результат действия «Получить значение по индексу».
- Параметр «Тип» - путь до папки.

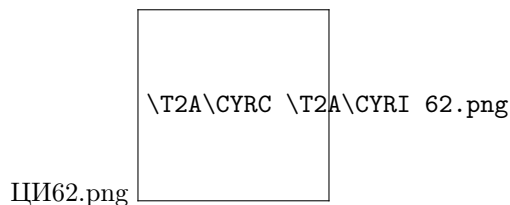
с. Для действия «Найти файл»:

- Параметр «Папка для поиска» - результат действия «Конвертировать данные».
- Параметр «Шаблон» - название искомого файла.



4. Для действия «Сложение»:

- Первое слагаемое – переменная «Счетчик».
- Второе слагаемое – 1.
- Результат – переменная «Счетчик».



*Результат:* алгоритм пройдет по циклу до тех пор, пока не будет найден искомый файл. Путь до папки с этим файлом будет сохранен в переменную «Значение».

### 2.14.5 Базы данных

Работу с базами данных робот осуществляет посредством действий группы «Базы данных». Группа действий содержит три подгруппы. Действия подгруппы «Запросы» позволяет обрабатывать базу данных, действия подгруппы «Общее» организует настройку подключения к базе данных и закрытие подключения с ней, действия подгруппы «Транзакции» позволяют закрыть и откатить транзакцию.

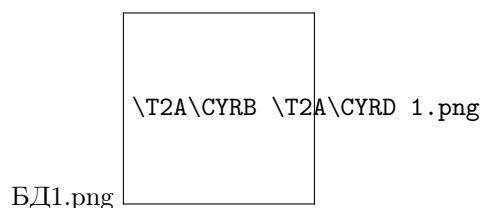
При создании сценария робота, работающего с базой данных, сначала необходимо настроить подключение к ней.

#### Подключение к базе данных

В Robin Studio существует два типа подключения к базе данных: пользовательское и стандартное.

##### Настройка стандартного подключения:

1. Выбрать действие «Стандартное подключение».



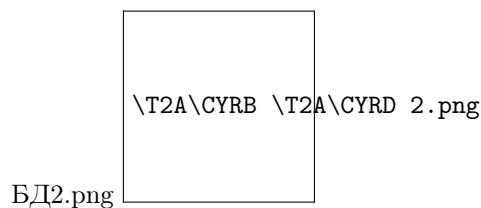
2. Заполнить параметры действия. \* Тип СУБД (Обязательное поле).

Доступно четыре типа СУБД: Oracle MySQL PostgreSQL MsSqlServer

- Логин (Обязательное поле);
- Пароль;
- Хост сервера;
- Порт сервера;
- Имя БД;
- Тайм-аут;
- Уровень изоляции;

- Параметры.

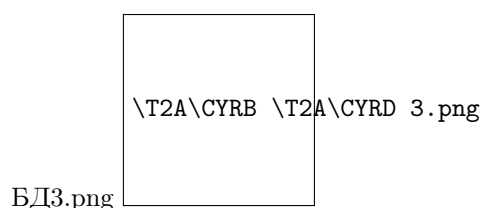
Пример заполнения параметров:



3. Перейти к созданию сценария робота. При запуске робот подключится к выбранной СУБД.

#### Настройка пользовательского подключения:

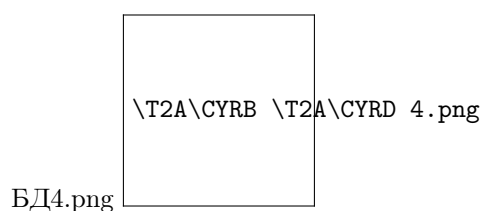
1. Выбрать действие «Пользовательское подключение».



2. Заполнить параметры действия: \* Выбрать уровень изоляции из выпадающего списка. Доступно пять уровней изоляции: Без транзакций; Чтение незаконченных транзакций разрешено; Повторное чтение данных вернет те же значения, что и в начале транзакции; Сериализуемость.

- Логин;
- Пароль;
- URL сервера, где размещена БД (обязательное поле);
- Класс драйвера (обязательное поле);
- Путь к классу драйвера (обязательное поле);
- Тайм-аут.

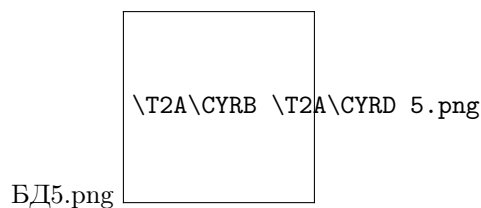
Пример заполнения параметров:



3. Перейти к созданию сценария робота. При запуске робот подключится к выбранной СУБД. По завершению работы с базой данных роботу необходимо закрыть подключение к ней.

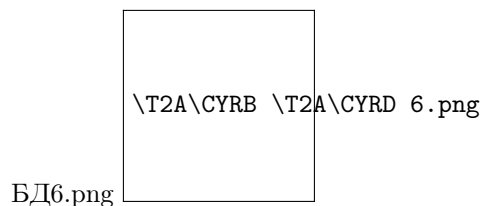
#### Заккрытие подключения к выбранной базе данных

1. Выбрать действие «Закрыть подключение».



2. Заполнить параметры действия: \* Контекст БД (Контекст подключения к выбранной базе данных).

Пример заполнения:

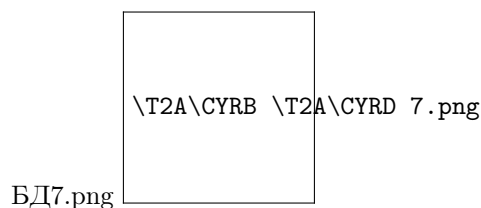


В Robin Studio работу с базой данных можно организовать посредством следующих действий:

### Выполнение запроса

Запрос в базу данных выполняется на языке запросов SQL.

1. Выбрать действие «Выполнить запрос».



2. Заполнить параметры: \* Контекст БД (обязательное поле); \* Шаблон запроса; \* Список параметров.

Пример заполнения запроса без параметров: В поле «Шаблон запроса» вводится запрос на языке SQL.

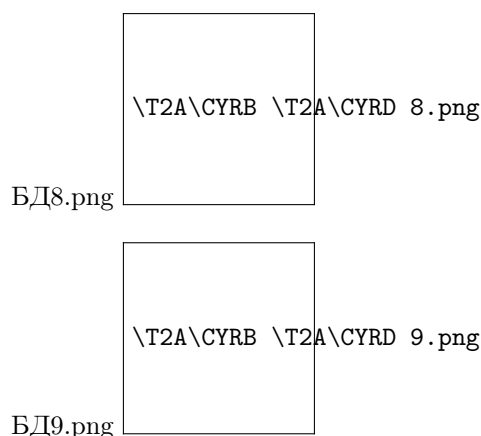
Пример заполнения запроса с параметром. В поле «Шаблон запроса» вводится запрос на языке SQL, где в месте, куда необходимо будет подставить параметр, знак вопроса. Список параметров оформляется в формате словаря в том порядке, в котором они будут заполняться в шаблоне.

Результат действия - таблица.

### Добавление записи

1. Выбрать действие «Добавить запись».





2. Заполнить параметры действия: \* Контекст БД (обязательное поле); \* Имя таблицы (обязательное поле); \* Данные для вставки (обязательное поле);

Поле заполняется в формате словаря. Пример заполнения:

- Список имен полей первичного ключа.

Пример заполнения параметров:

Результат действия - таблица.

### Извлечение записи из базы данных

1. Выбрать действие «Извлечь запись».

2. Заполнить параметры действия: \* Контекст БД (обязательное поле); \* Имя таблицы (обязательное поле); \* Поля таблицы. Заполняются в формате коллекции; \* Значения первичного ключа. Заполняется в формате словаря (обязательное поле).

Пример заполнения полей:

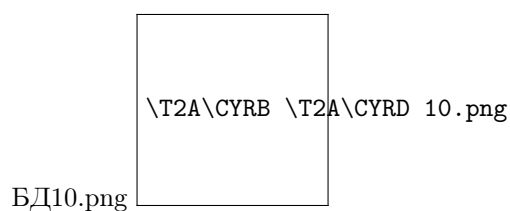
Результат действия: словарь, где ключи – указанные в запросе имена столбцов.

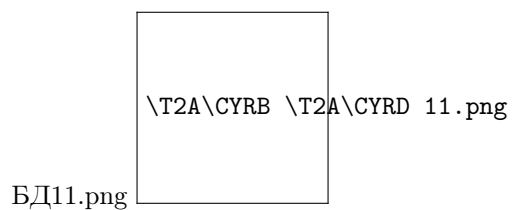
### Обновление записи в базе данных

1. Выбрать действие «Обновить запись».

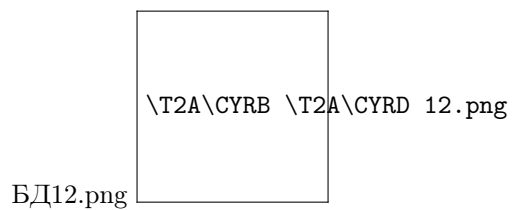
2. Заполнить параметры действия. Все поля обязательны для заполнения: \* Контекст БД; \* Имя таблицы; \* Данные для обновления записей; Поле заполняется в формате словаря.

Пример заполнения:

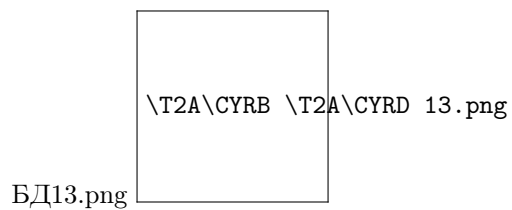




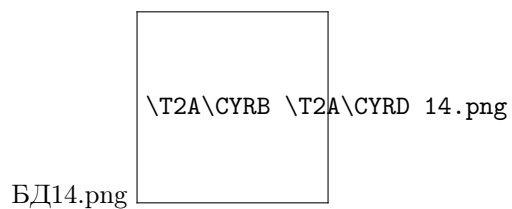
БД11.png



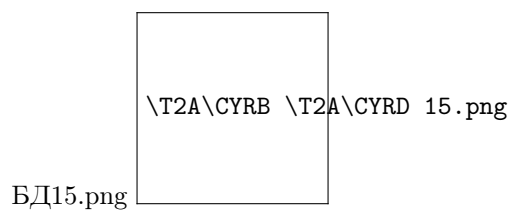
БД12.png



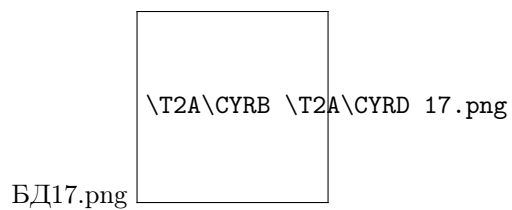
БД13.png



БД14.png



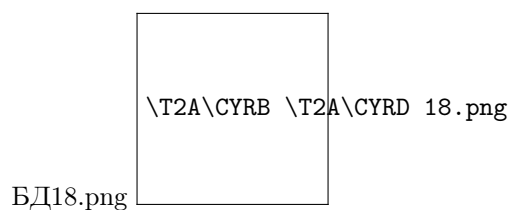
БД15.png



БД17.png

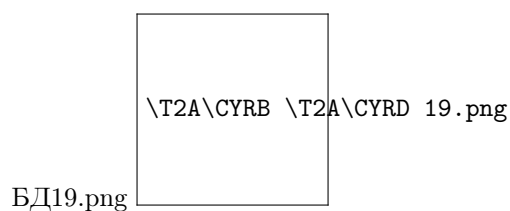
- Первичный ключ.

Пример заполнения параметров:



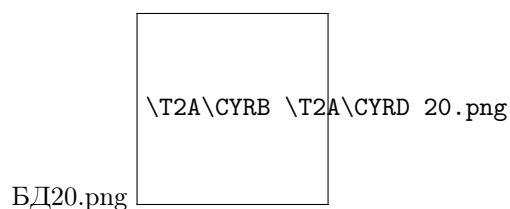
### Удаление записи из базы данных

1. Выбрать действие «Удалить запись».



2. Заполнить параметры действия. Все поля обязательны для заполнения: \* Контекст БД; \* Имя таблицы; \* Первичный ключ; Поле заполняется в формате словаря.

Пример заполнения параметров:



### Заккрытие транзакции

1. Выбрать действие «Заккрыть транзакцию».
2. Заполнить параметры:
  - Контекст БД

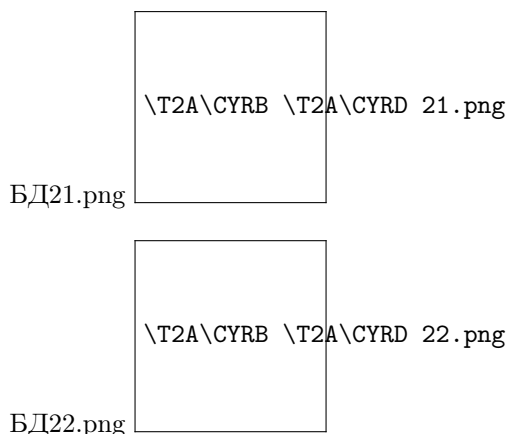
Пример заполнения параметров:

### Откат транзакции

1. Выбрать действие «Откатить транзакцию».

Заполнить параметры: \* Контекст БД

Пример заполнения параметров:



## 2.15 Дополнительные материалы

### 2.15.1 Решение проблем запуска на windows 7 и server 2008 r2 без обновления ESU

Проблема запуска на windows 7 и server 2008 r2 без обновления ESU заключается в минимальных требованиях .NET Desktop Runtime 5.0.x:

<https://docs.microsoft.com/en-us/dotnet/core/install/windows?tabs=net50>

При этом на машине все равно может быть установлен .NET Desktop Runtime 5.0.x.

Проблема проявляется при завершении установки платформы, при попытке запуска ROBIN Agent:

Ошибку можно обнаружить до установки ROBIN. Для проверки работоспособности .NET необходимо ввести выполнить команду: `dotnet --list-runtimes`

Результат выполнения должен выглядеть вот так:

```
Microsoft.NETCore.App 5.0.10 [C:/Program Files/dotnet/shared/Microsoft.NETCore.App]
Microsoft.WindowsDesktop.App 5.0.10 [C:/Program Files/dotnet/shared/Microsoft.WindowsDesktop.App]
```

#### Потенциальное решение:

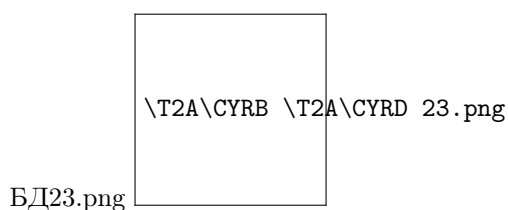
Поставить обновления ESU:

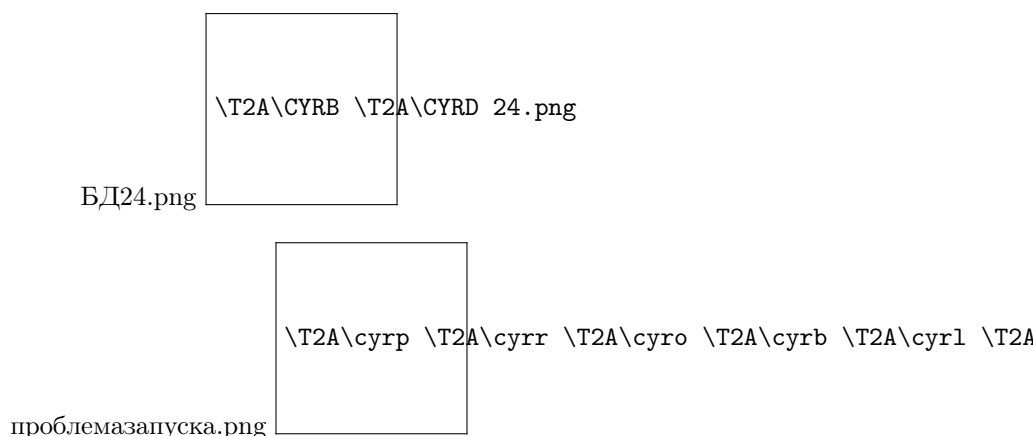
<https://www.catalog.update.microsoft.com/Search.aspx?q=KB4538483>

Обновления установятся только на windows 7 Professional и Enterprise (SP1) и Server 2008 r2 (SP1).

### 2.15.2 Действия по работе с Java приложениями

Работа с настольными Java приложениями происходит с помощью отдельной группы действий «Java». Действия этой группы используют технологию JavaBridge для получения свойств элементов





окон приложений.

Для корректной работы необходимо проверить, что JavaBridge активирован в панели управления:

1. Выберите « Пуск » > « Панель управления » > « Центр специальных возможностей » . Либо нажмите клавишу с логотипом Windows + u, чтобы получить доступ к Центру специальных возможностей.
2. Выберите «Использовать компьютер без дисплея» .
3. В разделе «Другие установленные программы» установите флажок «Включить Java Access Bridge» (возможно, вам придется прокрутить вниз).

---

**Важно:** После включения Java Access Bridge необходимо перезапустить Java приложения.

---

### 2.15.3 Инструкция по добавлению драйвера к браузеру

Платформа работает с веб-страницами через так называемые драйвера, которые выпускаются под каждый браузер. В составе дистрибутива ROBIN уже идут последние версии необходимых драйверов. Однако при очередном обновлении браузера могут потребоваться более новые версии драйверов, тогда при открытии браузера робот будет выдавать предупреждение: «<название действия> не удалось запустить. Требуется драйвер для «Chrome»/«версия хрома»».

При возникновении такой ошибки, может потребоваться указать более новый драйвер. Драйвера для соответствующего браузера доступны по ссылкам:

- Chrome: <https://chromedriver.chromium.org/downloads>
- Firefox: <https://github.com/mozilla/geckodriver/releases/>
- IE: <https://www.selenium.dev/downloads/>
- Yandex: <https://github.com/yandex/YandexDriver/releases>

Для использования скаченного драйвера, необходимо выполнить следующие шаги:

1. Скачать версию драйвера для соответствующей версии браузера.
2. Разместить скаченный драйвер в файловой системе. Например, в корень диска C.
3. Создать папку с именем «.robin» в профиле текущего пользователя. В Windows это обычно: «C:/Users/{имя текущего пользователя}».

При создании папки через интерфейс в Windows может возникнуть ошибка:

При возникновении ошибки, следует вводить имя не «.robin», а «.robin.» (с точкой в конце).

Папка создастся с именем «.robin».

ошибка.png

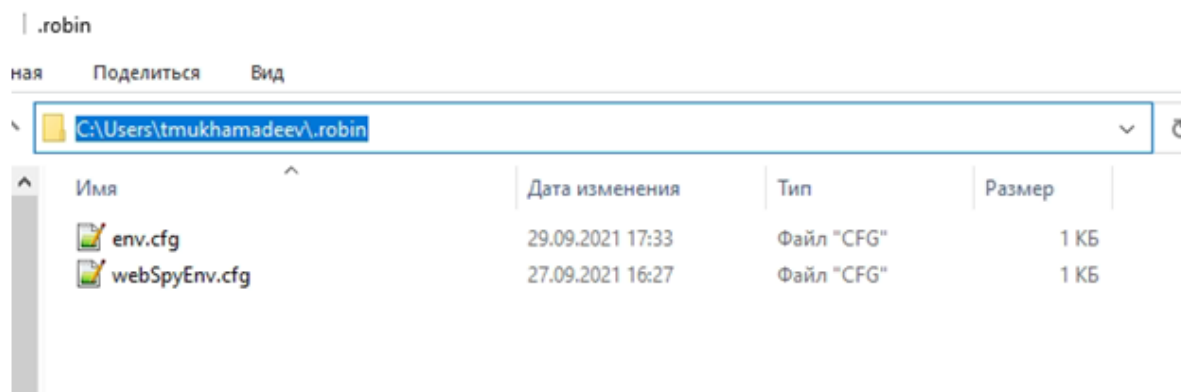
\T2A\cyro \T2A\cyrsh \T2A\cyri \T2A\cyrb \T2A\cyrk \T2A\cy

4. В папке необходимо создать 2 файла:

env.cfg - конфигурационный файл для робота.

webSpyEnv.cfg - конфигурационный файл для работы web-recorder.

Файлы можно создать с помощью Блокнот. При сохранении файла следует указать расширение .cfg, выбрав в окне сохранения «тип файла» значение «All types (.\*)».



5. Заполняются файлы в следующем формате:

[webdrivers]

webdriver.[название браузера].[версия браузера].path=[абсолютный путь к драйверу]

, где [название браузера] -

- chrome
- firefox
- ie
- yandex

[версия браузера] - указывается версия браузера в формате пишется в формате v[число].

**Внимание! Оба конфига работают с заголовком [webdrivers]**

Примеры заполнения:

файла.png

2\T2A\cyrf \T2A\cyra \T2A\cyrishrt \T2A\cyrl \T2A\cyra .png

6. Сохранить config файлы в папку «.robin». Теперь роботы и студия в случае отсутствия нужного драйвера в дистрибутиве будут использовать драйвера, указанные в конфигурационных файлах.

## 2.15.4 Работа с JSON

Группа действий предназначена для обработки данных в форме json.

Состав группы действий «JSON»:

Действия с JSON			
Название действия	Тип входные параметры	Обязательность параметра	Тип результат выполнения действия
Создать JSON	-	-	JSONObject
Установить значение	JSONObject	да	
	ключ (строка)	да	
	значение (объект)	нет	
Удалить элемент	JSONObject	да	объект (значение, связанное с ключом, или null)
	ключ (строка)	да	
Получить количество элементов	JSONObject	да	количество ключей
Получить коллекцию значений	JSONObject	да	коллекция значений
Получить коллекцию ключей	JSONObject	да	коллекция ключей
Содержит ключ	JSONObject	да	boolean
	ключ (строка)	да	
Очистить JSON	JSONObject	да	-
Преобразовать JSON в строку	JSONObject	да	строка
Преобразовать строку в JSON	строка	да	JSONObject
Преобразовать JSON в словарь	JSONObject	да	словарь
Преобразовать словарь в JSON	словарь	да	JSONObject
Сравнить JSON	первый JSONObject	да	boolean: true, если файлы одинаковые
	второй JSONObject	да	
Сохранить в файл	JSONObject	да	путь к файлу (path)
	имя файла (строка)	да	
Открыть файл	папка (path)	да	
	путь к файлу (path)	да	JSONObject

## 2.15.5 Режим отображения сценария работы

На верхней панели управления в ROBIN Studio с помощью кнопки «Переключить режим отображения» можно переключить режим отображения сценария. Доступны два вида отображения алгоритма работы:

- Построчный, где каждая строка - это действие робота с вызываемыми параметрами и его результатами. Такой режим удобен для навигации по сценарию.
- Диаграмма (блок-схема), где каждый блок - действие. Такой режим представления является режимом по умолчанию, в нем удобно просматривать процесс в нотации ROBIN, а также его редактировать.

### 2.15.6 Подключение инструментов OCR

Платформа ROBIN поставляется вместе со встроенным OCR-модулем Google Tesseract 5. Он поставляется бесплатно и как есть, может работать on-premise в контуре заказчика без необходимости подключения к сети Интернет.

Также с платформой поставляются действия по работе со следующими инструментами (движками):

- ABBYY FlexiCapture Distributed (<https://www.abbyy.com/ru/flexicapture/>)
- ABBYY Cloud (<https://www.abbyy.com/cloud-ocr-sdk/>)
- Yandex Vision OCR (<https://cloud.yandex.ru/docs/vision/concepts/ocr/>)
- Microsoft Azure OCR (<https://docs.microsoft.com/ru-ru/azure/cognitive-services/computer-vision/overview-ocr>)
- DBrain (<https://dbrain.io/>)

Также в рамках проекта готовы рассмотреть интеграцию с решением Заказчика.

### 2.15.7 Формирование и хранение логов

На машине, где разрабатывают и запускают готовых роботов, формируются логи. Лог файлы помогают контролировать работу компонентов и в случае возникновения ошибок быстро найти их и устранить. В таблице ниже представлено описание лог-файлов ROBIN Studio, ROBIN Player, ROBIN Agent, ROBIN Executor, исходников (схем) роботов, исполнения готовых программных роботов.



№	Компонент	Пример лога	Формат записи	Место формирования и хранения логов (куда передаются)
---	-----------	-------------	---------------	---

**ROBIN Studio** Логи работы самого приложения. Формируются и хранятся локально там же, где установлено приложение. Удаляются файлы логов автоматически, по достижению лимита количества этих файлов (старые файлы заменяются новыми). Через конфигурационный файл приложения Studio.exe.config можно настроить путь сохранения и название файла лога, кодировку, максимальный размер файла лога, максимальное количество хранимых файлов лога, формат записи (паттерн) строки лога. Логи позволяют контролировать процессы самого приложения и выявить ошибки, возникшие во ходе работы ROBIN Studio.

1	ROBIN Studio	Формат названия: log[n].log, где n – это порядковый номер файла лога	<ul style="list-style-type: none"> <li>Дата и время записи;</li> <li>Пользователь под которым запущен компонент;</li> <li>Тип сообщения лога;</li> <li>Информационное сообщение (datetime – дата и время выполнения процесса; pid- process id – идентификатор процесса; thread – количество потоков; message – название совершенного процесса; source – источник процесса; category – категория процесса; identity – пользователь под которым выполнен процесс)</li> </ul>	%USERPROFILE%\AppData\Local\Platform2.0.0StudioLogs
---	--------------	--	--	---

**Типы сообщения логов** \* <INFO> содержит информацию о работе самой студии (загрузка студии; подключение экзекьютора; завершение работы студии; подключение визуальных модулей и т.д.) \* <DEBUG> Level – debug, содержит информацию о процессах, происходящих внутри студии (открытие/закрытие проекта; действия, совершаемые роботом и т.д.). \* <ERROR> содержит информацию об ошибке. Level – error. После тех же блоков, что и в <INFO> и <DEBUG> следует блок exception, содержащий в себе блоки message – текст ошибки. stackTrace - Трассировка стека, позволяющая отслеживать последовательность вызванных функций - до точки, в которой трассировка стека была создана.

**ROBIN Player** Логи работы самого приложения. Формируются и хранятся локально там же, где установлено приложение. Удаляются файлы логов автоматически, по достижению лимита количества этих файлов (старые файлы заменяются новыми). Настройка производится в конфигурационном файле RobotPlayer.exe.config. Логи позволяют контролировать процессы самого приложения и выявить ошибки, возникшие во ходе работы ROBIN Player.

2	ROBIN Player	Формат названия: log[n].log, где n – это порядковый номер	<ul style="list-style-type: none"> <li>Информационное сообщение (datetime</li> </ul>	%USERPROFILE%\AppData\Local\Platform2.0.0StudioLogs
---	--------------	---	--	---

В таблице ниже представлены примеры логов, сформированных в результате работы каждого компонента.

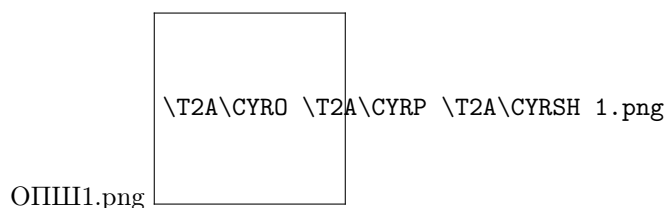


## 2.15.8 Варианты оповещения пользователя при ошибке робота

При работе робота могут возникнуть какие-либо ошибки, в результате которых выполнение сценария завершиться. Существует несколько способов оповещения пользователя о возникновении ошибки, рассмотрим их подробнее.

### Обработка ошибок внутри сценария

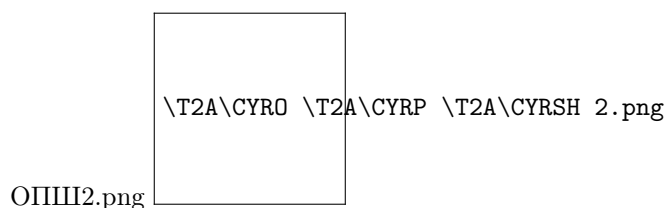
#### 1. Try-Catch



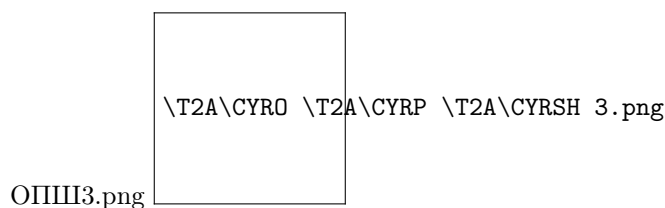
При возникновении ошибок среди действий, помещенных в блок «Try» в лог будет выведено сообщение. Работа робота не прекратиться, а будут выполнены либо действия из блоков «Catch» или «Finally», либо робот продолжит выполнение сценария вне действия «Try-Catch».

#### 2. Действие «Вывести ошибку»

1. Из группы действий «Базовые действия» выбрать действие «Вывести ошибку» и перенести его в схему робота;

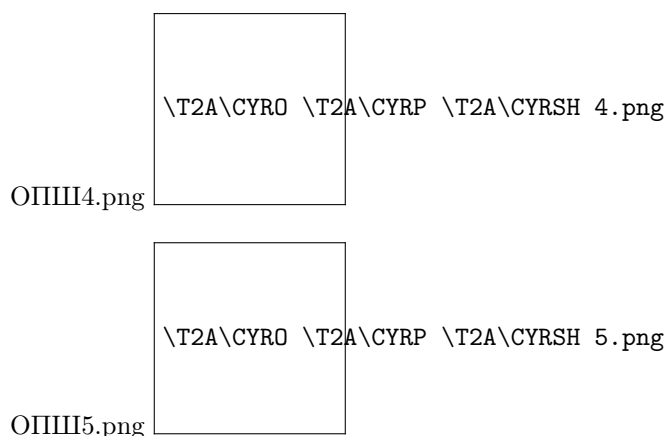


2. В параметрах действия в строке «Сообщения» указать текст ошибки, который будет выведен в лог. Если оставить это поле пустым, то в логе будет выведена «Неизвестная ошибка»;



3. После настройки действия запустить робота нажатием кнопки запуска;
4. Когда робот дойдет до действия «Вывести ошибку», в лог в студии появится запись об ошибке и робот завершит свою работу;

#### 3. Информационное сообщение



Сценарий робота может быть настроен таким образом, что в случае возникновения ошибки можно вывести информационно сообщение посредством действия «Сообщения» из группы действий «Диалоговые окна». Пример: Установить в «Catch» данное действие, тогда перед тем, как робот перейдет к обработке ошибки, на экране появится информационное сообщение.

#### 4. Отправка письма на почту пользователя в случае возникновения ошибки.

Сценарий робота можно организовать таким образом, что в случае возникновения ошибки пользователю будет направлено письмо на электронный почтовый сервис. Отправка письма осуществляется посредством действий из группы действий «Почта».

- Выбрать действие «Настройка подключения»;
- Заполнить параметры действия;

Пример:

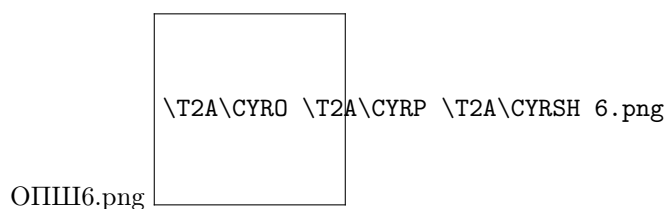
- Выбрать действие «Отправить сообщение»;
- Заполнить параметры действия, в поле «Тело письма» указать необходимую информацию;

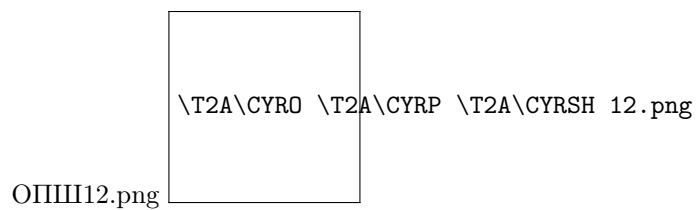
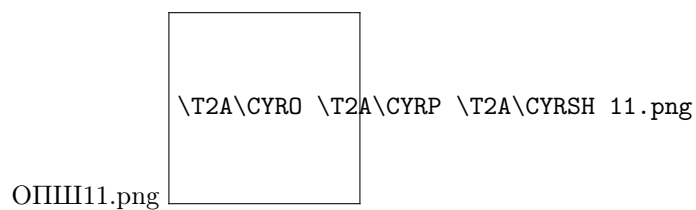
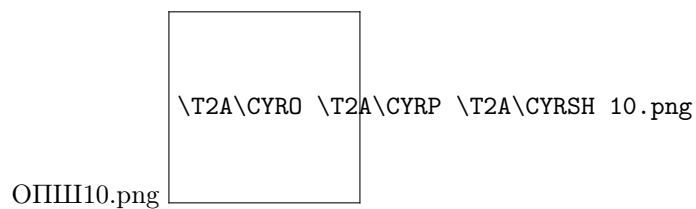
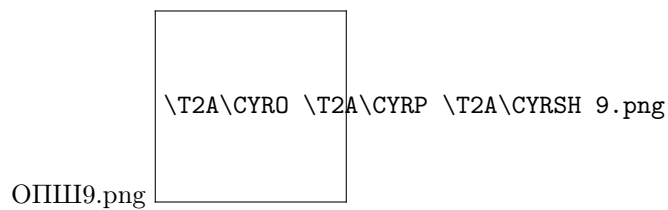
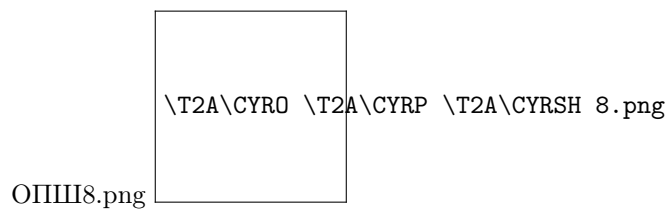
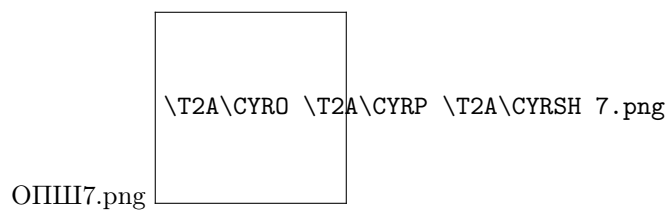
Пример заполнения:

#### Настройка оповещения пользователя о возникновении ошибки в Robin Player.

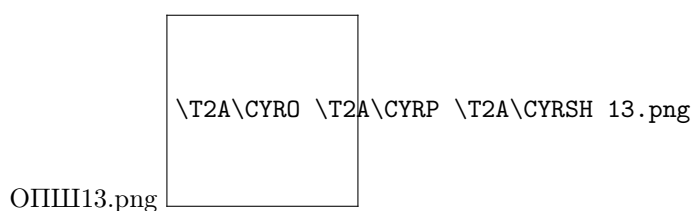
Компонент Robin Player позволяет настраивать отправку сообщений на почту в случае корректного или некорректного завершения работы робота. Для того, чтобы настроить информирование пользователя в случае, если робот завершил свою работу с ошибкой, необходимо выполнить следующие действия:

1. Перейти на вкладку «Настройки» в Robin Player;
2. Перейти на вкладку «Отправка писем при завершении выполнения»;

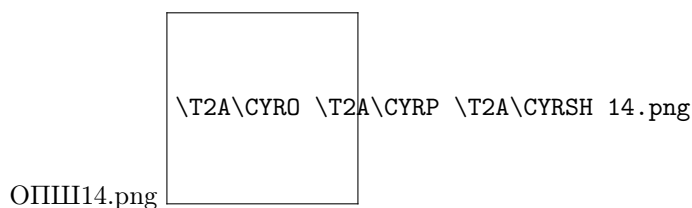




3. Заполнить требуемые параметры:



4. Заполнить «Список получателей уведомлений о выполнении роботов», в поле «Неуспешные выполнения» указать почтовый адрес пользователя, которому будет направлено уведомление. При выборе опции «Прикрепить лог», пользователь получит лог работы робота.



Указав почтовый адрес в графе «Успешные выполнения», пользователь получит информацию о том, что робот завершил свою работу без ошибок.

## 2.15.9 Действия по работе с SAP GUI

### Общая информация

Группа действий по работе с SAP GUI позволяет выполнять роботу автоматизированные действия с интерфейсом приложения. Для обеспечения работы группы действий используется SAP GUI Scripting API, при этом сервер SAP не видит разницы между взаимодействием с помощью SAP GUI Scripting и действиями выполняемыми пользователем. Сценарий робота имеет те же права на выполнение операций SAP и ввод данных, что и пользователь. При работе с помощью SAP GUI Scripting работают такие же правила проверки данных в полях, как и при работе пользователя. Для обеспечения работы SAP GUI Scripting API возможность должна быть включена как на сервере, так и на клиенте.

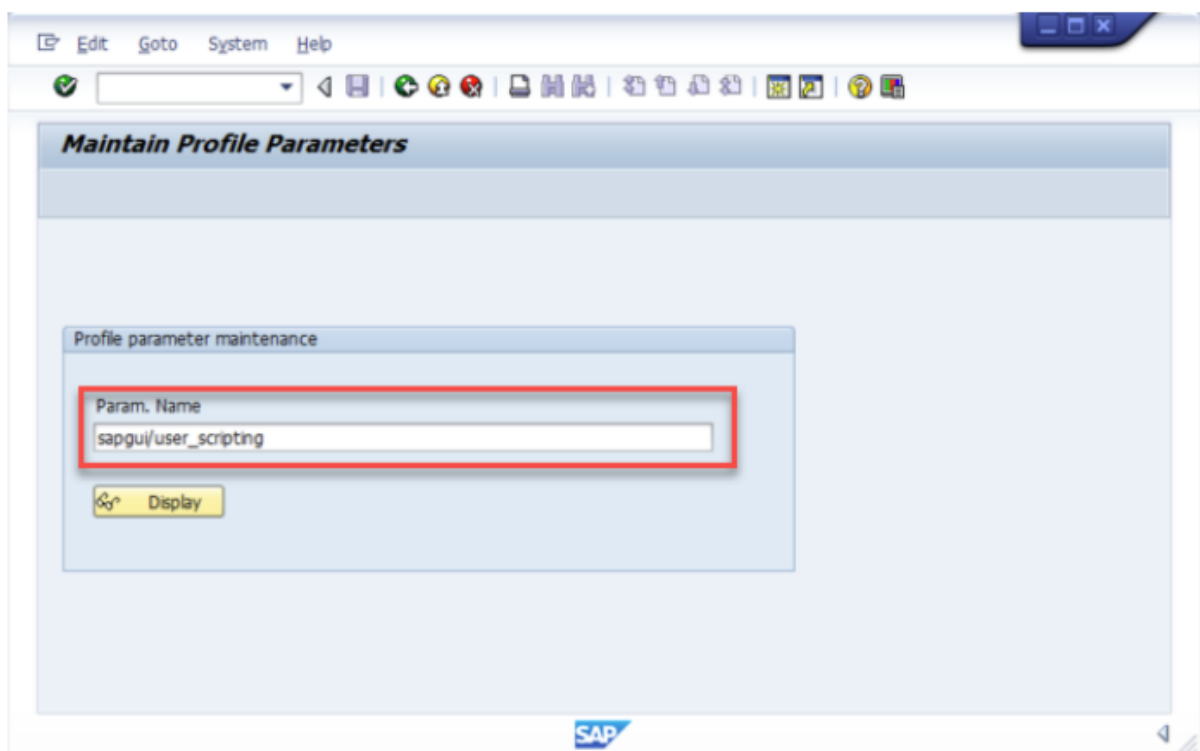
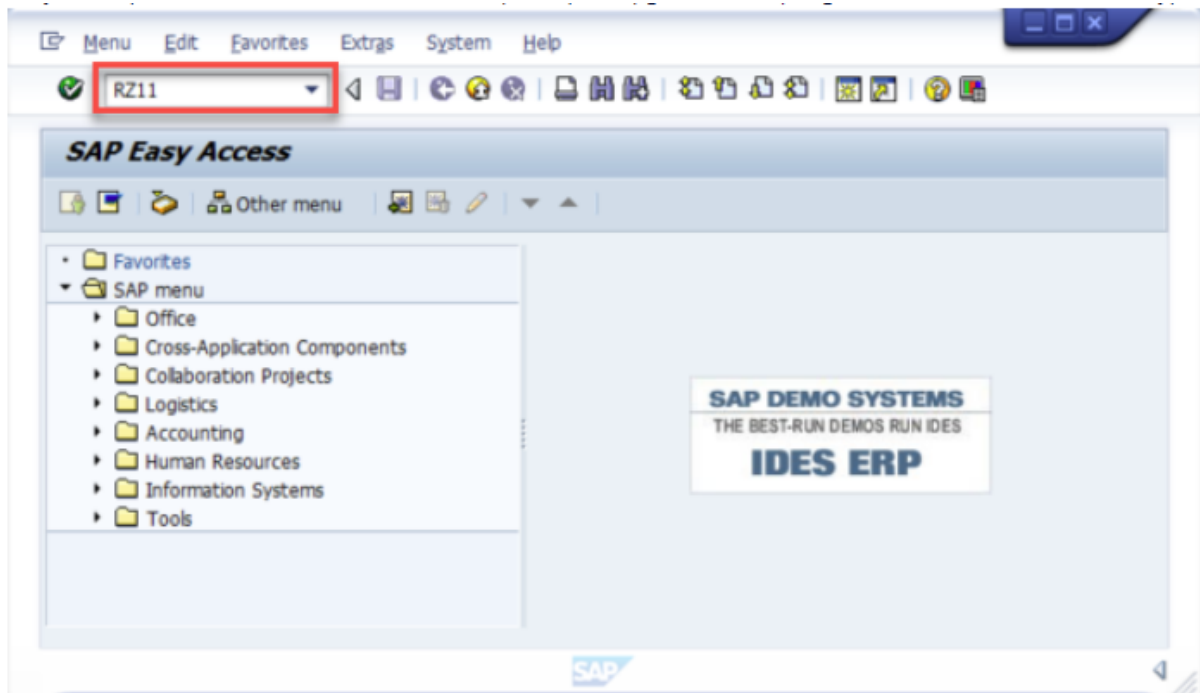
### Активация сценариев SAP GUI на стороне сервера

1. Запустите «saplogon.exe», войдите на сервер SAP, используя свои учетные данные. Откроется окно SAP Easy Access. Запустите транзакцию RZ11. Укажите имя параметра «sapgui/user\_scripting» и нажмите Enter на клавиатуре или «Отобразить» («Display») в интерфейсе SAP.

2. В окне «Отображение атрибутов параметра» профиля («Display Profile Parameter Attributes») нажмите кнопку «Изменить значение» («Change Value») на панели инструментов и задайте для параметра «sapgui/user\_scripting» новое значение (New value) = TRUE. Сохраните изменения.

3. Выйдите из системы и снова войдите в нее, чтобы проверить вступление изменений в силу.

Сообщение в строке состояния «Имя параметра неизвестно» («Parameter name is unknown») указывает на то, что параметр sapgui/user\_scripting не может быть изменен





### Display Profile Parameter Attributes

Param. Name  
sapgui/user\_scripting

Short description(Engl)  
Enable or disable user scripting on the frontend.

Appl. area  
GUI Frontend ☐

ParameterTyp  
Logical value ☐

Changes allowed  
Change permitted ☐

Valid for oper. system  
All operating systems ☐

DynamicallySwitchable  
☒

Same on all servers  
☐

Dflt value  
FALSE

ProfileVal  
FALSE

Current value  
TRUE

Switched from  
SAPUSER 12.06.2018 16:53:35

Change Parameter Value

Parameter values

Param. Name  
sapgui/user\_scripting



Dflt value  
FALSE

ProfileVal  
FALSE

Current value  
TRUE

New value  
TRUE

☐ Switch on all servers

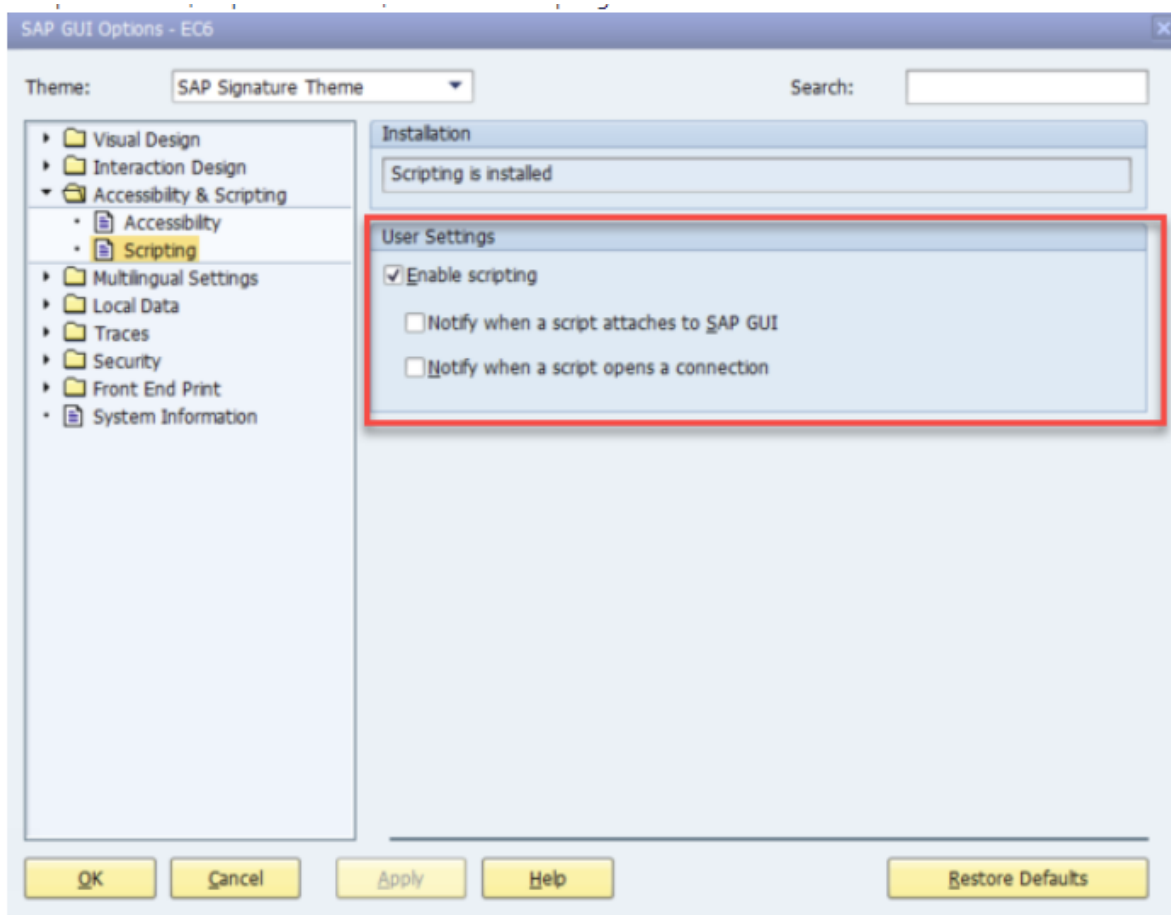
 

с помощью этой транзакции, поскольку отсутствует пакет сервера SAP. Необходима предварительная установка отсутствующего пакета.

### Активация сценариев SAP GUI на стороне клиента

1. На панели инструментов нажмите кнопку «Изменить локальный макет» («Customise Local Layout») и выберите «Параметры» («Options») из раскрывающегося меню. Откроется окно «Параметры SAP GUI» («SAP GUI Options»).
2. Перейдите в раздел «Доступность и сценарии» («Accessibility & Scripting») и нажмите «Сценарии» («Scripting»). В разделе «Настройки пользователя» («User Settings») установите флажок «Включить сценарии» («Enable scripting»). Два других параметра — «Уведомить, когда сценарий подключается к SAP GUI» («Notify when a script attaches to SAP GUI») и «Уведомить, когда сценарий открывает соединение» («Notify when a script opens a connection») — должны быть выключены. Сохраните изменения, нажав кнопку «OK».

Теперь запись сценариев с помощью SAP GUI Scripting активна.



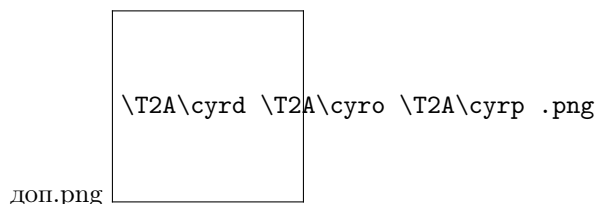
### Дополнительная информация

С помощью использования SAP GUI Scripting ROBIN позволяет автоматизировать действия пользователя. Одновременная автоматизация SAP GUI для Windows и SAP Business Client невозможна. Сценарии робота, работающие с этими приложениями должны выполняться последовательно.

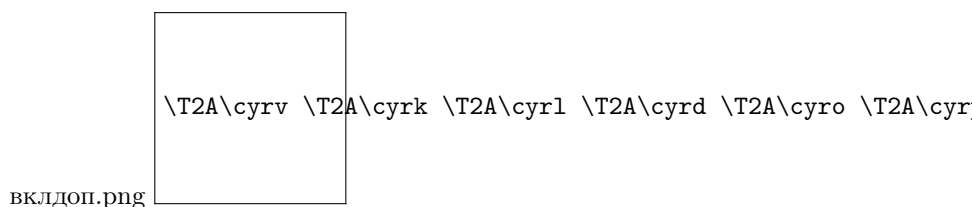
### 2.15.10 Указание дефолтной папки для скачивания из браузера

Для работы некоторых роботов необходимо скачивать файлы из браузера в конкретную папку, а не в папку «Загрузки». Для этого возможно изменить дефолтную папку для скачивания из браузера. Шаги для решения задачи описаны ниже:

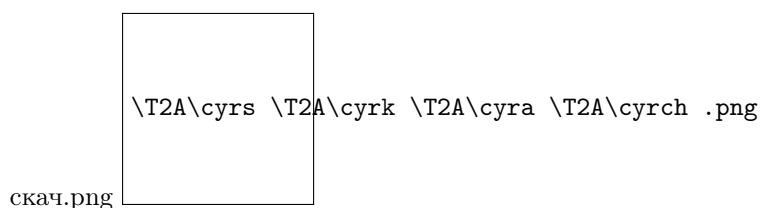
1. Открыть настройки в браузере «Chrome».



2. На открывшейся вкладке выбрать опцию «Дополнительные»:



3. В раскрывшемся подменю выбрать пункт «Скачанные файлы»:



4. Затем нажать кнопку «Изменить»:

5. Выбрать папку для скачивания файлов.

6. Теперь скаченные роботом файлы из браузера сохранятся в выбранную папку.

### 2.15.11 Динамическая замена Xpath

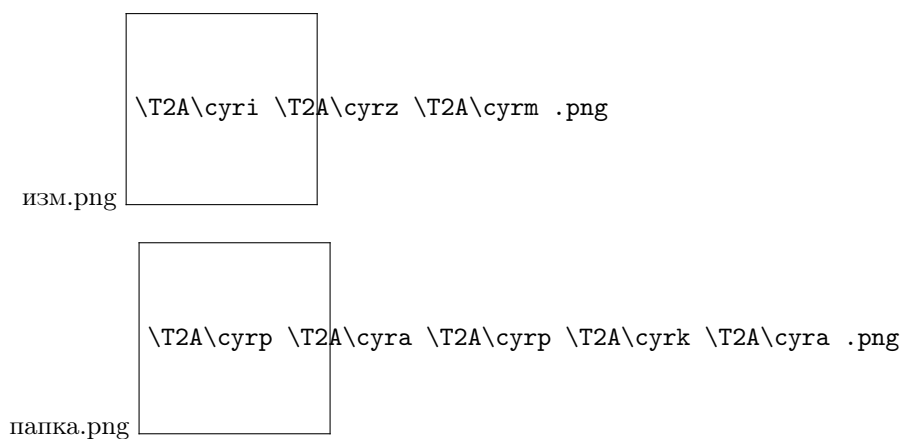
**Xpath** - язык запросов к элементам xml или html документа.

Действия студии группы «Браузеры» обращаются к элементам страницы по их пути в html разметке. Путь до элемента задается в параметре действий «XPath» с помощью языка XPath.

Для получения нескольких веб-элементов, имеющих практически одинаковые значения Xpath, необходимо организовать динамическую замену Xpath посредством функционала ROBIN Studio.

Используя действия Robin Studio в значении XPath, сохраненном в текстовую переменную, происходит замена части, которая различна в XPath каждого веб-элемента.

Рассмотрим порядок работы XPath более подробно.



1. Создать переменную с типом данных «Строка», в качестве значения переменной указать полный XPath элемента, где различающаяся часть будет заменена каким-либо словом. Пример заполнения:

**НОВАЯ ПЕРЕМЕННАЯ**

**Имя переменной**

XPATH

**Тип переменной**

{ABC} Строка

**Значение**

/html/body/form/section[2]/div/div/div[1]/div[3]/div[ЗАМЕНА]/div

Значение Xpath:

2. Выбрать действие «Заменить по шаблону» из группы «Текст».
3. Заполнить параметры действия:
  - Параметр «Текст» - указать созданную ранее переменную, в которой храниться XPath.
  - Параметр «Шаблон» - указать регулярное выражение для сопоставления, т. е. то слово, которое необходимо заменить.
  - Параметр «Подстановка» - указать новое значение для текста, который необходимо заменить.

Пример заполнения параметров:



```
/html/body/form/section[2]/div/div/div[1]/div[3]/div[ЗАМЕНА]/div[1]/div[1]/div[1]/div/div[1]/a
```

**Сохранить**

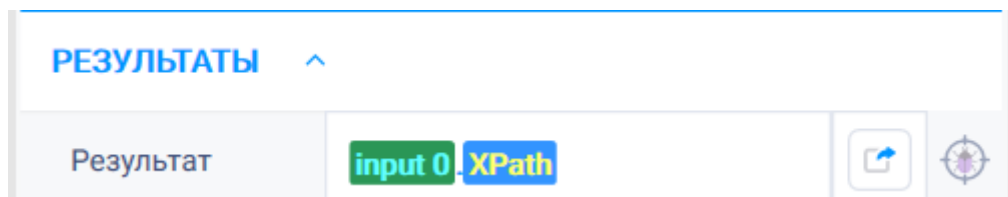
Отмена

Заменить по  
шаблону**ПАРАМЕТРЫ** ^

* Текст	<b>ХРАТН</b>	...
* Шаблон	ЗАМЕНА	...
* Подстановка	<b>счетчик</b>	...

- В параметре «Результат» указать XPath веб-элемента, XPath которого нужно изменить.

Пример:



- Затем, после заполнения параметра «Элемент» какого-либо действия группы «Браузеры» переменной, чей XPath указан для сохранения результата в действии «Заменить по шаблону», действие будет выполнено с измененным XPath.

Для лучшего понимания рассмотрим кейс.

### Кейс «Планы закупок»

**Задача:** С сайта «Портал поставщиков» получить наименование заказчика первых 5 планов закупки, сохранив это в excel-файл.

**Решение:**

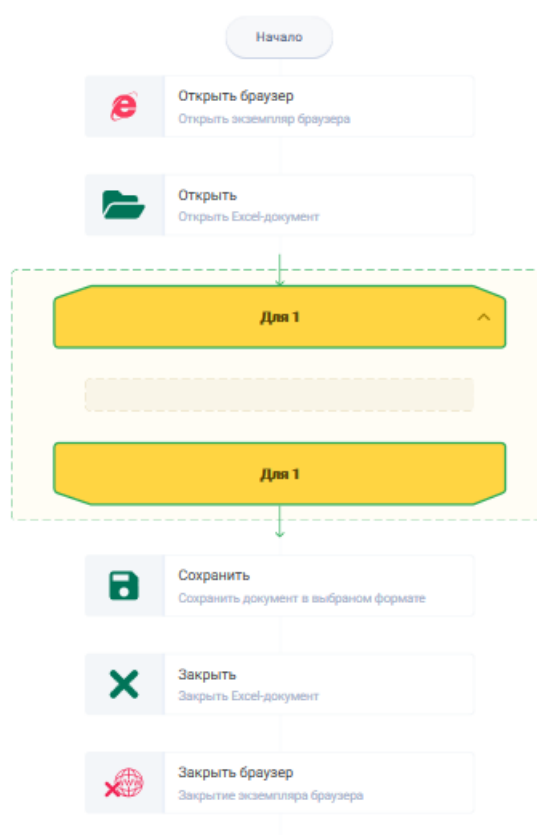
- Перенесем на рабочую область действия:
  - «Открыть браузер», «Закрыть браузер» из группы действий «Браузеры»;
  - «Открыть», «Сохранить», «Закрыть» из группы действий «Excel»;
  - «Для» группы «Базовые действия»;

Пример:

- Заполнить параметры действий:

- Действие «Открыть браузер» - выбрать из выпадающего списка браузер «Chrome», в строке «Адрес» указать ссылку на сайт «Портал поставщиков», вкладку с планами закупок. Пример:
- Действие «Закрыть браузер» - контекст браузера.
- Действие «Открыть» - в параметре «Путь к файлу» указать путь до excel-файла.
- Действия «Сохранить» и «Закрыть» - указать для параметра «Контекст» указать открытый ранее excel-файл.
- Действие «Для»:
  - «Начальное значение» -1;
  - «Конечное значение» - 5;
  - «Итератор» - 1;
- Проанализировать XPath веб-элемента:

Сравнить XPath двух подряд идущих планов закупок для информации о заказчике, выписав их в текстовый файл. Видно, что отличия между ними лишь в одной цифре.



ПАРАМЕТРЫ ^	
Браузер	Chrome ^
Свойства	...
Время явного о...	...
Фоновый режим	<input type="checkbox"/> ...
Адрес	https://old.zakupki.mos.ru/#/plans? ...
Сохранить в	...

ПАРАМЕТРЫ ^	
* Контекст	Открыть браузер Контекст

ПАРАМЕТРЫ ^	
* Путь к файлу	C:\Users\persi\OneDrive\Рабочий стол\планы.xlsx
Видимый	<input type="checkbox"/>

ПАРАМЕТРЫ ^	
* Контекст	<div>Открыть</div> <div>Контекст</div>

4. Создать переменную с типом данных «Строка», значение которой будет XPath заказчика, где изменяющаяся цифра будет заменена словом «Заказчик».

Пример:

5. Заполнить тело цикла действиями «Конвертировать данные» из группы действий «Базовые действия», «Заменить по шаблону» из группы действий «Текст», «Получить текст» из группы действий «Браузер» и «Установить значение ячейки» из группы действий «Excel».

Пример:

6. Заполнить параметры действий:

а. Действие «Конвертировать данные»: Параметр «Источник» - состояние итератора цикла «Для»; Параметр «Тип» - «Строка». Пример:

б. Действие «Получить текст»: Параметр «Контекст» - экземпляр открытого ранее браузера Параметр «Элемент» - веб элемент, который необходимо создать. Пример:

**Для создания веб-элемента необходимо выполнить следующие шаги:**

1. Из доступных способов заполнения параметра выбрать «Добавление нового ресурса».
2. Заполнить параметры «DisplayName», например, «Заказчик» и «Default Attribute», выбрав из выпадающего списка значение «XPath».

Файл Правка Формат Вид Справка

XPath реестровый номер

/html/body/div[5]/div/div/div/div[1]/div/div/div[1]/div[2]/div[1] div/div/div[1]/div[1]/h2/a  
/html/body/div[5]/div/div/div/div[1]/div/div/div[1]/div[2]/div[2] div/div/div[1]/div[1]/h2/a

XPath заказчик

/html/body/div[5]/div/div/div/div[1]/div/div/div[1]/div[2]/div[1] div/div/div[1]/div[2]  
/html/body/div[5]/div/div/div/div[1]/div/div/div[1]/div[2]/div[2] div/div/div[1]/div[2]



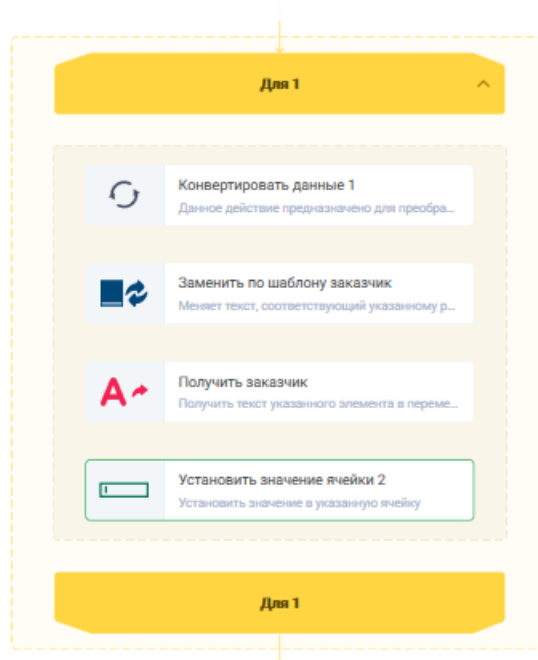

### НОВАЯ ПЕРЕМЕННАЯ

**Имя переменной**

**Тип переменной**

{ABC} Строка

**Значение**



ПАРАМЕТРЫ	
* Источник	Для 1 Состояние итератора
* Тип	Строка

3. Нажать кнопку «Сохранить».

4. В действии «Заменить по шаблону» указать:

- Параметр «Текст» - созданная ранее переменная «XPath заказчик»;
- Параметр «Шаблон» - слово «ЗАКАЗЧИК», которое будет заменено;
- Параметр «Подстановка» - результат действия «Конвертация данных».

Параметр «Результат» - XPath веб-элемента «Заказчик».

5. В действии «Установить значение ячейки» указать:

- Параметр «Контекст» - экземпляр открытого excel-файла;
- Параметр «Ячейка» - с помощью инструмента построения выражений заполнить данный параметр;

Посредством действия «Соединить из группы «Текст» соединить номер столбца «А» и номер строки, равный состоянию итератора цикла. Так как соединяются текстовые значения, указать результат действия «Конвертация данных».

Параметр «Значение» - результат действия «Получить текст».

6. Запустить работа.

ДОБАВЛЕНИЕ НОВОГО РЕСУРСА

заказчик

Page title

Url

ElementId

ElementName

XPath

Selector

Class

TagName

LinkText

JavaScript



Iframe

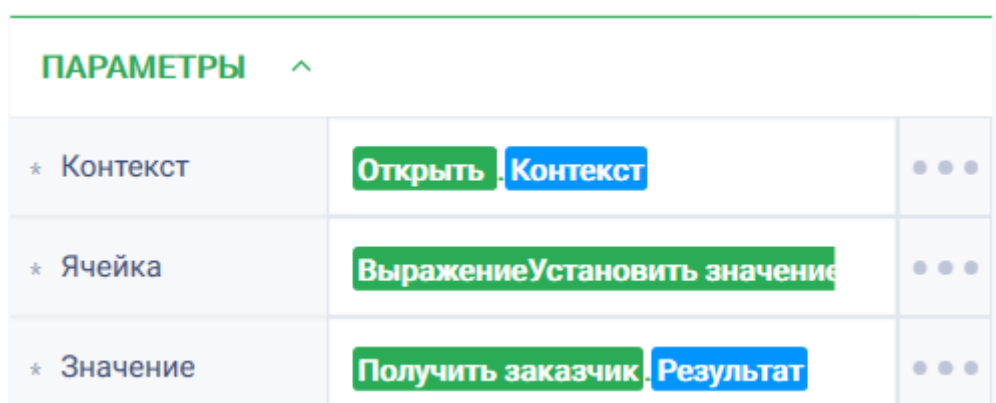
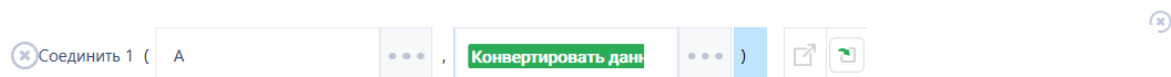
Default Attribute XPath

Spy

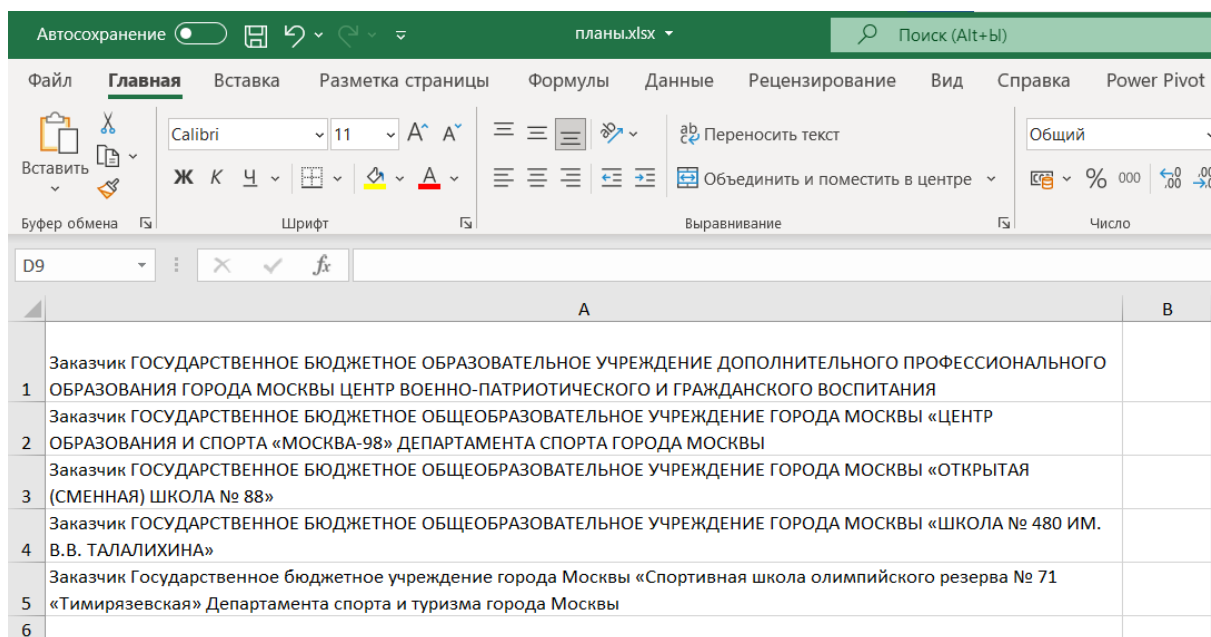
Сохранить Отмена

ПАРАМЕТРЫ ^		
* Текст	XPath заказчик	...
* Шаблон	ЗАКАЗЧИК	...
* Подстановка	Конвертировать данные 1. Резул	...

РЕЗУЛЬТАТЫ ^		
Результат	Заказчик XPath	 



7. Результат работы робота - заполненный excel-файл.



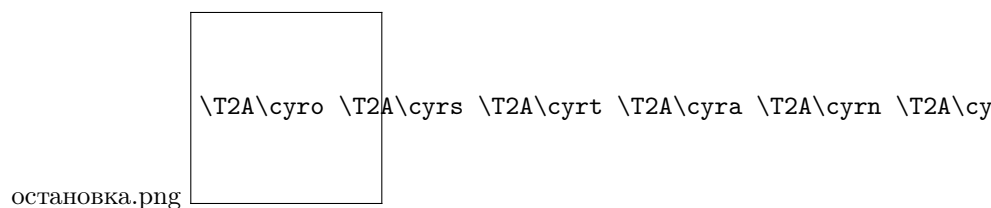
	A	B
1	Заказчик ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ ЦЕНТР ВОЕННО-ПАТРИОТИЧЕСКОГО И ГРАЖДАНСКОГО ВОСПИТАНИЯ	
2	Заказчик ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ «ЦЕНТР ОБРАЗОВАНИЯ И СПОРТА «МОСКВА-98» ДЕПАРТАМЕНТА СПОРТА ГОРОДА МОСКВЫ	
3	Заказчик ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ «ОТКРЫТАЯ (СМЕННАЯ) ШКОЛА № 88»	
4	Заказчик ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ «ШКОЛА № 480 ИМ. В.В. ТАЛАЛИХИНА»	
5	Заказчик Государственное бюджетное учреждение города Москвы «Спортивная школа олимпийского резерва № 71 «Тимирязевская» Департамента спорта и туризма города Москвы	
6		

### 2.15.12 Принудительная остановка робота

Необходимость в принудительном завершении работы робота возникает в связи с тем, что алгоритм на этом шаге продолжаться не должен. Остановить робота можно следующими способами:

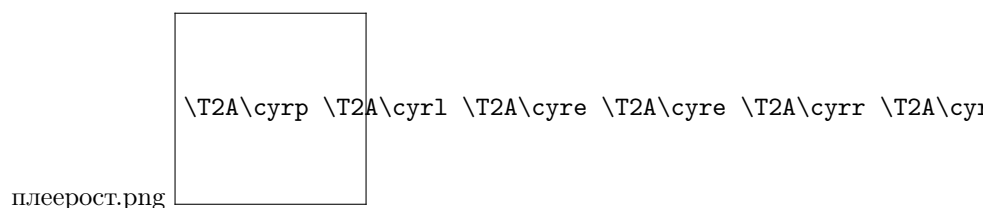
1. В приложении ROBIN Studio посредством нажатия кнопки «Стоп».

- Зайти в студию;
- Запустить робота;
- В момент, когда необходимо прервать работу робота, нажать на кнопку «Стоп». Она находится на верхней панели серого цвета. После ее нажатия робот прекратит выполнения алгоритма без ошибок.

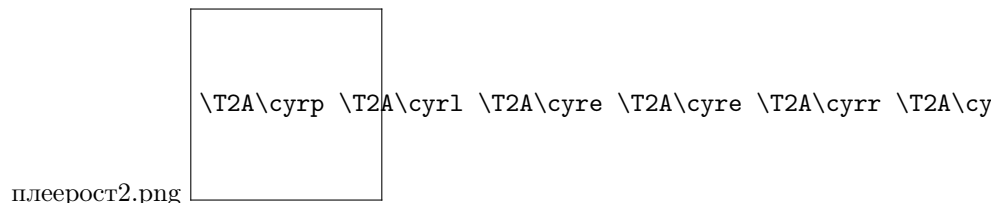


2. В приложении ROBIN Player посредством нажатия кнопки «Стоп».

- Открыть приложение ROBIN Player.
- На вкладке «Каталог» выбрать робота и запустить нажатием кнопки запуска.



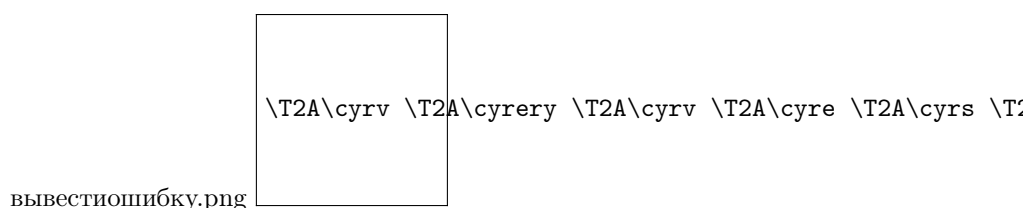
- На вкладке «Запущенные» отображается исполняемый в данный момент робот. Справа доступны кнопки «Пауза» и «Стоп». При нажатии на кнопку «Стоп» будет выполнена принудительная остановка работы робота.



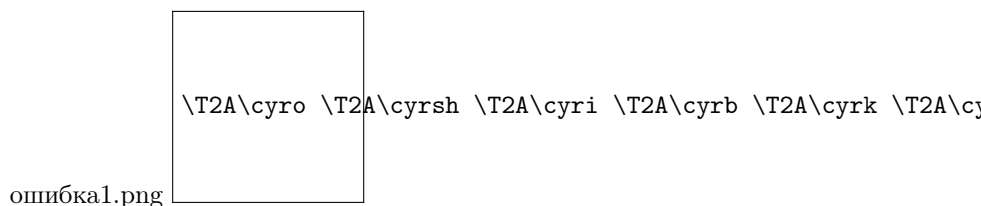
При нажатии на эту кнопку робот завершит свою работу без ошибки.

3. С помощью действия студии «Вывести ошибку».

- Из группы действий «Базовые действия» выбрать действие «Вывести ошибку» и перенести его в схему робота.



- В параметрах действия в строке «Сообщения» указать текст ошибки, который будет выведен в лог. Если оставить это поле пустым, то в логе будет выведена «Неизвестная ошибка».



- После настройки действия запустить робота нажатием кнопки запуска.
- Когда робот дойдет до действия «Вывести ошибку», в логе в студии появится запись об ошибке и робот завершит работу.

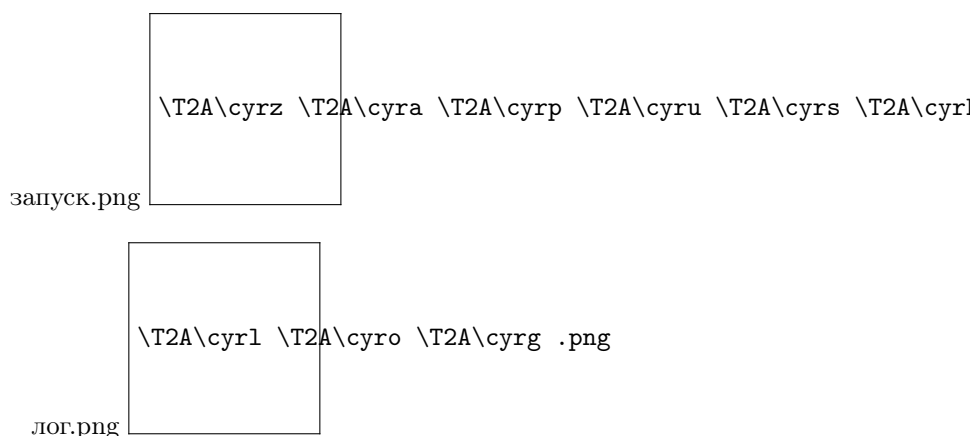
### 2.15.13 Запуск и работа роботов в консольном режиме

#### Когда необходим перевод машины в консольный режим

Есть несколько типов задач, которые программные роботы могут выполнить только при наличии активной графической сессии (т.е. не в фоновом режиме):

- Создание скриншота экрана
- Выполнение действий из группы «Имитация устройств»
- Выполнение некоторых действий их группы «Машинное зрение»: «Ждать исчезновения», «Ждать появления», «Существует на экране»
- Выполнение действий из подгруппы «Работа с win-элементами»

Если роботов запускают на машинах исполнения удаленно через приложение ROBIN Management Console (RMC), то проблем с выполнением таких задач не возникнет, т.к. там будет поднята графическая RDP-сессия. Однако если по каким-либо причинам запуск роботов планируется осуществлять вне RMC (вручную или по локальному расписанию) на удаленной машине, а окно с



RDP-соединением пользователь не планирует оставлять в открытом виде, необходимо перевести текущую RDP-сессию в режим console (консольный режим).

В консольном режиме графическая сессия остается активной, но при этом RDP-соединение закрыто и не мешает пользователю.

### Способы перевода машины в консольный режим

#### Вручную

1. Запустить робота на удаленной машине. Если запуск предполагается производить по расписанию, настроенному на этой машине, то этот шаг можно пропустить.
2. Открыть cmd и выполнить команду `query session`. Необходим номер сессии (ID) у текущего пользователя:

```
Администратор: Командная строка
Microsoft Windows [Version 10.0.19042.1110]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\WINDOWS\system32>query session
```

	ПОЛЬЗОВАТЕЛЬ	ID	СТАТУС	ТИП	УСТР-ВО
services		0	Диск		
rdp-tcp#38	AGolovleva	2	Активно		
console		3	Подключено		

Этот номер меняется при каждом новом подключении к машине, поэтому запоминать его для постоянного использования не имеет смысла.

3. Выполнить команду `tscon.exe <ID> /dest:console`

Подробнее: <https://support.smartbear.com/testcomplete/docs/testing-with/running/via-rdp/keeping-computer-unlocked.html>

#### При помощи робота

Выполнять самостоятельно все команды, которые описаны в первом способе, необязательно. Их точно также может выполнить и сам робот. Просто необходимо добавить в начало целевого алгоритма действия по исполнению этих команд через cmd. Но перед запуском робота нужно перезапустить ROBIN Agent от имени администратора. Закрывать RDP-соединение самостоятельно не надо, необходимо дождаться, пока это сделает сам робот.

#### При помощи bat-файла

Чтобы каждый раз не открывать cmd и не вводить команды вручную, можно просто запускать заранее созданный bat-файл, в котором уже указаны необходимые команды. Запуск файла стоит производить также от имени администратора и только после выполнения шага №1 из первого способа. ID сессии менять в файле не надо, его определение происходит автоматически. Этот же bat-файл может запускать и сам робот, при помощи действия «Исполнить скрипт» (пример заполнения

КонсольныйРежим/команда.png:alt:

действия в схеме, приложенной выше). Но перед запуском робота нужно перезапустить ROBIN Agent от имени администратора. Закрывать RDP-соединение самостоятельно не надо, необходимо дождаться, пока это сделает сам робот.

### Ограничения при использовании консольного режима

При запуске и работе роботов в консольном режиме нужно учитывать также несколько особенностей:

1. Чаще всего исполнение команды по переводу машины в консольный режим должно происходить от имени администратора (в редких случаях необходимая учетная запись уже может обладать необходимым уровнем прав).
2. В политиках удаленной машины должна быть отключена автоматическая блокировка экрана и переход в спящий режим при бездействии. Это требование должно быть выполнено, если время бездействия робота (например, между его запусками по расписанию) превышает установленный лимит бездействия, после которого учетная запись блокируется или переводится в спящий режим. Дело в том, что при заблокированной учетной записи, робот сможет запуститься, но корректно исполнит только те действия, которые выполняются в фоновом режиме. Самостоятельно же разблокировать учетную запись робот не сможет.
3. Необходимо, чтобы в консольном режиме разрешение экрана оставалось таким же, какое оно было при активной RDP-сессии, когда аналитик производил создание исходника робота (не влияет только при исполнении действия по созданию скриншота). Это правило должно всегда соблюдаться. Если разрешение при переходе в консольный режим все же меняется, можно попробовать немного модифицировать команду:

```
tscon.exe <ID> /dest:console & <Путь до QRes.exe> /x:<Значение по X> /y:<Значение по Y>
```

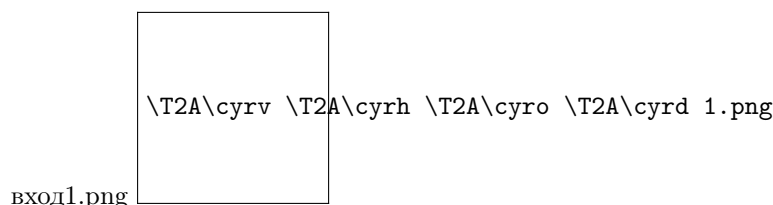
где первая команда - это переход в консольную сессию, вторая - установка разрешения при помощи специализированной утилиты. Исходники утилиты здесь: <https://sourceforge.net/projects/qres/>

Но здесь следует учитывать, что на некоторых виртуальных средах после отключения RDP-соединения может не сохраниться то разрешение, с которым работал пользователь. В консольном режиме может происходить автоматическая установка того разрешения, с каким в последний раз заходили администраторы в консоль управления виртуальной средой.

Такая проблема, например, существует в виртуальных средах VMware. Предлагаемое решение описано по ссылке: VM Console Display Resolution Change - vGemba.net: <https://www.vgemba.net/vmware/powercli/VMConsole-Resolution-Change/>

То есть необходимо:

а) войти на необходимую машину через консоль управления и поменять на машине разрешение на необходимое:



б) затем прописать в настройках виртуальной машины (через консоль управления) следующие параметры:

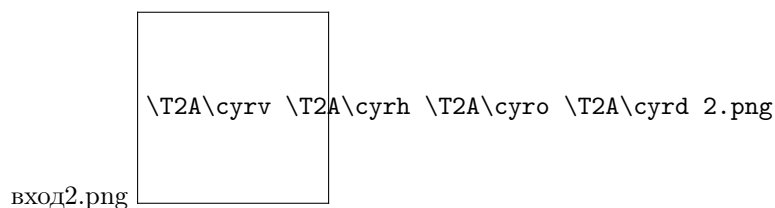
```
guestInfo.svga.wddm.modeset="FALSE"
```

```
guestInfo.svga.wddm.modesetCCD="FALSE"
```

```
guestInfo.svga.wddm.modesetLegacySingle="FALSE"
```



guestInfo.svga.wddm.modesetLegacyMulti="FALSE"



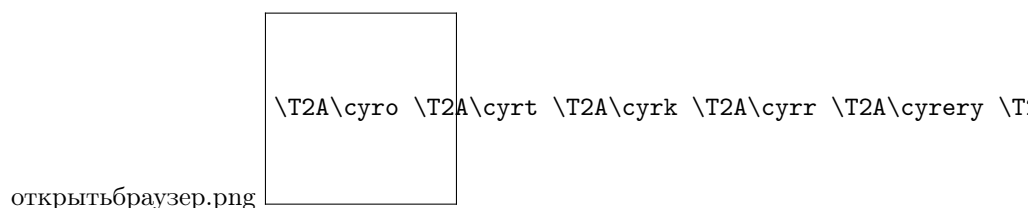
В таком режиме манипуляции с размером консоли VMware не влияют на разрешение, которое будет на машине в консольном режиме при работе робота.

### 2.15.14 Запуск с профилем для всех браузеров

Для решения задач, где требуется работа в браузере под определенной учетной записью, существует возможность запуска браузера с профилем. В Robin Studio существует возможность работы с такими браузерами, как «Яндекс.Браузер», «Chrome», «MozillaFirefox» и «Internet Explorer». Шаги для решения этой задачи описаны ниже:

#### Браузер «Chrome»

1. Из группы действий «Браузер» выбрать действие «Открыть браузер».



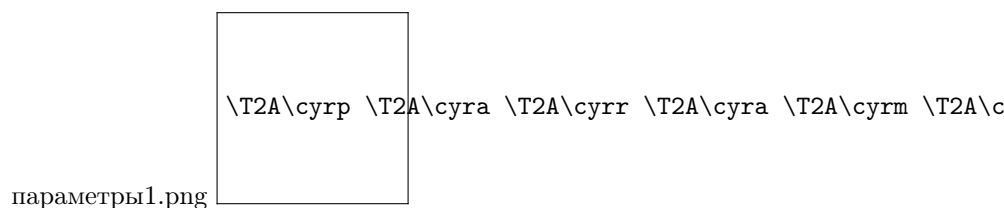
2. В параметрах действия в строке «Браузер» выбрать «Chrome».
3. В строке «Свойства» указать в JSON формате, описывающий список свойств браузера и их значений, которые будут установлены при запуске.

Шаблон:

```
{«acceptInsecureCerts»:true,«browserName»:»chrome»,«goog: loggingPrefs»: { «browser»:»ALL» }, «timeouts»: { «implicit»: 30000, «pageLoad»: 30000, «popup-blocking»: 30000, «disable-notifications»: 30000, «no-sandbox»: 30000, «user-data-dir»: %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default }, «extensions»: [] }
```

4. %USERPROFILE%/AppData/Local/Google Chrome/User Data/Default необходимо заменить на актуальный путь.

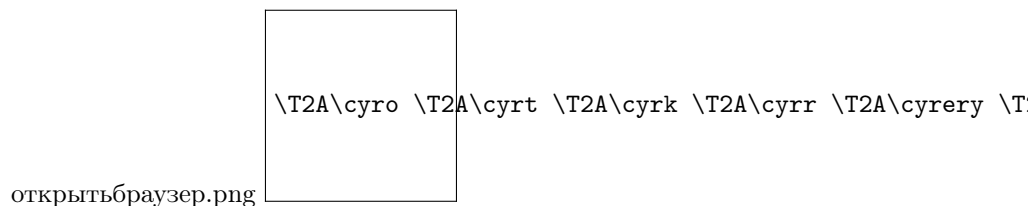
Пример заполнения полей:



5. Запустить робота. Браузер откроется с профилем.

## Яндекс.Браузер

1. Из группы действий «Браузер» выбрать действие «Открыть браузер».

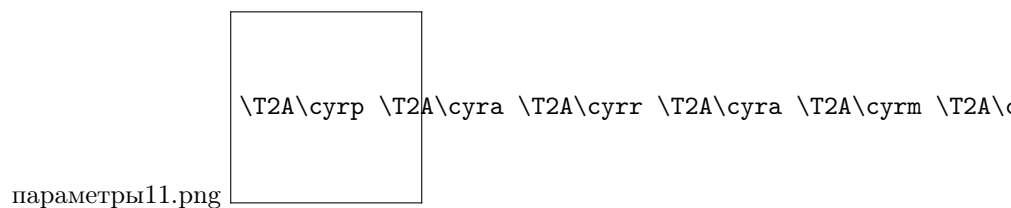


2. В параметрах действия в строке «Браузер» выбрать «Yandex».
3. В строке «Свойства» указать в JSON формате, описывающий список свойств браузера и их значений, которые будут установлены при запуске.

Шаблон: {"acceptInsecureCerts":true,"browserName":"chrome","goog:loggingPrefs":{"browser":"ALL"},"timeouts":{"implicit":0,"pageLoad":300003,"script":30003},"unhandledPromptBehavior":"ignore","disable-notifications","no-sandbox","user-data-dir":"%USERPROFILE%\AppData\Local\Yandex\YandexBrowser\User Data"},"extensions":[]}

4. %USERPROFILE%/AppData/Local/Yandex/YandexBrowser/User Data необходимо заменить на актуальный путь.

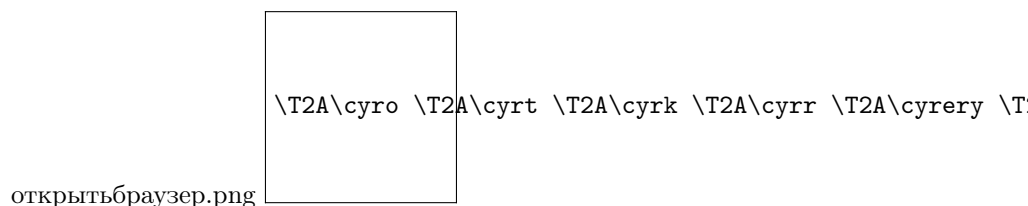
Пример заполнения полей:



5. Запустить робота. Браузер откроется с профилем.

## Браузер «MozillaFirefox»

1. Из группы действий «Браузер» выбрать действие «Открыть браузер».

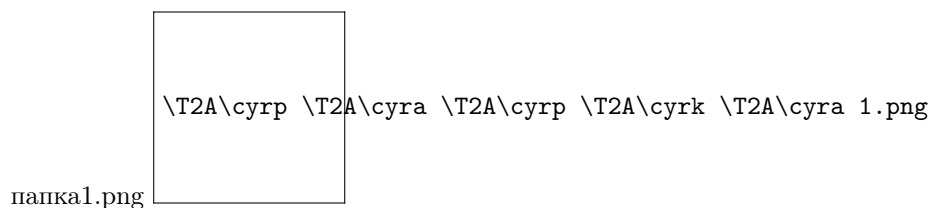


2. В параметрах действия в строке «Браузер» выбрать «Firefox».
3. В строке «Свойства» указать в JSON формате, описывающий список свойств браузера и их значений, которые будут установлены при запуске.

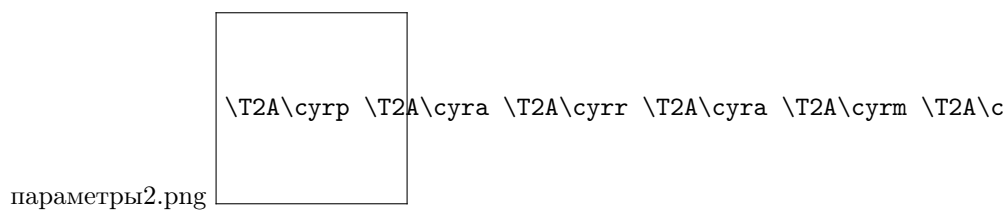
Шаблон: {"acceptInsecureCerts":true,"browserName":"firefox","timeouts":{"implicit":0,"pageLoad":300003,"script":30003},"args":["-profile","%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\ProfileDir"],"prefs":{}}

4. %USERPROFILE%/AppData/Roaming/Mozilla/Firefox/Profiles/ProfileDir требуется заменить на актуальный путь до папки с профилем.

На скриншоте изображен пример папки с профилем:



Пример заполнения полей:



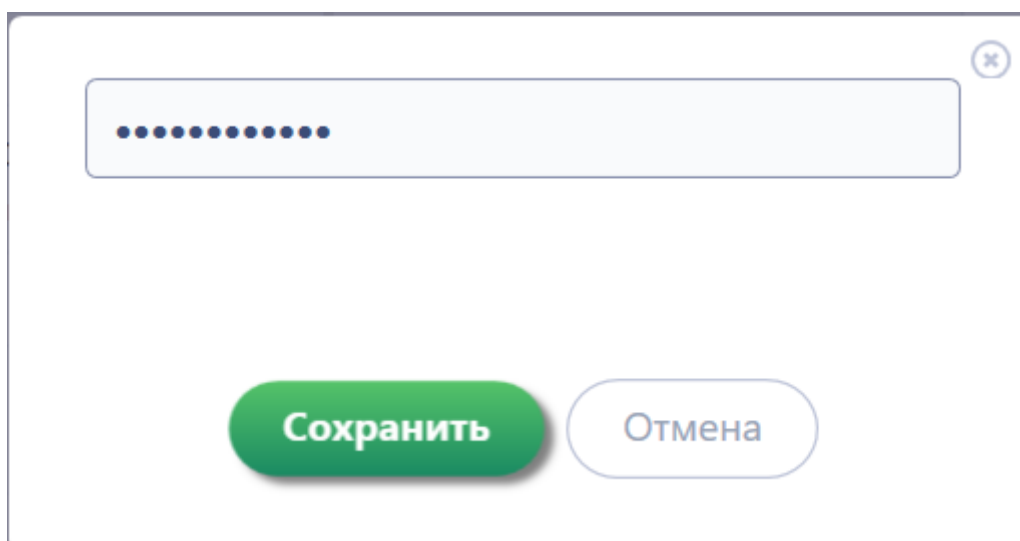
### Браузер «Internet Explorer»

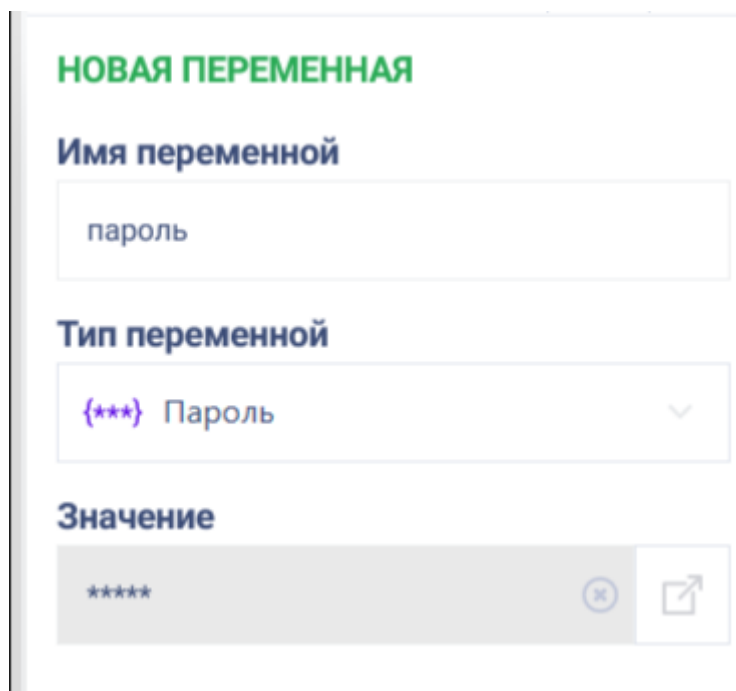
У браузера Internet Explorer нет профилей, соответственно отсутствуют параметры для такого варианта запуска.

## 2.15.15 Хранение и использование паролей в сценариях

### Хранение паролей в переменной типа «Пароль»

В Robin Studio существует возможность создания переменной с типом «Пароль» для хранения паролей. При создании переменной пароль вводится в поле «Значение», где сам текст пароля будет скрыт. Пример:





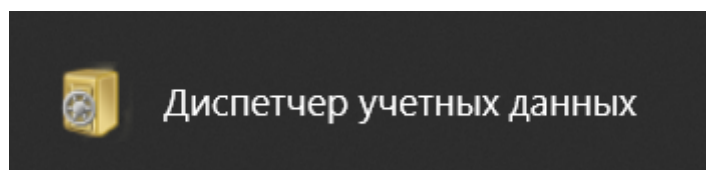
После сохранения в поле «Значения» пароль будет отображаться в виде звездочек:

Значение пароля, сохраненное в переменной, не отображается в логе студии, его нельзя получить посредством вывода текста или иным способом.

Шифрование происходит с помощью AES шифрования. AES – это симметричный алгоритм блочного шифрования, принятый в качестве стандарта шифрования, который оперирует блоками по 128 бит. Это значит, что AES берет 128 бит исходного сообщения и превращает их с помощью некоего ключа в 128-битный шифротекст. Размер ключа может быть 128, 192 или 256 бит.

### Получение пароля из хранилища

Пароли, находящиеся в хранилище, можно посмотреть в диспетчере учетных данных Windows.

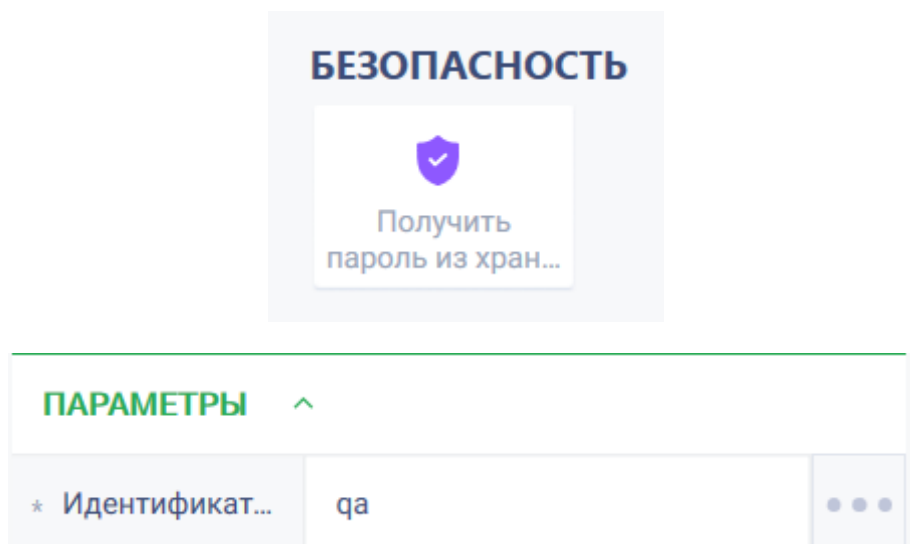


Получить информацию оттуда, можно с помощью действия «Получить пароль из хранилища» из группы действий «Безопасность».

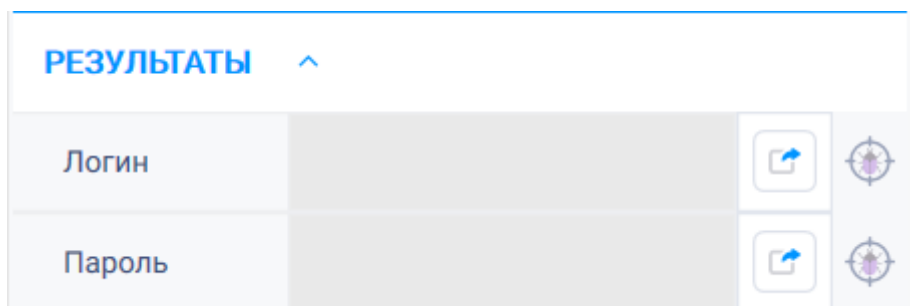
Для этого необходимо:

1. Перенести действие «Получить пароль из хранилища» на рабочую область.
2. Заполнить параметры действия: а. «Идентификатор» - указать идентификатор записи с учетными данными в диспетчере задач учетных данных Windows.

Пример:



3. Результатом действия является логин и пароль, которые можно использовать для работы робота.



### 2.15.16 Конфигурация Internet Explorer

Для работы Программного робота необходимо сконфигурировать браузер Internet Explorer. Ссылка на источник: <https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver>

Для этого необходимо выполнить следующие шаги:

1. Для IE 7 и выше на Windows Vista или Windows 7. Необходимо установить Защищенный режим для каждой зоны в одинаковом положении – вкл./ выкл.

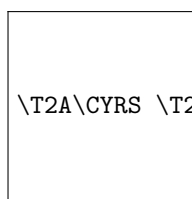
Для этого перейдите в Настройки → Свойства браузера → Вкладка «Безопасность» → Для каждой зоны поставить/ убрать галочку напротив „Включить защищенный режим“.

2. Для IE 10 и выше. Настройка «Включить расширенный защищенный режим» должна быть отключена. Эту опцию можно найти в Свойствах браузера → Вкладка „Дополнительно“ → Пункт „Безопасность“.

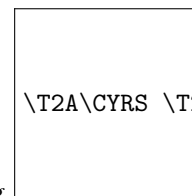
3. Уровень масштабирования браузера должен быть установлен на 100%, чтобы события мыши могли быть установлены в правильные координаты.

Чтобы проверить масштабирование, необходимо открыть Настройки → Масштаб → Установить значение в 100%

СвойстваБраузера.png



СвойстваБраузера2.png



4. Для Windows 10. Необходимо установить «Изменение размера текста, приложений и других элементов» в значении 100%.

Откройте Пуск → Настройки → Дисплей → «Изменение размера текста, приложений и других элементов» установить в значении 100%.

5. Только для IE 11. Нужно установить запись реестра на целевом компьютере, чтобы драйвер браузера мог поддерживать соединение с экземпляром создаваемого Internet Explorer.

Для 32-битной Windows:

Открываем редактор реестра: ПКМ по Пуск → Выполнить → вводим regedit в поле Открыть и нажимаем ОК. Далее переходим в раздел HKEY\_LOCAL\_MACHINE → SOFTWARE → Microsoft Internet Explorer → Main → FeatureControl → FEATURE\_BFCACHE. В случае, если раздела FEATURE\_BFCACHE нет, его необходимо создать. ПКМ по FeatureControl → Создать → Раздел Внутри раздела создайте „Параметр DWORD (32 бита)“ с именем „iexplore.exe“ со значением 0.

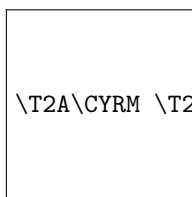
Для 64-битной Windows:

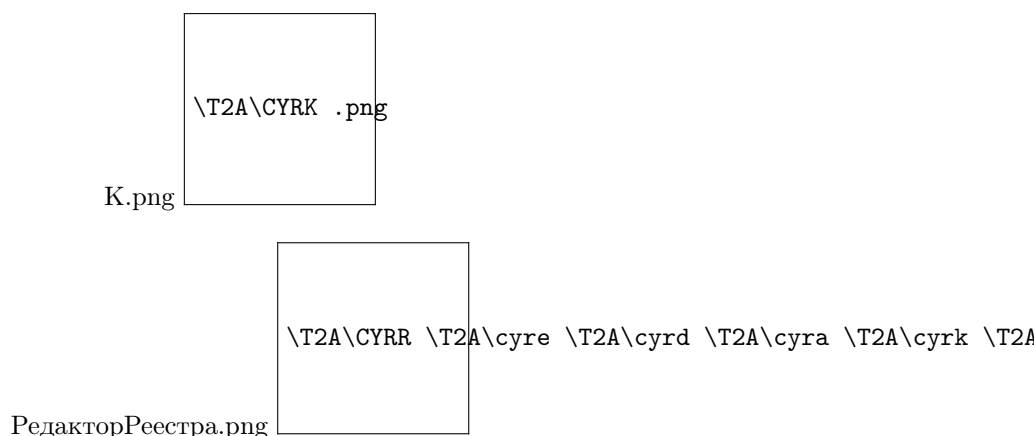
Открываем редактор реестра: ПКМ по Пуск → Выполнить → вводим regedit в поле Открыть и нажимаем ОК. Далее переходим в раздел HKEY\_LOCAL\_MACHINE → SOFTWARE → Wow6432Node → Microsoft → Internet Explorer → Main → FeatureControl → FEATURE\_BFCACHE. В случае, если раздела FEATURE\_BFCACHE нет, его необходимо создать. ПКМ по FeatureControl → Создать → Раздел Внутри раздела создайте „Параметр DWORD (32 бита)“ с именем „iexplore.exe“ со значением 0.

### 2.15.17 Передача параметров между роботами

При запуске одного Программного робота через другого иногда требуется передача некоторых параметров.

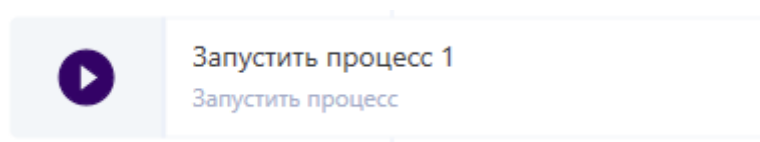
Масштабирование.png





### Работа с первым роботом

В первом роботе, который передает параметр, необходимо использовать действие «Запустить процесс».



В разделе «Свойства» (блок «Параметры») необходимо заполнить поля:

- Задать - выбрать значение «Имя» или «Путь к файлу» (какое приложение запускаем);
- Файл процесса (производится запуск плеера)
- Аргументы - шаблон аргумента: -г «путь до готового робота» -р «путь до файла с параметрами»

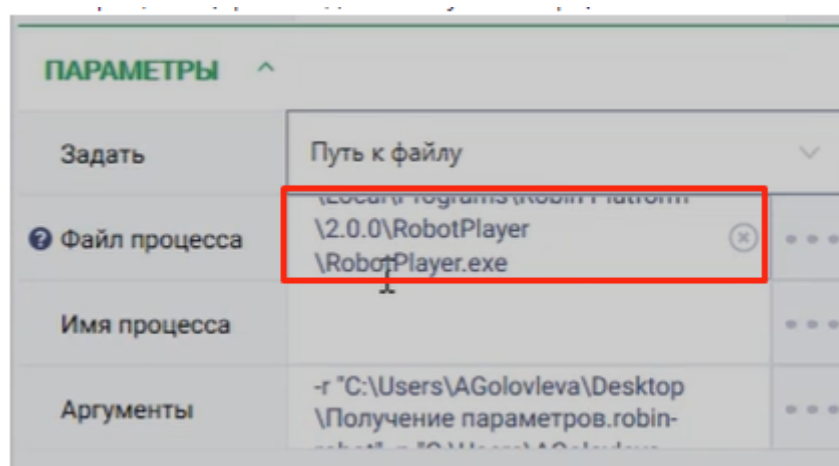
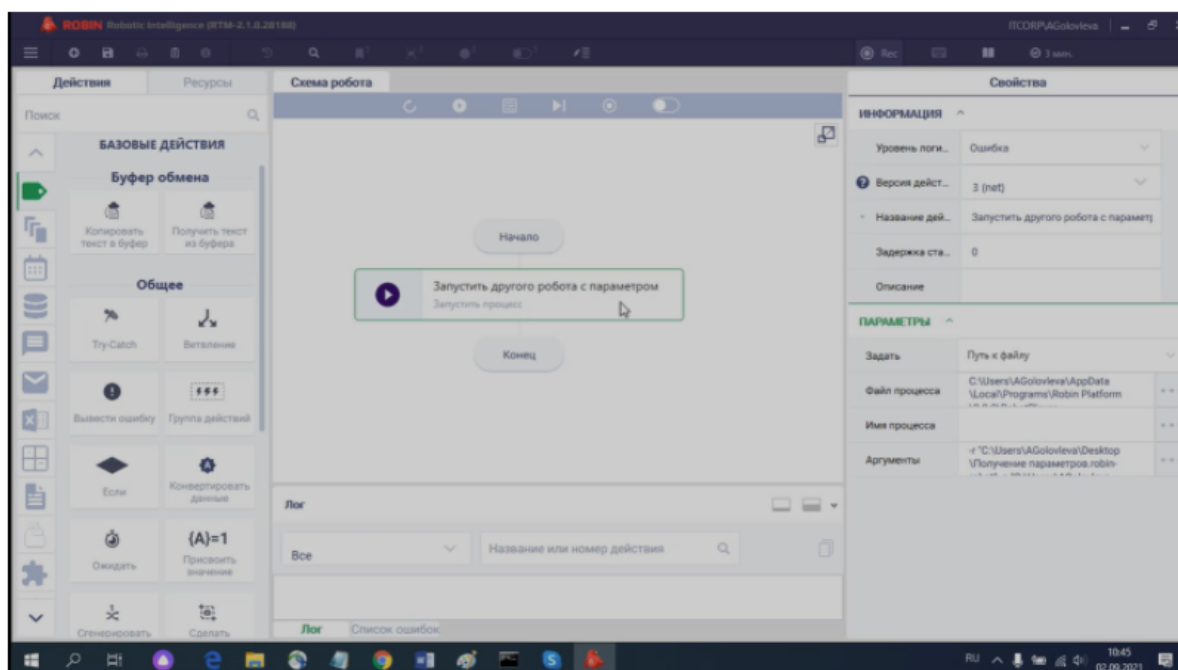
Шаблон файла с параметрами:

В шаблоне необходимо изменить значение, которое нужно передать второму роботу и название переменной в которую это значение запишется у второго робота. Изменения необходимо внести в параметр InOutResource.

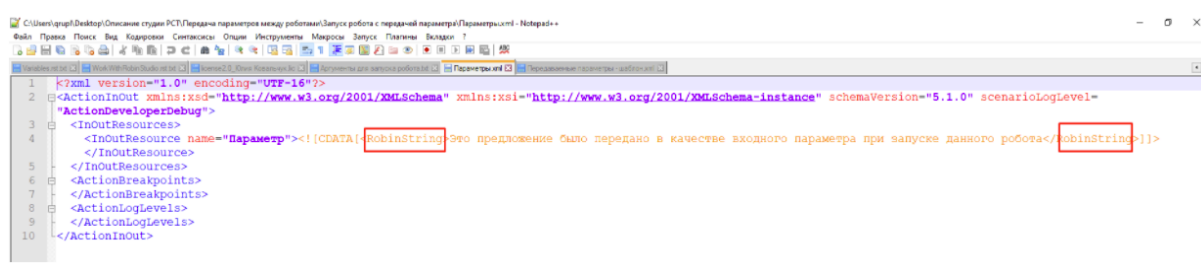
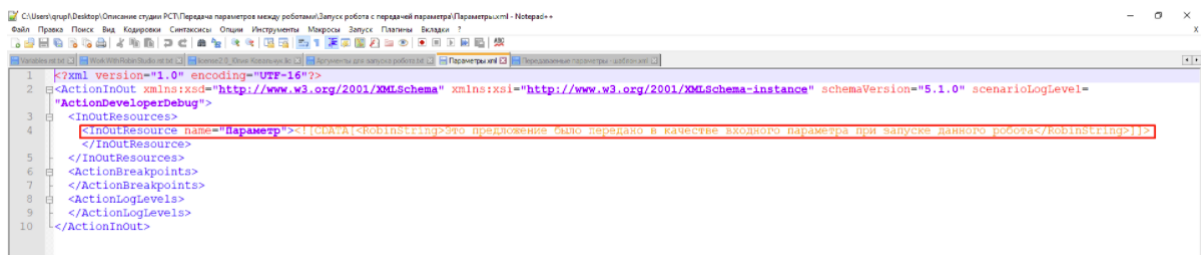
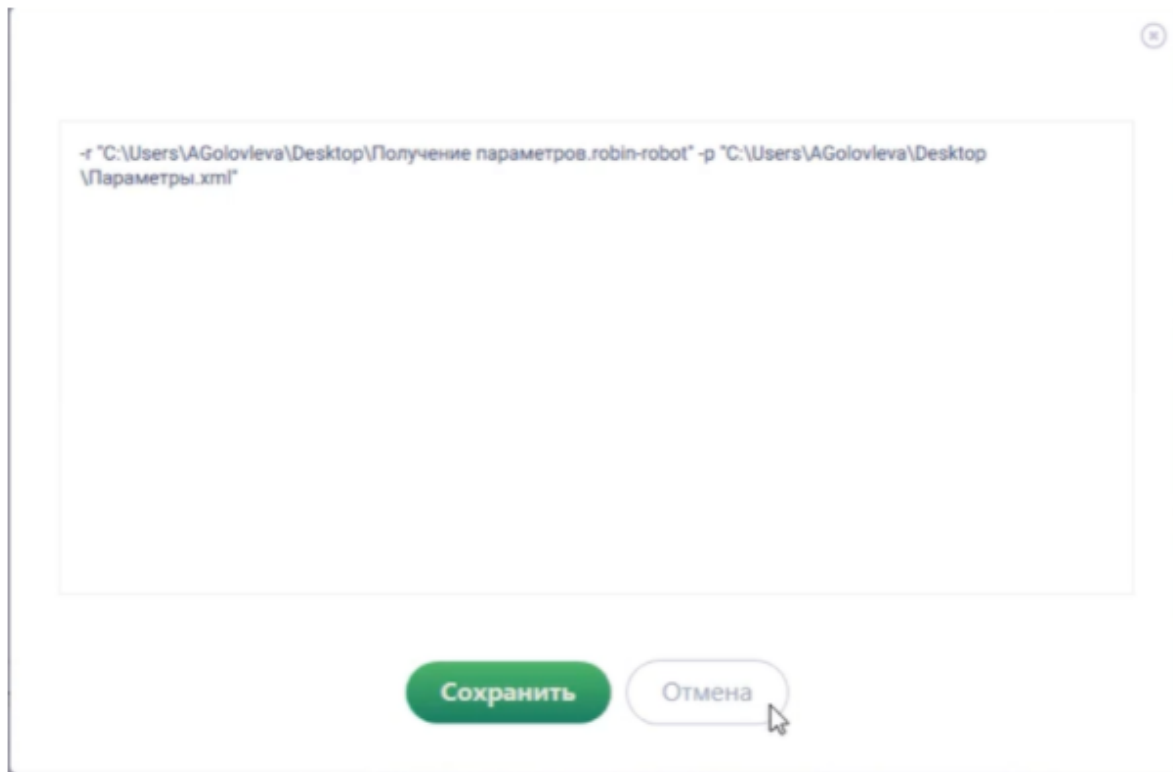
Далее, необходимо указать тип переменной. В данном случае указана переменная «Строка» (RobinString).

Ниже перечислен формат записи для указания типа передаваемого значения:

- RobinBoolean - Логическая переменная
- RobinCollection - переменная типа Коллекция
- RobinDataTable - переменная типа Таблица
- RobinDateTime - переменная типа Время/Дата
- RobinDictionary - переменная типа Словарь
- RobinFilePath - переменная типа Путь к файлу
- RobinFolderPath - переменная типа Путь к папке
- RobinHotKeys - переменная типа Горячие клавиши

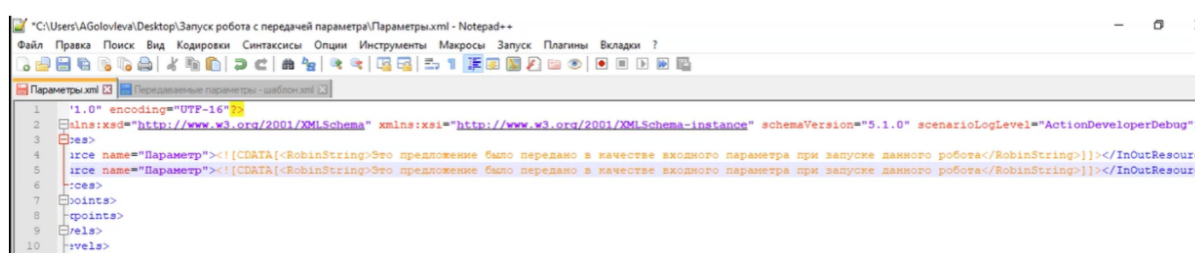






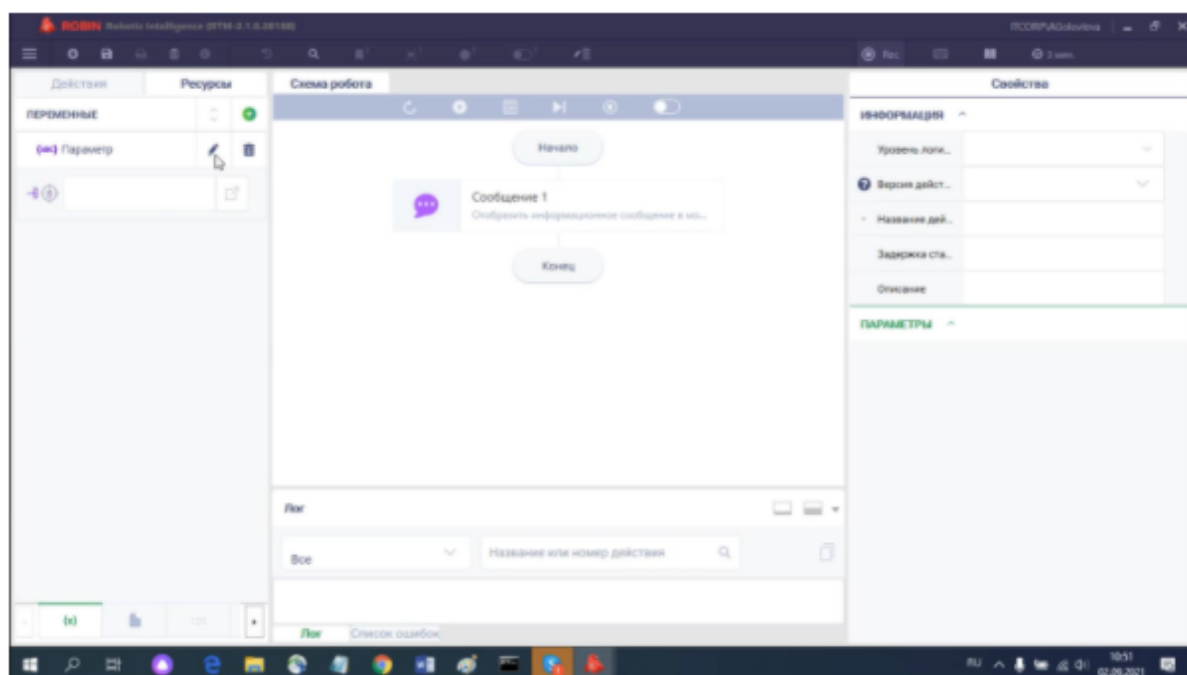
- RobinString - переменная типа Строка
- RobinPassword - переменная типа Пароль
- RobinNumeric - переменная типа Число
- RobinLocationPoint - переменная типа Координаты точки
- RobinObject - переменная типа Объект
- RobinRectangleLocation - переменная типа Область

При передаче сразу нескольких значений необходимо добавить эти значения строкой ниже.

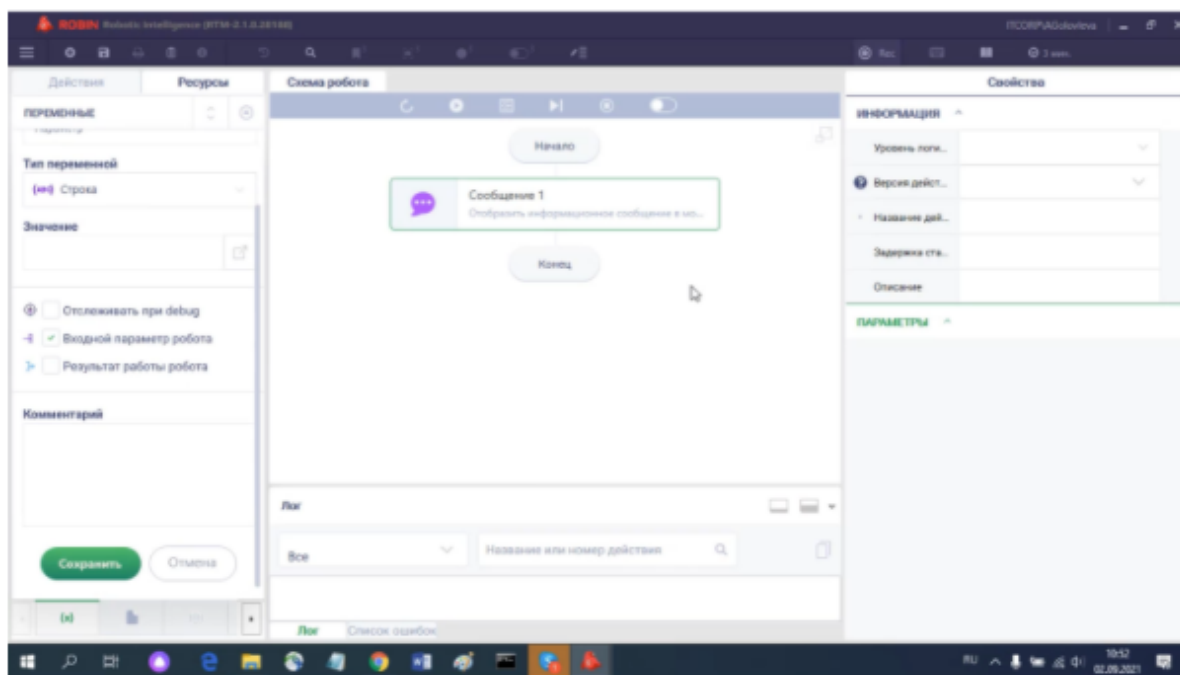


## Работа с вторым роботом

Необходимо зайти во второго робота (которому нужно передать параметры) и создать те переменные, которые ранее были указаны в xml-файле.



Необходимо указать в переменной значение «Входной параметр робота». Далее произвести запуск робота.

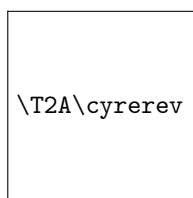


### 2.15.18 Запуск робота из робота

Для решения некоторых задач требуется запуск одного робота в ходе работы другого робота. Шаги по выполнению данного процесса описаны ниже:

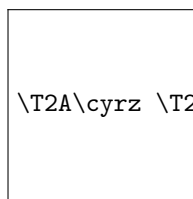
1. Экспортировать готового робота, которого необходимо запустить с помощью другого робота (расширение «.robin-robot»):

экспорт.png



2. В схеме робота, из которого будет запущен другой робот, выбрать действие «Запустить процесс» из группы действий «Процессы».

запускпроцесса.png



3. В параметрах действия в строке «Задать процесс» выбрать «По пути к файлу» из выпадающего списка. В строке файл процесса указать путь к экспортированному ранее готовому роботу.

### 2.15.19 Запуск робота с параметром

Для решения некоторых задач требуется передача каких-либо параметров роботу. Решить данную задачу можно посредством запуска робота из робота, запуском робота с помощью приложения ROBIN Robot Player и с помощью командной строки.

параметры3.png

\T2A\cyrp \T2A\cyra \T2A\cyrr \T2A\cyra \T2A\cyrm \T2A\c

**Запуск робота из робота с передачей параметров**

1. В роботе, в которого будет передан параметр, необходимо создать переменную, в которую будет записан параметр и выбрать опцию «Входной параметр робота»:

окнопеременные.png

\T2A\cyro \T2A\cyrk \T2A\cyrn \T2A\cyro \T2A\cyrp \T2A\

2. Экспортировать готового робота, которого необходимо запустить с помощью другого робота (расширение «.robin-robot»):

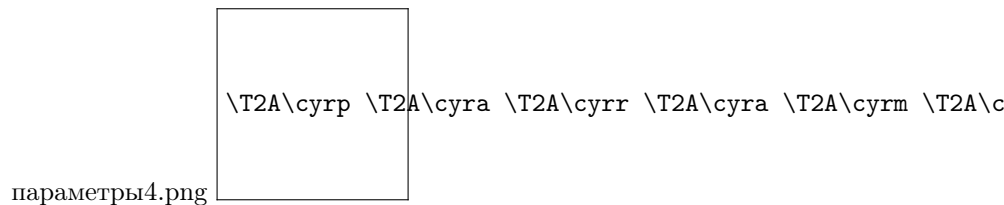
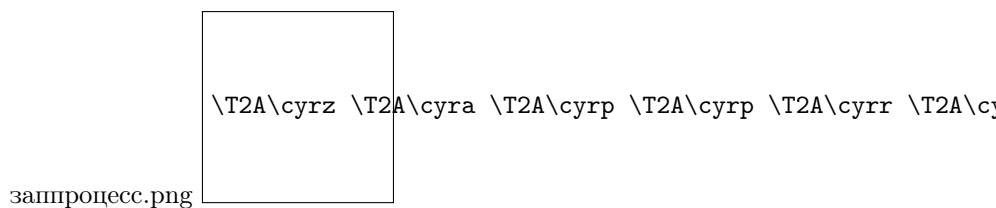
выгрузка.png

\T2A\cyrv \T2A\cyrery \T2A\cyrg \T2A\cyrr \T2A\cyru \T2A\

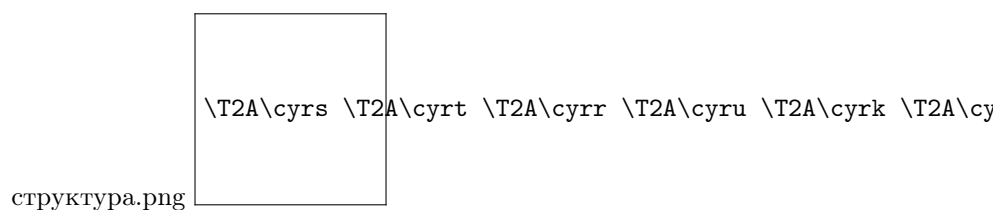
3. В схеме робота, из которого будет запущен другой робот, выбрать действие «Запустить процесс» из группы действий «Процессы».
4. В параметрах действия в строке «Задать процесс» выбрать «По пути к файлу» из выпадающего списка.
5. В строке «файл процесса» указать путь до приложения ROBIN Robot Player (%USERPROFILE%/AppData/Local/Programs/RobinPlatform/2.0.0/RobotPlayer/RobotPlayer.exe).
6. В поле «Аргументы» указать строку следующего формата: -r «путь до готового робота» -p «путь до файла с параметрами».

Параметры передаются посредством xml-файла. Такой xml-файл может иметь любое название, но порядок и название его элементов должны быть неизменны. Пользователь изменяет значения только в строке с элементом InOutResource:

- значение атрибута «name» – это название переменной, в которую передаем параметр. Эта переменная должна быть создана именно в том роботе, которому передаем параметр. Робот, который передает параметр, может и не иметь такой переменной;
- в значении элемента InOutResource необходимо указать тип переменной, в которую будет записан передаваемый параметр, и передаваемое значение, т.е. сам параметр;
- если роботу требуется передать несколько параметров, нужно скопировать элемент InOutResource необходимое количество раз и записать в него значения параметров.



### Структура xml-файла



### Запуск работа с помощью приложения ROBIN Player

1. Загрузить файл готового робота (расширение «.robin-robot») в ROBIN Player. Для загрузки робота нужно:

а. Нажать на кнопку добавления роботов:

2. Выбрать робота в каталоге.

2. Выбрать настройку расписания загруженного робота.

3. Включить параметры запуска.

4. Выбрать xml-файл с параметрами для этого робота.

Робот будет запущен по расписанию.

### Запуск работа посредством командной строки

1. Экспортировать готового робота, которому будут переданы параметры (расширение «.robin-robot»).

2. Подготовить xml-файл с параметрами, которые будут переданы роботу.

\T2A\cyrz \T2A\cyra \T2A\cyrg 1.png

zar1.png

\T2A\cyrz \T2A\cyra \T2A\cyrg 2.png

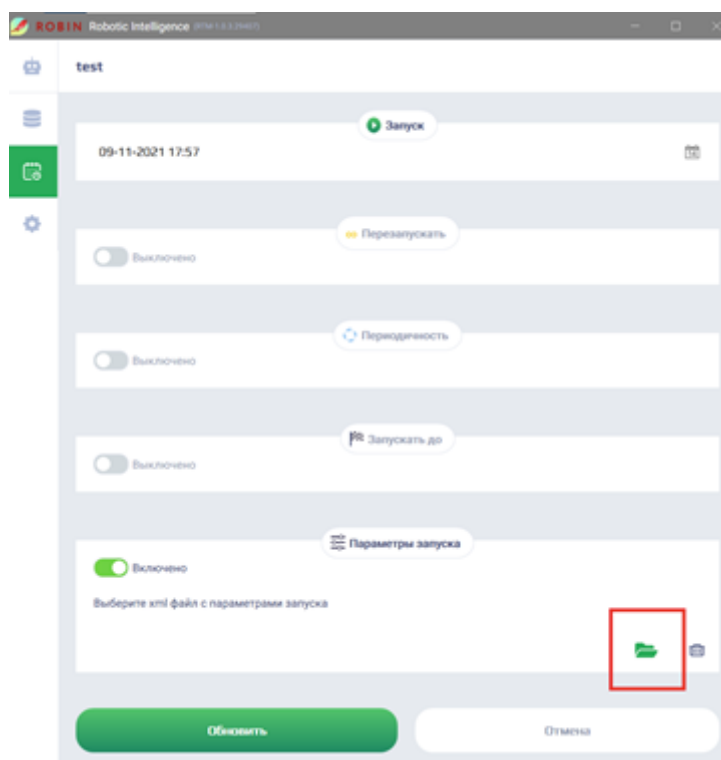
zar2.png

\T2A\cyrv \T2A\cyrery \T2A\cyrb \T2A\cyro \T2A\cyrr \T

выборнастройки.png

\T2A\cyrrz \T2A\cyra \T2A\cyrrp 1.png

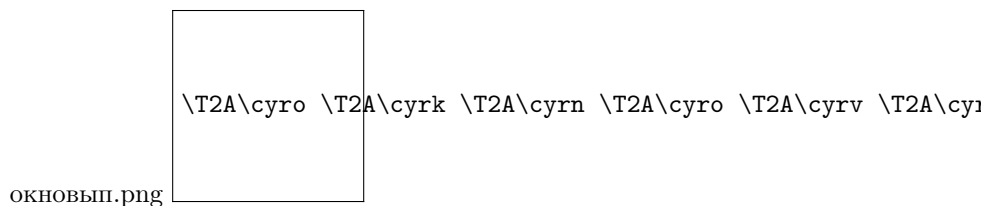
зап1.png



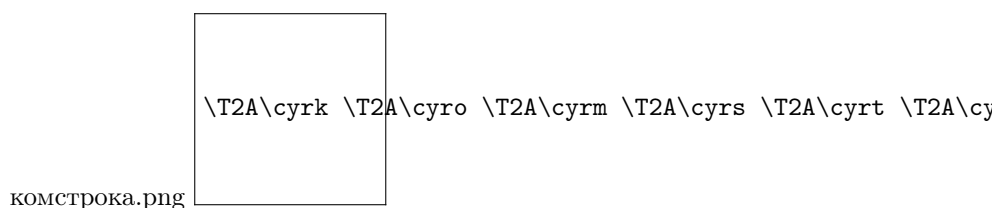
\T2A\cyrrerev	\T2A\cyrk	\T2A\cyrs	\T2A\cyrp	\T2A\cyro	\T2A\cyrorev
---------------	-----------	-----------	-----------	-----------	--------------

экспорт1.png

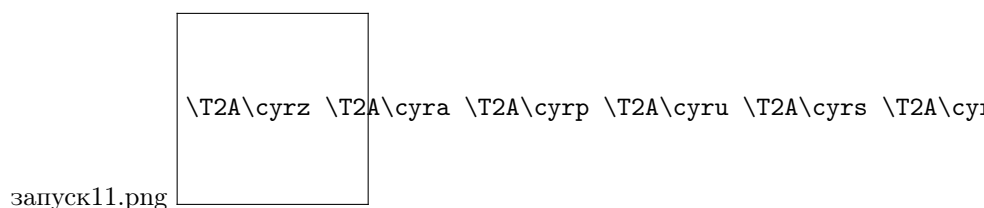
3. Открыть командную строку, нажав комбинацию клавиш Win+R и в появившемся окне ввести cmd, после чего нажать Enter (также командную строку можно открыть, введя в поиске windows cmd.exe).



4. В командной строке ввести следующее: «%USERPROFILE%/AppData/Local/Programs/RobinPlatform/2.0.0/RobinPlatform.exe -g «путь до готового робота» -p «путь до файла с параметрами»».



После этого начнется выполнение программного робота. В приложении ROBIN Robot Player на вкладке «Запущенные» появится информация о выполнении запускаемого робота:



## 2.15.20 Действия по работе с приложениями 1С

### Общая информация

Платформа ROBIN поддерживает несколько вариантов работы с приложениями 1С:

1. Через графический интерфейс. В этом случае робот имитирует поведение пользователя и работает с приложением используя действия групп «Машинное зрение», «Windows», «Имитация устройств».
2. С помощью специальных действий по группы «1С». Действия этой групп позволяют работать с приложениями 1С не через графический интерфейс, окна и кнопки, а используя программные интерфейсы 1С.

Такой способ работы является более надежным и предпочтительным для обработки больших массивов данных.

### Включение oData на стороне сервера:

Одним из способов взаимодействия программных роботов с 1С через программные интерфейсы, является протокол oData. Это стандартизированный протокол работы с данными, в основе которого

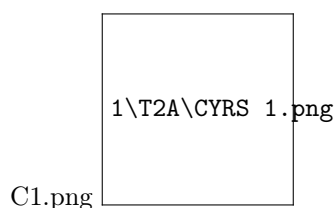
лежит REST. 1С активно развивает его с версии 8.3.5 и на сегодняшний день этот протокол является наиболее полным и стандартизованным, простым и безопасным способом работы с 1С\*.

\*<https://infostart.ru/1c/articles/711302/>

<https://infostart.ru/public/403426/>

Для того, чтобы роботом иметь возможность использовать стандартные методы 1С, необходимо включить oData на стороне сервера, для этого необходимо выполнить шаги:

1. Запустить 1С от имени администратора.
2. В пункте меню «Администрирование» выбрать «Публикация на веб-сервере».
3. Включить «Публиковать стандартный интерфейс OData».
4. Нажать «Опубликовать».



5. После этого нужно будет определить состав интерфейса oData, т.е. указать — какие объекты конфигурации в него входят, а какие нет (изначально в составе нет ни одного объекта).
6. Сформировать ссылку на опубликованный web-сервис можно по шаблону:  
<http://<имя|АдресСервера1С>/<ИмяБазы1С>/odata/standard.odata>
7. Все, теперь вы можете обращаться через программного робота в 1с.





## 2.15.21 Комбинации горячих клавиш

Горячие клавиши	Действие	Использование
Ctrl + N	Создать новый проект	Для создания нового проекта нажмите Ctrl + N. Студия запросит решение о сохранении открытой схемы. После будет открыто окно выбора проектов, в котором можно будет создать новый проект и начать работу с ним.
Ctrl + S	Сохранить открытую схему	Для сохранения текущей схемы нажмите Ctrl + S
Ctrl + O	Опубликовать схему в хранилище	Для публикации текущей схемы нажмите Ctrl + O
Ctrl + Shift + O	Публикация готового робота	Для публикации готового робота нажмите Ctrl + Shift + O
Ctrl + Shift + E	Экспорт готового робота	Для экспорта готового робота нажмите Ctrl + Shift + E
ЛКМ	Переместить действие на схеме	Для того чтобы переместить действие (или выделенную группу действий) в другую часть схемы, необходимо выделить действие кликом ЛКМ и сохраняя нажатие переместить его в нужную часть схемы. Группа действий на новом месте воссоздаст порядок, с которым действия были выделены.
Ctrl + клик ЛКМ	Выделить несколько действий схемы	Для того, чтобы выделить несколько действий необходимо зажать Ctrl и выполнить клик ЛКМ по нужным действиям.
Shift + клик ЛКМ	Выделить отрезок схемы	Для того, чтобы выделить отрезок схемы на рабочей области, необходимо зажать Shift и выполнить клик ЛКМ по первому и последнему действию нужного отрезка.
Ctrl + A	Выделить все действия на схеме	Для выделения всех действий на рабочей области необходимо нажать Ctrl + A
Ctrl + C	Копировать в буфер	Для копирования в буфер ранее выделенного действия или группы действий необходимо нажать Ctrl + C
Ctrl + V	Вставить из буфера	Для вставки из буфера скопированного ранее действия или группы действий необходимо нажать Ctrl + V. Действия будут добавлены после выделенного действия, если такое есть на схеме, или в ее конец, если выделенных действий нет.
Ctrl + X	Вырезать из схемы	Для вырезания ранее выделенного действия или группы действий необходимо нажать Ctrl + X
Delete	Удалить из схемы	Для удаления ранее выделенного действия или группы действий необходимо нажать Delete.
Ctrl + ЛКМ	Переместить рабочую область относительно окна просмотра	Для того чтобы переместить рабочую область в окне просмотра, необходимо зажать Ctrl + ЛКМ на белом поле области и переместить ее в нужную сторону.
Ctrl + скролл	Изменение масштаба отображения	Для того чтобы изменить масштаб отображения рабочей области, зажмите клавишу Ctrl и прокручивайте скролл в нужном направлении, увеличивая или уменьшая масштаб. Позволяет изменять масштаб отображения выбранного действия схемы, сохраняя его в центре экрана.
Alt	Выбор области приближения	Чтобы выделить область, которая будет увеличена во все окно, нажмите Alt и появившимся указателем выделите нужную область.

---

ROBIN Studio. Описание действий

---

---

## ROBIN Robot. Руководство пользователя

---

### 4.1 Общая информация

**ROBIN Robot 2.0** - пользовательское приложение, входящее в состав платформы Robin и предназначенное для локального запуска и контроля программных роботов, созданных с помощью ПО «Robin Studio».

Является обязательным компонентом для запуска и исполнения программных роботов. Объектом лицензирования является рабочее место = возможность запуска роботов под одной учетной записью на одной вычислительной мощности (это может быть, как рабочий компьютер пользователя, так и сервер - физический или виртуальный).

ПО «ROBIN Robot» представлено тремя редакциями:

- «ROBIN Robot. Desktop Edition» - предназначено для запуска на рабочей станции, не подключенной к корпоративной сети,
- «ROBIN Robot. Enterprise Edition» - предназначено для запуска на рабочей станции, подключенной к корпоративной сети,
- «ROBIN Robot. Cloud Edition» - предназначено для запуска на рабочей станции, расположенной в облаке.

**ROBIN Robot состоит из двух модулей: ROBIN Agent и Robin Player:**

- ROBIN Agent -программный модуль, который обеспечивает подключение компонента ROBIN Orchestrator к ПК/серверу для запуска роботов. ROBIN Agent получает от оркестратора команду на выполнения роботов и передает «логи» (информацию об исполнении) роботов оркестратору в режиме реального времени.
- ROBIN Player-модуль исполняющий сценарий робота. ROBIN Playerпредоставляет возможность запускать роботов, созданных в ROBIN Studio (файлы .robin-robot), в соответствии с различными сценариями: по настроенному расписанию, по событиям в информационных системах, по нажатию горячих клавиш или по команде, полученной от оркестратора(через ROBIN Agent).

**ROBIN Robot 2.0 обладает следующими функциональными возможностями:**

- ручной запуск и остановка программных роботов;
- настройка расписания для запуска/перезапуска/периодичности запуска программных роботов;

- настройка отправки писем при завершении выполнения процесса;
- загрузка программных роботов в каталог;
- просмотр истории запусков;
- просмотр логов роботов.

Одна лицензия ROBIN Robot позволяет исполнять неограниченное количество сценариев роботов на одном рабочем месте. Доступен как последовательный, так и параллельный запуск сценариев. Параллельное исполнение сценариев возможно, если алгоритм может быть выполнен в фоновом режиме (например, без необходимости работы с интерфейсом desktop-приложений или машинным зрением).

## 4.2 Условия эксплуатации ROBIN Robot 2.0

### Минимальные аппаратные требования

- CPU: 1.8 ГГц или быстрее. Два ядра или больше.
- RAM: 8 Гб или больше.
- HDD: 10 Гб или больше свободного места.
- Видеоподсистема поддерживает разрешение дисплея WXGA (1366x768) или выше

### Требования к системному программному обеспечению:

Microsoft Windows 7 Professional или Enterprise (SP1) с обновлением ESU, 8, 10, Server 2008 r2 (SP1) с обновлением ESU, 2012, 2016.

### Требования к прикладному программному обеспечению:

- .NET Framework 4.8
- .NET Desktop Runtime 5.0.x
- Microsoft Visual C++ Redistributable Package 2017

При установке на Windows Server 2012R, для работы функций зрения на компьютерах с операционной системой необходимо установить компонент операционной системы (Feature). Для этого необходимо в PowerShell выполнить команду:

*Install-WindowsFeature server-media-foundation*

---

**Важно:** Для обеспечения корректного запуска и выполнения программного робота на АРМе необходимо убедиться, что всё используемое роботом специализированное ПО заранее установлено, например: браузеры, офисные пакеты, корпоративные системы и прочее.

---

## 4.3 Установка и конфигурирование программы ROBIN Robot 2.0

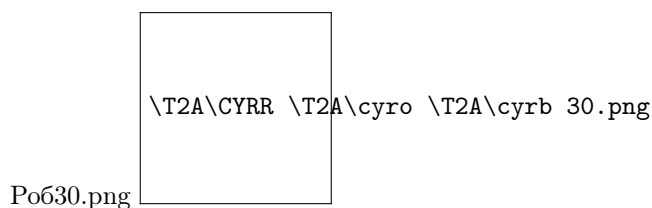
В данном разделе описана установка и настройка Программы для трех редакций – Desktop, Cloud, Enterprise.

Установка программного обеспечения производится с помощью мастера установки. Во время установки параметрам конфигурации Программы (адреса, порты, логины, пароли, протоколы и т.д.) будут присвоены значения по умолчанию.

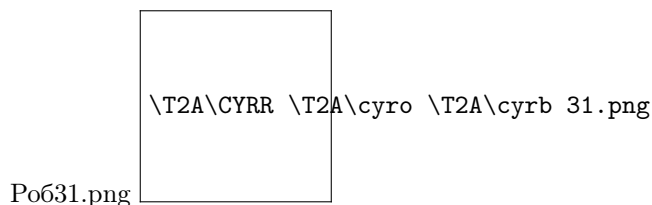
### 4.3.1 Установка программы ROBIN Robot 2.0 для редакции Desktop

Для установки ПО необходимо скачать по ссылке, предоставленной разработчиками или загрузить на АРМ файл с дистрибутивом ПО, имеющим название Robin Platform-std\_ordinal\_mix-2.0.0-RTM-Standalone и выполнить следующую последовательность действий:

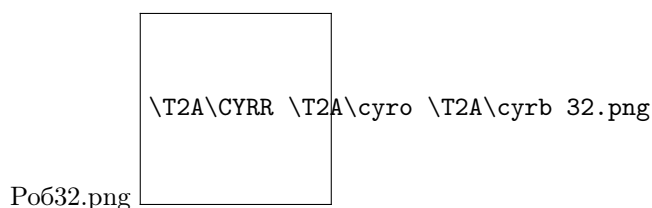
1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла.
2. Выбрать режим установки Программы - для одного или нескольких пользователей АРМа.



3. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].



4. Выбрать компоненты для установки (рекомендуется выбрать все компоненты) и нажать на кнопку [Далее].



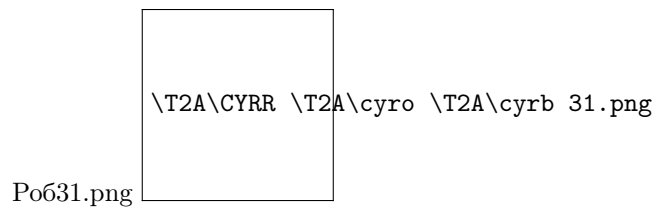
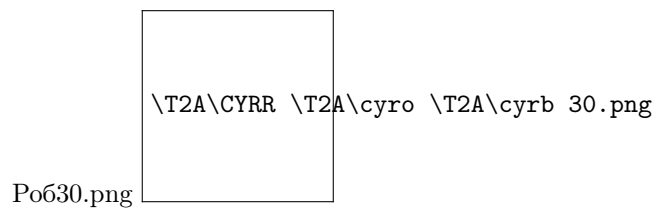
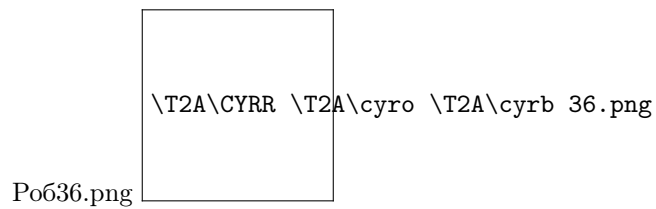
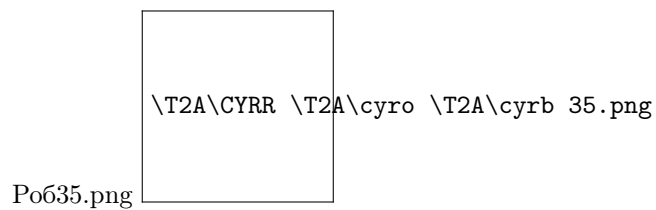
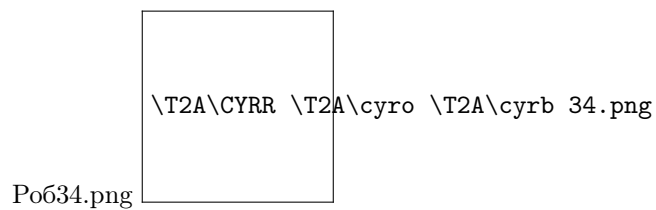
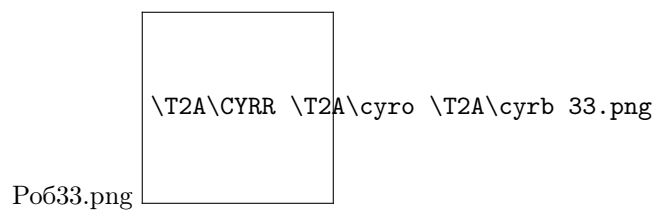
5. Подтвердить начало установки, нажав на кнопку [Установить], программа будет установлена в папку по адресу C:/Program Files/Robin Platform.
6. Далее начнется процедура установки.
7. Для завершения установки необходимо нажать на кнопку [Завершить].
8. Для установки файла лицензии, необходимо скопировать файл вида license\*.lic в папку C:/Program Files/Robin Platform/2.0.0/RobotPlayer.

### 4.3.2 Установка программы ROBIN Robot 2.0 для редакции Cloud

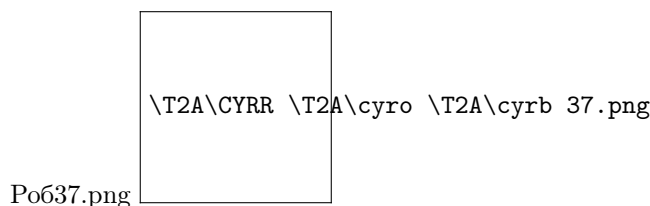
Для скачивания ПО необходимо зарегистрироваться на портале «Облачная фабрика роботов (ОФР)», адрес портала зависит от реализации проекта. Затем необходимо перейти в раздел портала «Дистрибутивы» и нажать на кнопку [Скачать] в строке с названием «Robin Robot».

Для установки ПО необходимо выполнить следующую последовательность действий:

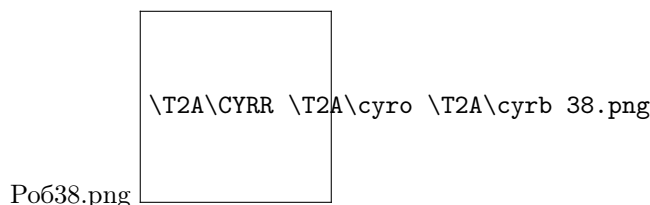
1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла Robin Platform-Robot -2.0.0-RTM-Cloud.exe.
2. Выбрать режим установки Программы - для одного или нескольких пользователей АРМа.
3. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].



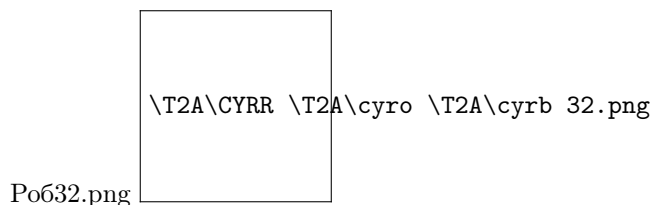
4. Ввести токен, который был сгенерирован для пользователя портала ОФР, и нажать на кнопку [Далее].



Токен пользователя находится в правом верхнем углу под логином пользователя на портале ОФР.



5. Выбрать компоненты для установки (рекомендуется выбрать все компоненты) и нажать на кнопку [Далее].



6. Подтвердить начало установки, нажав на кнопку [Установить], программа будет установлена в папку по адресу C:/Program Files/Robin Platform.
7. Далее начнется процедура установки.
8. Для завершения установки необходимо нажать на кнопку [Завершить].

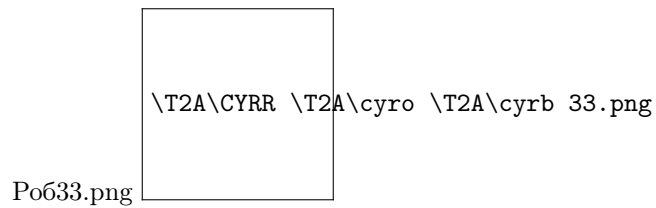
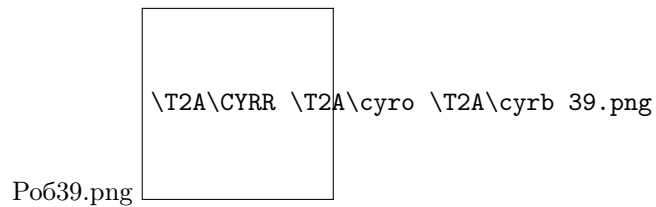
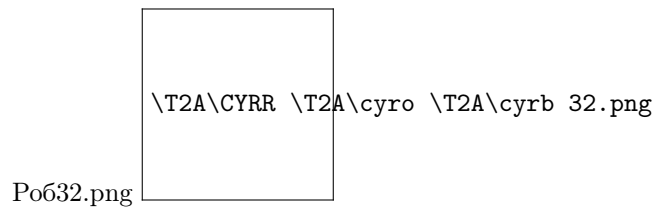
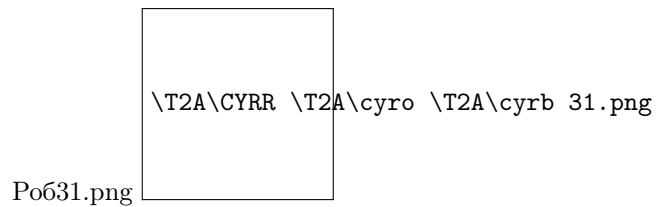
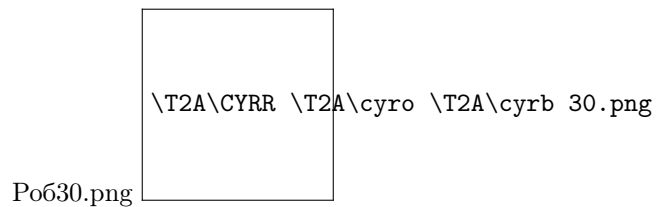
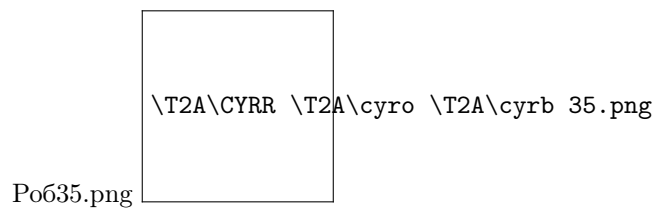
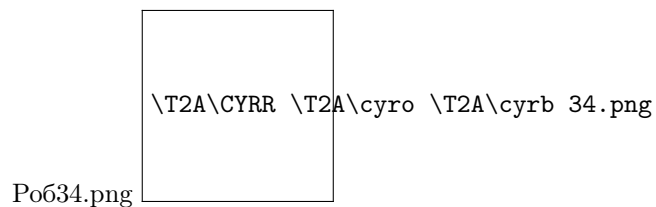
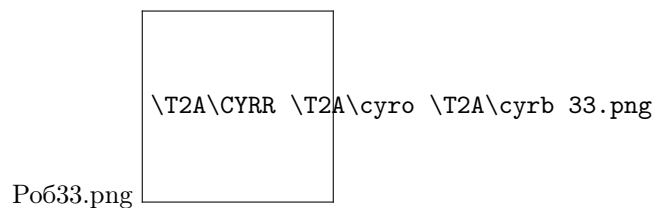
### 4.3.3 Установка программы ROBIN Robot 2.0 для редакции Enterprise

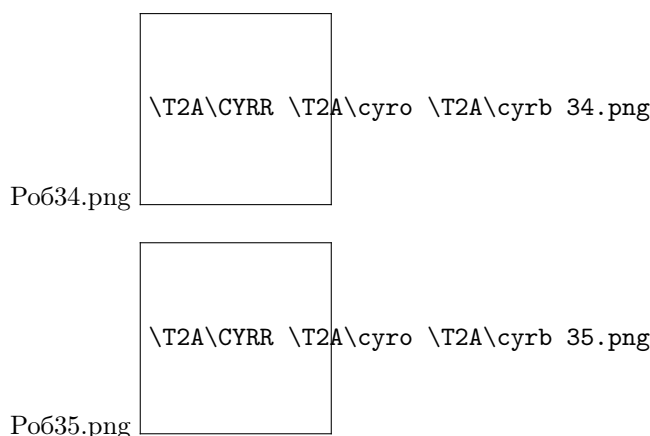
Для установки ПО необходимо скачать по ссылке, предоставленной разработчиками или загрузить на АРМ файл с дистрибутивом ПО, имеющим название Robin Platform-WO\_ENV\_mix-2.0.0-RTM-Enterprise.

Для установки с помощью мастера установки, необходимо выполнить следующую последовательность действий:

1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла.
2. Выбрать режим установки Программы - для одного или нескольких пользователей АРМа.
3. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].
4. Выбрать компоненты для установки (рекомендуется выбрать все компоненты) и нажать на кнопку [Далее].
5. Ввести IP-адрес ROBIN сервера и нажать на кнопку [Далее].
6. Подтвердить начало установки, нажав на кнопку [Установить], программа будет установлена в папку по адресу C:/Program Files/Robin Platform.
7. Далее начнется процедура установки.
8. Для завершения установки необходимо нажать на кнопку [Завершить].







## 4.4 ROBIN Agent

ROBIN Agent модуль отвечающий за:

- получение команды на запуск конкретного робота и сами зашифрованные пакеты этого программного робота. Получает эту команду от ROBIN Orchestrator 2.0, когда речь идет о запуске из оркестратора, и от ROBIN Studio 2.0 при запуске в режиме отладки
- передачу в ROBIN Orchestrator 2.0 логов выполнения робота в режиме реального времени, логов ROBIN Studio 2.0 и Robin Player и своих собственных логов после завершения их работы.

ROBIN Agent устанавливается в операционную систему как служба, поэтому отсутствует полноценный графический интерфейс пользователя.

Изменять настройки модуля можно только через его конфигурационный файл Robin.Agent.Service.exe.config:

- логирование (путь и название файла лога, кодировка, максимальный размер файла лога, количество файлов лога, формат записи (паттерн) строки лога);
- адрес для обращения к хранилищу роботов через сервер приложений;
- настройки подключения к RabbitMQ (имя хоста, порт, логин и пароль пользователя, добавленного в RabbitMQ).

### Взаимодействие ROBIN Agent 2.0 с другими компонентами и модулями

- ROBIN Player. ROBIN Agent отправляет команды на исполнения программных роботов в ROBIN Player, который установлен на том же хосте. Перед началом рабочей сессии приложения проходят процедуру взаимной аутентификации.
- ROBIN Studio 2.0. Если робот запущен в режиме отладки, агент передает логи исполняющегося процесса в ROBIN Studio 2.0 (формирование этого лога пользователь видит сразу на панели лога в Studio). Agent осуществляет взаимодействие только с той Studio, которая установлена на том же хосте. Перед началом рабочей сессии приложения проходят процедуру взаимной аутентификации.

## 4.5 ROBIN Player

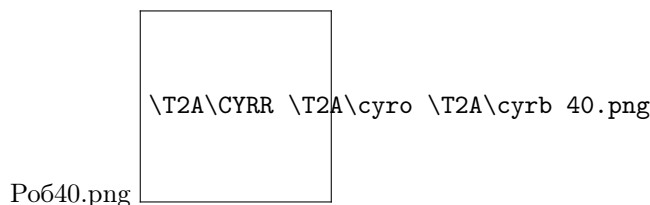
ROBIN Player графическое пользовательское приложение, позволяющее управлять запуском роботов на локальной машине. Данный модуль может запускать сценарий по настроенному расписанию, по команде, полученной от ROBIN Agent или от пользователя (ручной запуск). Роботы выполняются локально, на той же машине и УЗ, на которой установлен модуль.

### 4.5.1 Работа с модулем ROBIN Player

Модуль имеет графический интерфейс для работы пользователей.

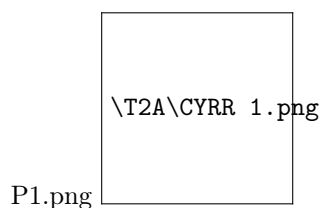
*Порядок действий при входе в приложение ROBIN Player:*

1. Запуск приложения двойным кликом по иконке приложения.

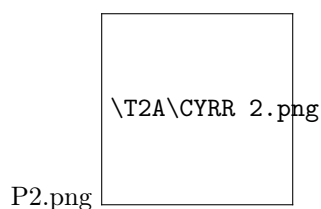


2. Нажатие на иконку приложения в панели задач, после чего будет открыто основное окно с вкладками.
  - Для редакции Desktop авторизация для доступа к ПО не требуется, право доступа предоставляется после установки файла лицензии.
  - Для редакции Cloud авторизация программы осуществляется по ключу с портала ОФР, который вводится пользователем при установке.
  - Для редакции Enterprise необходимо ввести логин и пароль пользователя и нажать на кнопку [Войти].

*Примечание.* Логин и пароль предоставляется пользователю Администратором системы.



После входа в приложение откроется основное окно с вкладками:



#### Взаимодействие ROBIN Player с другими компонентами

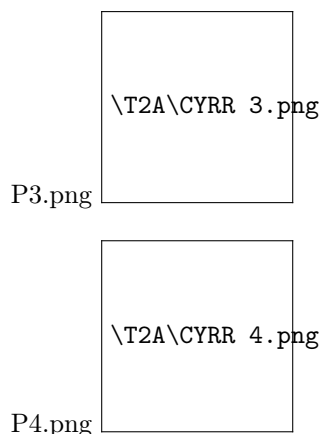
ROBIN Agent. ROBIN Player передает информацию об исполнителе роботов (ROBIN Executor - расшифровывает и исполняет пакеты программного робота) в ROBIN Agent, который установлен на том же хосте. Перед началом рабочей сессии приложения взаимно аутентифицируют друг друга.

#### Вкладка «Запущенные роботы»

На вкладке отображается список запущенных на текущий момент роботов под УЗ пользователя. В списке отображаются робот удаленно через Robin Orchestrator, либо локально самим пользователем или автоматически по расписанию.

Для завершения работы робота необходимо нажать на кнопку «стоп»:

Для приостановки работы робота (поставить выполнение сценария на паузу) необходимо нажать кнопку «пауза»:



### Вкладка «Каталог роботов»

На вкладке «Каталог роботов» имеется возможность запуска роботов доступных пользователю из хранилища.

На вкладке находятся кнопки:

- поиск роботов по названию
- добавление роботов в закладки (избранное)
- добавление новых роботов в каталог из хранилища

### Вкладка «Расписание запуска»

На вкладке пользователь может просмотреть настройки локального расписания роботов.

Для добавления нового расписания для запуска необходимо нажать на кнопку:

Для отключения расписания необходимо передвинуть бегунок:

Для редактирования расписания необходимо нажать кнопку:

Для удаления расписания необходимо нажать кнопку:

Для обновления сделанных изменений необходимо нажать на кнопку:

Для запуска планировщика задач Windows необходимо нажать на кнопку:

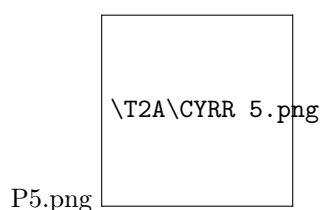
При редактировании расписания открывается окно:

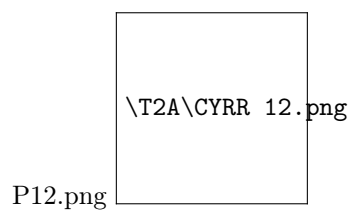
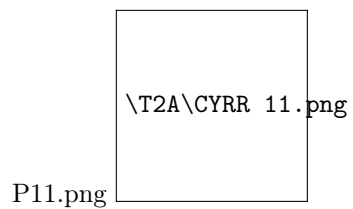
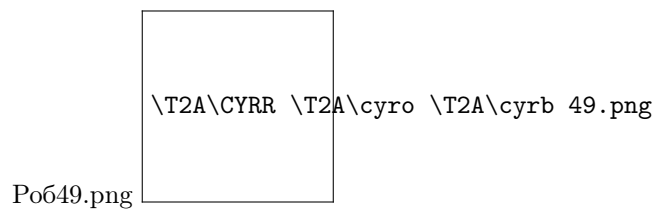
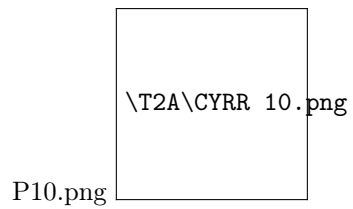
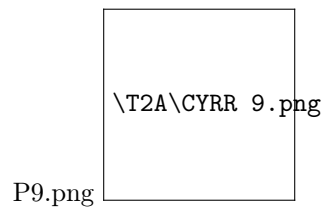
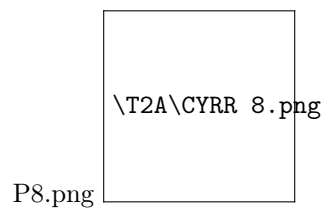
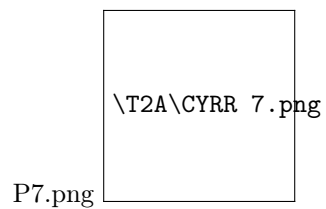
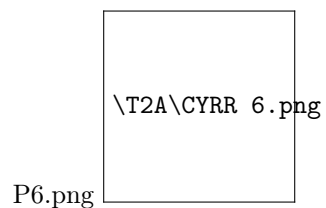
В окне пользователь может редактировать расписание запуска робота, перезапуска, периодичности запуска:

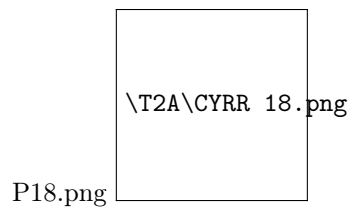
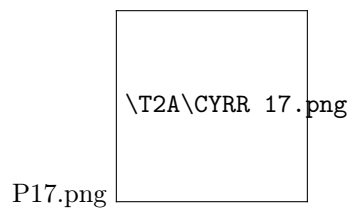
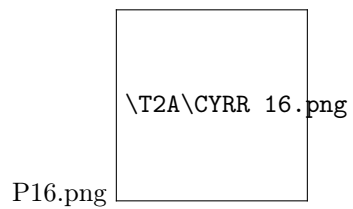
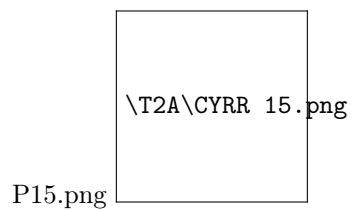
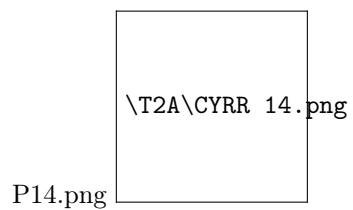
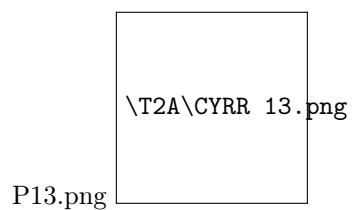
Для активации раздела необходимо передвинуть бегунок в состояние «Включено» и настроить необходимые параметры, затем нажать на кнопку «Обновить».

Ниже приведено описание доступных опций на вкладке:

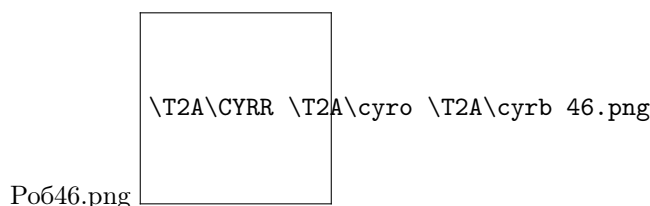
1. Опция «Запуск» предназначена для настройки даты и времени запуска робота. Для настройки опции «Запуск» (дата и время запуска), необходимо нажать на кнопку , а затем с помощью «календаря» выбрать дату и время запуска робота.



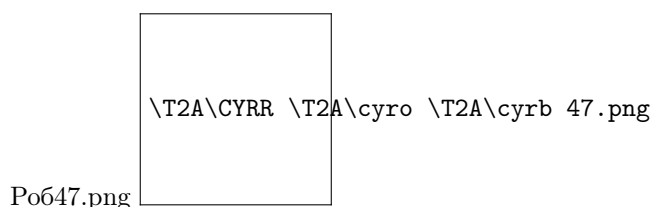




2. Опция «Перезапуск» предназначена для настройки интервала и количество повторных перезапусков робота. Для настройки опции «Перезапуск» необходимо перевести переключатель в режим «Включено», задать интервал повторения, при необходимости перевести переключатель «Повторять в течение» в режим «Включено» и задать количество повторений в интервал времени.

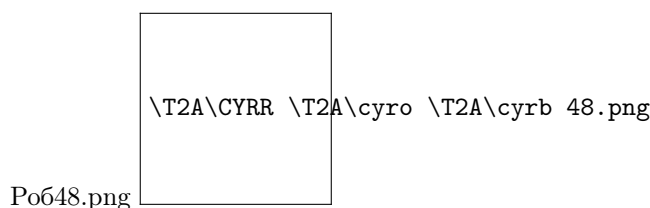


3. Опция «Периодичность» предназначена для настройки периодичности (еженедельной, ежемесячной, ежедневной) запуска робота. Для настройки опции «Периодичность» необходимо перевести переключатель в режим «Включено», а затем выбрать периодичность запуска – ежедневно, еженедельно (далее необходимо выбрать день недели) или ежемесячно (далее необходимо выбрать день месяца).



4. Опция «Запускать до» предназначена для настройки даты и времени, до которой необходимо запускать робота. Для настройки опции «Запускать до» необходимо перевести переключатель в режим «Включено», а затем нажать на кнопку календаря и задать дату и время, до которой необходимо запускать робота.
5. Опция «Параметры запуска» предназначена для загрузки параметров запуска. Для настройки опции «Параметры запуска», необходимо нажать на кнопку в виде папки, а затем выбрать XML-файл с параметрами запуска в папке на АРМе.

Для удаления файла с параметрами запуска из программы, необходимо совершить ЛКМ по строке с названием файла, а затем нажать на соответствующую кнопку.



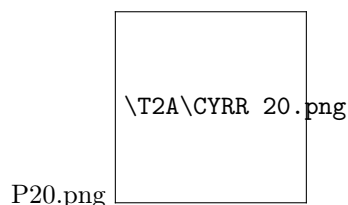
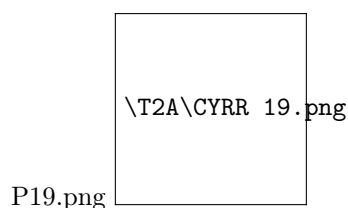
*Примечание.* XML-файл с параметрами запуска будет удален из расписания программы, но останется в папке на АРМе. Для обновления расписания запуска робота, необходимо нажать на кнопку [Обновить], для отмены созданного расписания – на кнопку [Отмена].

## Вкладка «Настройки»

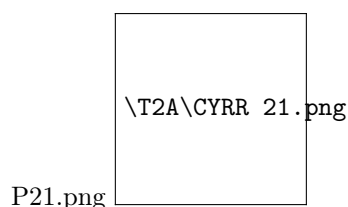
Для перехода на вкладку «Настройки» следует нажать на кнопку в виде шестерёнки.

На данной вкладке имеется доступный функционал по настройке отправки писем при завершении выполнения процесса.

Для настройки отправки писем при завершении выполнения сценария роботом необходимо заполнить поля в разделе «Настройки соединения с почтовым сервером»: Сервер, Порт, Логин, Пароль. Если требуется использовать SSL/TLS необходимо перевести бегунок во включенное состояние. В



разделе «Список получателей уведомлений о выполнении роботов» есть возможность добавления почтовых адресов при успешных выполнениях и неуспешных. Для добавления адресов необходимо нажать на кнопку:



Для прикрепления лога к письму необходимо передвинуть бегунок во включенное состояние. Чтобы сохранить изменения необходимо нажать на кнопку «Сохранить».

Логирование работы роботов осуществляется в автоматическом режиме. Файл с логом размещается в каталоге C:/Program Files/Robin Platform/2.0.0/RobotLogs в папке с названием робота.

В имени файла с логами работы содержится название робота и дата запуска в формате гггг.мм.дд., для каждого запуска робота создается отдельный файл с логом. Для открытия и просмотра файла с логами можно воспользоваться любым приложением, работающим с txt-файлами.

Для смены пароля пользователя необходимо перейти в подраздел «Смена пароля» раздела «Настройки».

Для смены пароля необходимо ввести логин пользователя, а затем текущий пароль, новый пароль и повтор нового пароля и нажать на кнопку [Заменить пароль].

Настройки модуля ROBIN Player можно изменить через его конфигурационный файл RobotPlayer.exe.config:

- логирование (путь и название файла лога, кодировка, максимальный размер файла лога, количество файлов лога, формат записи (паттерн) строки лога);
- настройка размера окна приложения;
- настройки подключения к FreeIPA (включение/отключение аутентификации в Студии, хост и порт подключения, путь к дереву объектов).

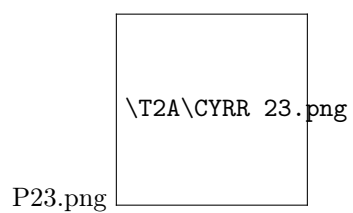
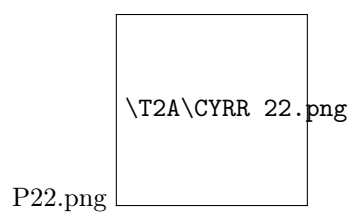
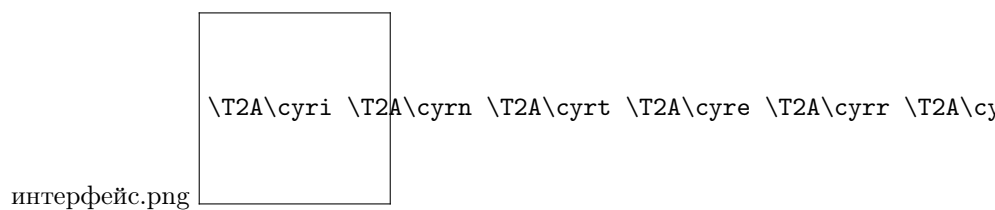
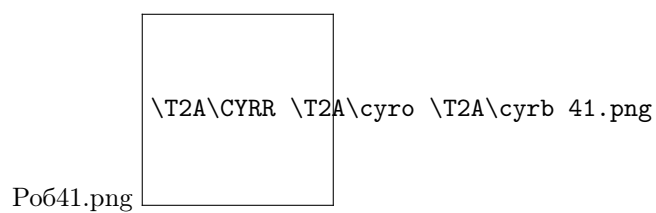
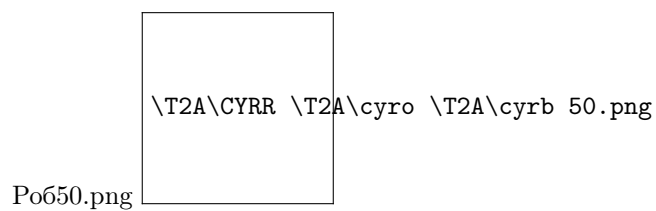
## Вкладка «История запусков»

Интерфейс Robin Player на вкладке «История запусков»:

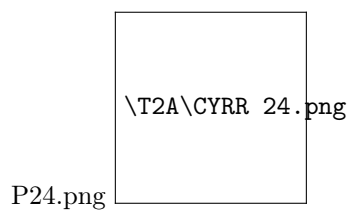
На вкладке «История запусков» доступен следующий функционал:

1. Фильтрация по дате запуска. Если фильтр не установлен, то выводится вся история запусков.
2. Поиск по наименованию запущенного ранее робота. Если фильтрация не задана, выводится вся история запусков.



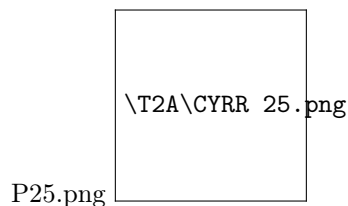


3. Кнопка сброса всех установленных фильтров.

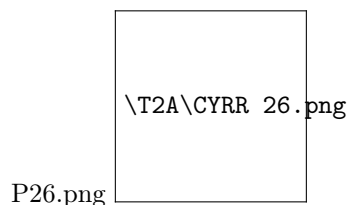


4. Поле с информацией о запущенном роботе, которое включает в себя следующее:

- Поле «Дата и время» - отражается дата и время каждого действия в схеме робота.



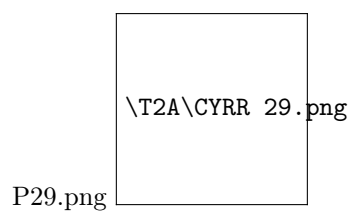
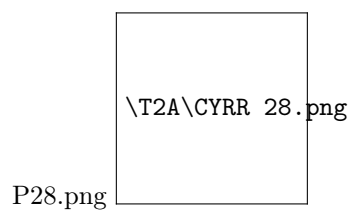
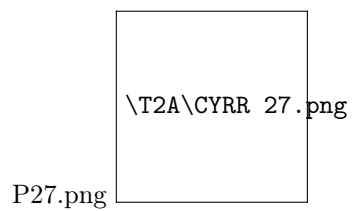
- Поле «Сообщение» - отражает наименование действия, начало действия, конец действия, либо ошибку в действии.



- Поле «Параметр» - отражает параметры, имеющиеся в действии на входе, а также результаты, полученные на выходе.
- Поле «Значение» - отражает значения, внесенные в параметры, и значения, полученные в результатах.

Отражение работы робота на вкладке «История запусков»

Если робот запущен в данный момент, то в строке отсутствует символ.



---

## ROBIN Management Console. Руководство пользователя

---

### 5.1 Аннотация

Настоящий документ «Программное обеспечение Robin Management Console. Руководство пользователя» входит в состав эксплуатационной документации платформы «ROBIN» и содержит описание назначения, условий выполнения, запуска и выполнения программы. Программное обеспечение ROBIN Robot разработано в двух редакциях –Cloud и Enterprise.

В случае, если работа с программой в одной из редакций имеет свои особенности, данная информация указывается в названии раздела документа.

При изложении информации в документе следующие соглашения:

- [Кнопка] – обозначение кнопки в интерфейсе Программы;
- «Поле» - обозначение поля в интерфейсе Программы;
- «пример заполнения» - пример заполнения поля при настройке расписания;
- поле, обозначенное звездочкой - обязательное для заполнения;
- [1] – ссылка на документ.

### 5.2 Назначение программы

«ROBIN Management Console» - пользовательское приложение, входящее в состав платформы ROBIN и предназначенное для контроля функционирования программных роботов, созданных с помощью ПО «ROBIN Studio».

#### 5.2.1 Наименование программы

Полное наименование программы – «ROBIN Management Console». Краткое наименование – Программа.

## 5.2.2 Функции программы

Основными функциями программы являются:

- мониторинг работы ПО «ROBIN Agent» на удаленных компьютерах;
- подключение по протоколу RDP к удаленным компьютерам для запуска ПО «ROBIN Agent»;
- мониторинг работы программных роботов;
- просмотр истории выполнения сценариев программных роботов;
- настройка и отображение ошибок работы RMC и серверного ПО;
- скачивание лога ошибок компонентов системы на АРМ пользователя в формате \*.xlsx и \*.csv;
- создание расписания для запуска сценариев роботов;
- отображение списка сценариев роботов.

## 5.2.3 Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю ПО

Перед началом работы с ПО, пользователю рекомендуется внимательно ознакомиться с настоящим Руководством.

## 5.3 Условия выполнения

Сведения о технических и программных средствах. Таблица ниже содержит минимальные требования к АРМу, необходимые для функционирования Программы.

Характеристика	Значение
CPU	2 ядра x 1.8 GHz или лучше
RAM	2 ГБ
HDD/SSD	Не менее 10 ГБ

Для функционирования программного приложения необходим доступ в сеть Интернет с помощью Wi-Fi или Ethernet. Для функционирования Программы также необходимо установить на АРМе следующее программное обеспечение:

- операционная система: одна из - Windows 7, 8,10, Server 2008, Server 2012 x64.

### 5.3.1 Требования к персоналу

Эксплуатация Программы проводится обученным персоналом. Обучение персонала производится до начала ввода Программы в промышленную эксплуатацию в соответствии с программой обучения.

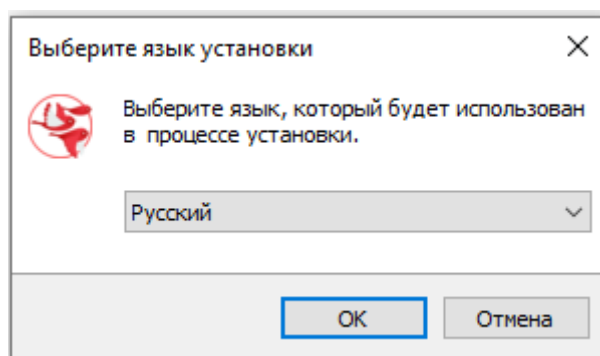
## 5.4 Установка программы

Установка программного обеспечения производится с помощью мастера установки. Во время установки параметрам конфигурации Программы (адреса, порты, логины, пароли, протоколы и т.д.) будут присвоены значения по умолчанию. Изменить значения параметров можно с помощью утилиты RobinLdbConfigUpdater.exe. Дистрибутив для установки программного обеспечения предоставляется разработчиками в виде ссылки для скачивания или файла, который должен быть скопирован на АРМ.

### 5.4.1 Порядок установки для редакции «Cloud»

Для установки ПО необходимо выполнить следующую последовательность действий:

1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла ROBIN MANAGEMENT Console-2.0.0-RTM-Cloud.exe.
2. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].



3. Ввести токен, который был сгенерирован для пользователя портала ОФР, и нажать на кнопку [Далее].

Токен пользователя находится в правом верхнем углу под логином пользователя на портале ОФР.

4. Выбрать папку для установки ПО, нажав на кнопку [Обзор], по умолчанию установка производится в папку C:/Program Files (x86)/ROBIN Management Console 2.3.0 , а затем нажать на кнопку [Далее].
5. Установить «флаг» «Создать значок на Рабочем столе» и нажать на кнопку [Далее].
6. Подтвердить начало установки, нажав на кнопку [Установить].
7. Далее начнется процедура установки.
8. Для завершения установки необходимо нажать на кнопку [Завершить].

### 5.4.2 Порядок установки для редакции «Enterprise»

Для установки ПО необходимо выполнить следующую последовательность действий:

1. Запустить установку, совершив двойное нажатие ЛКМ по названию файла ROBIN MANAGEMENT Console-2.0.0-RTM-Cloud.exe.
2. Разрешить внесение изменений на АРМе, нажав на кнопку [Да], а затем выбрать из выпадающего списка язык интерфейса установки и нажать на кнопку [Ок].
3. Указать адрес Robin Server и нажать на кнопку [Далее].

Установка — ROBIN Management Console 2.3.0-RTM

**Установка токена**  
Укажите актуальный токен

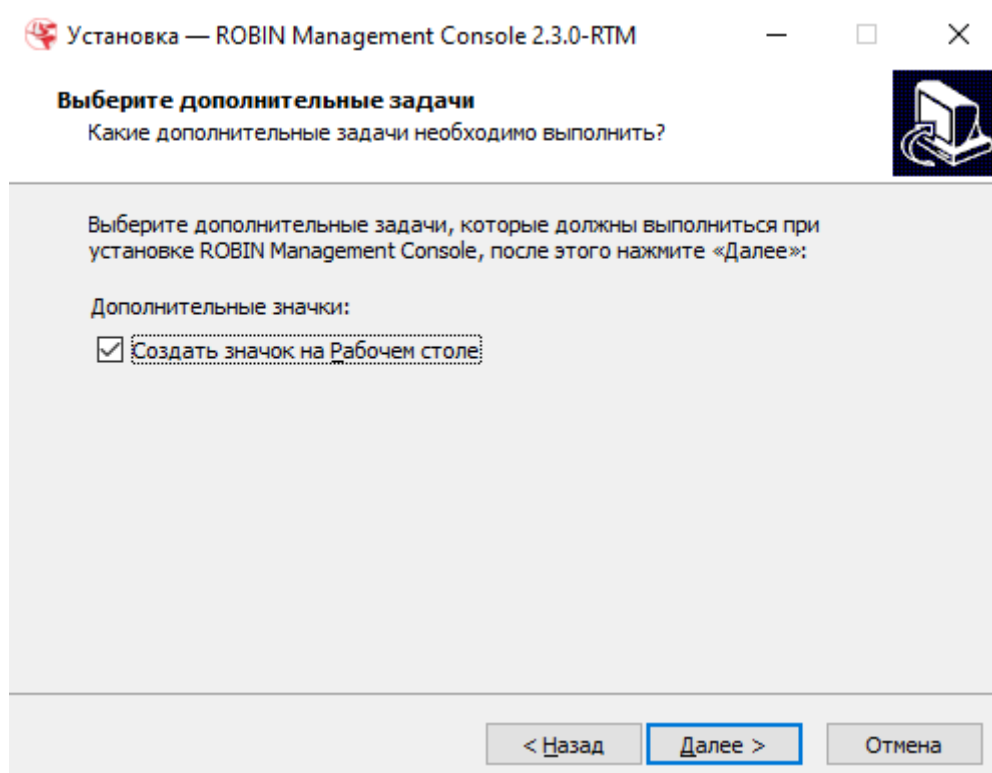
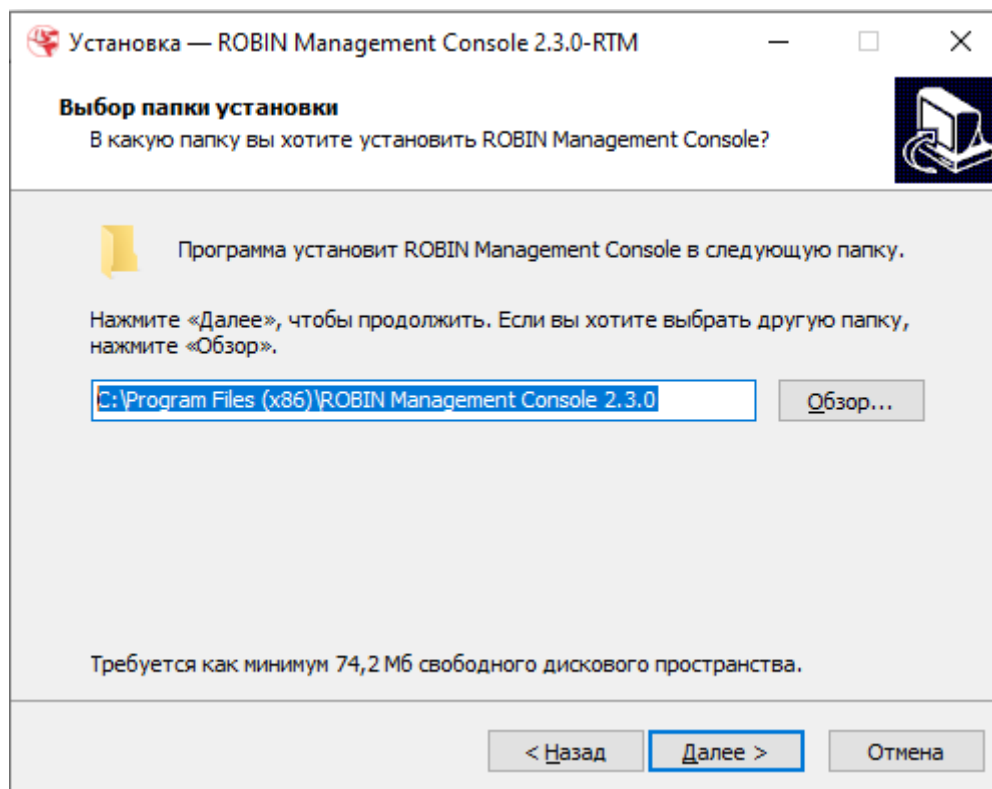
Укажите токен (обязательно):

Неверное указание токена приведет к ошибкам работы

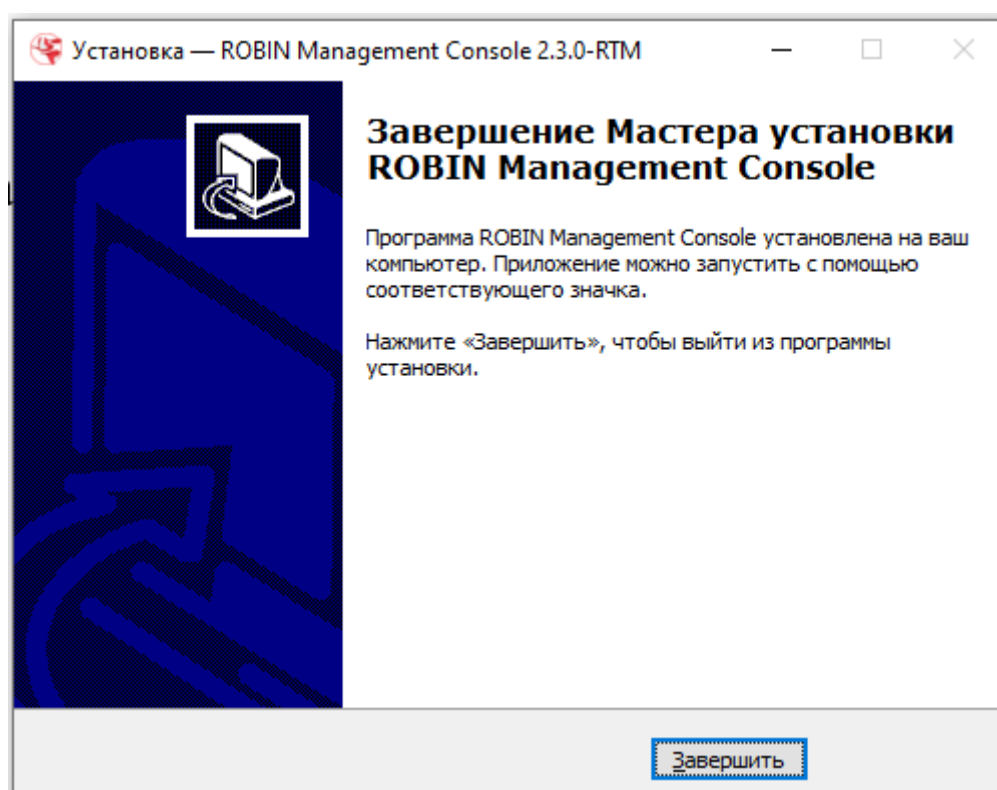
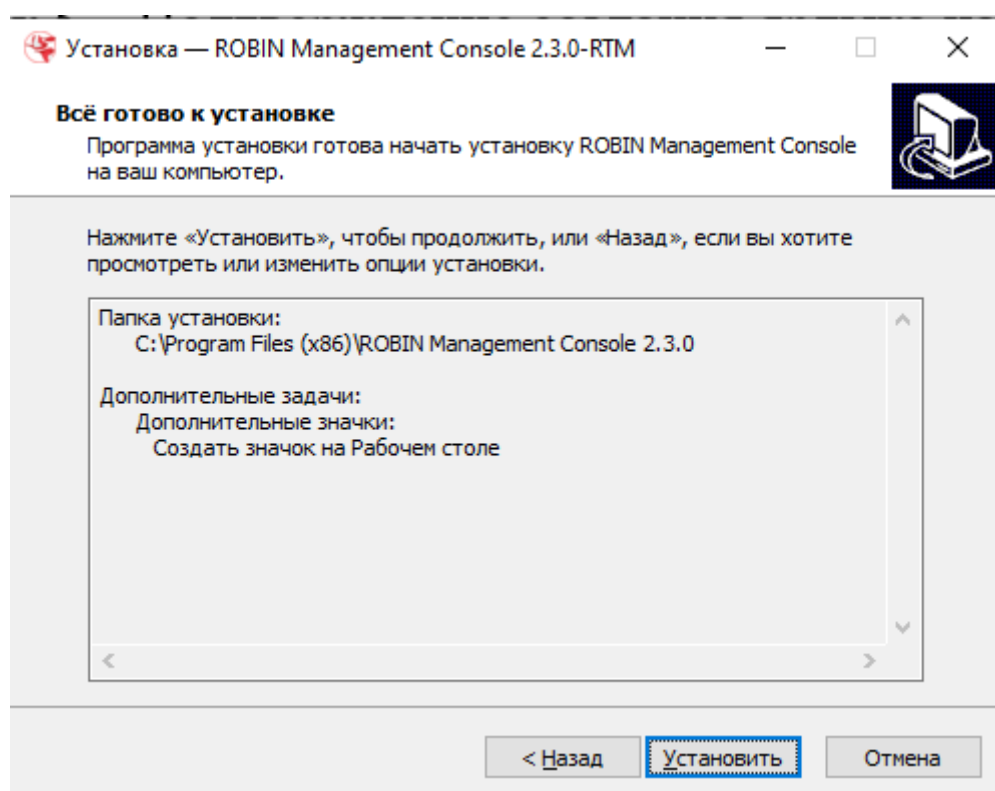
Далее > Отмена

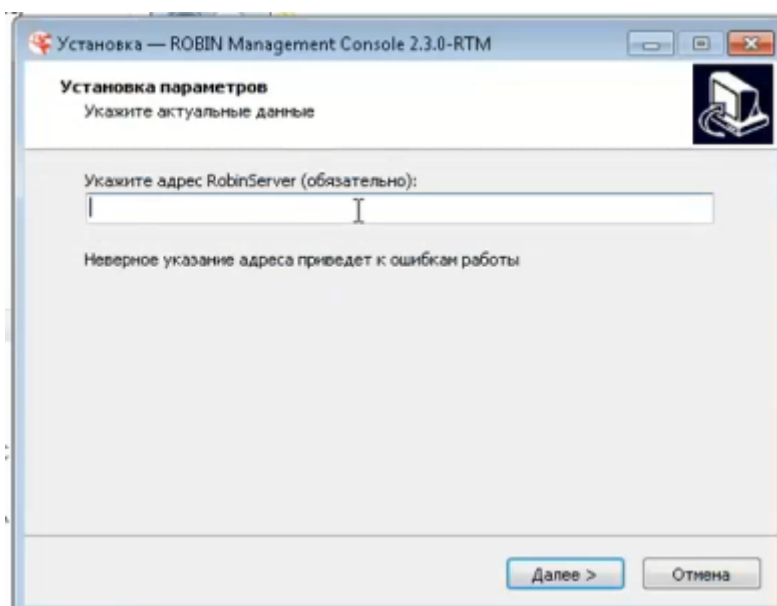
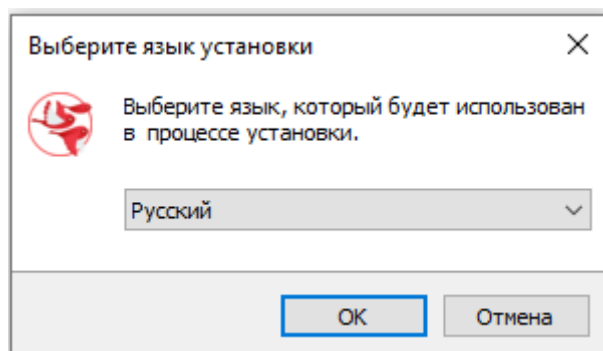
Семен Семенович

Ключ подключения 594f0ae4-e00e-4f2a-b790-809d1ee09184

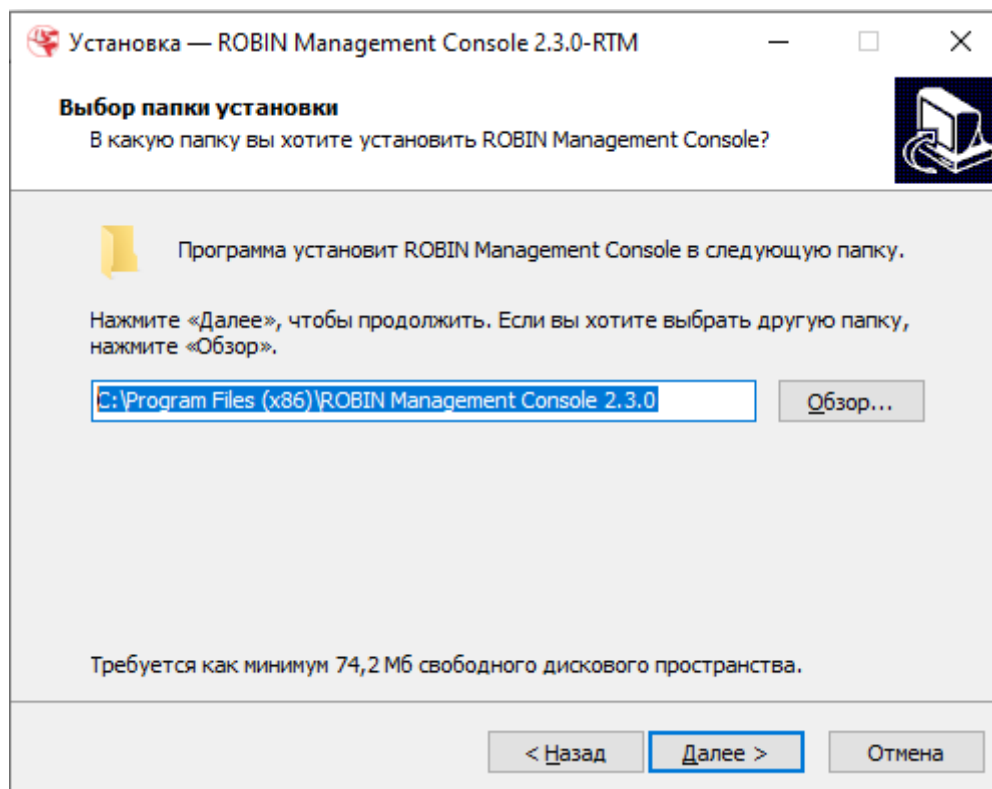








4. Выбрать папку для установки ПО, нажав на кнопку [Обзор], по умолчанию установка производится в папку C:/Program Files (x86)/ROBIN Management Console 2.3.0 , а затем нажать на кнопку [Далее].



5. Установить «флаг» «Создать значок на Рабочем столе» и нажать на кнопку [Далее].
6. Подтвердить начало установки, нажав на кнопку [Установить].
7. Далее начнется процедура установки.
8. Для завершения установки необходимо нажать на кнопку [Завершить].

## 5.5 Конфигурирование программы

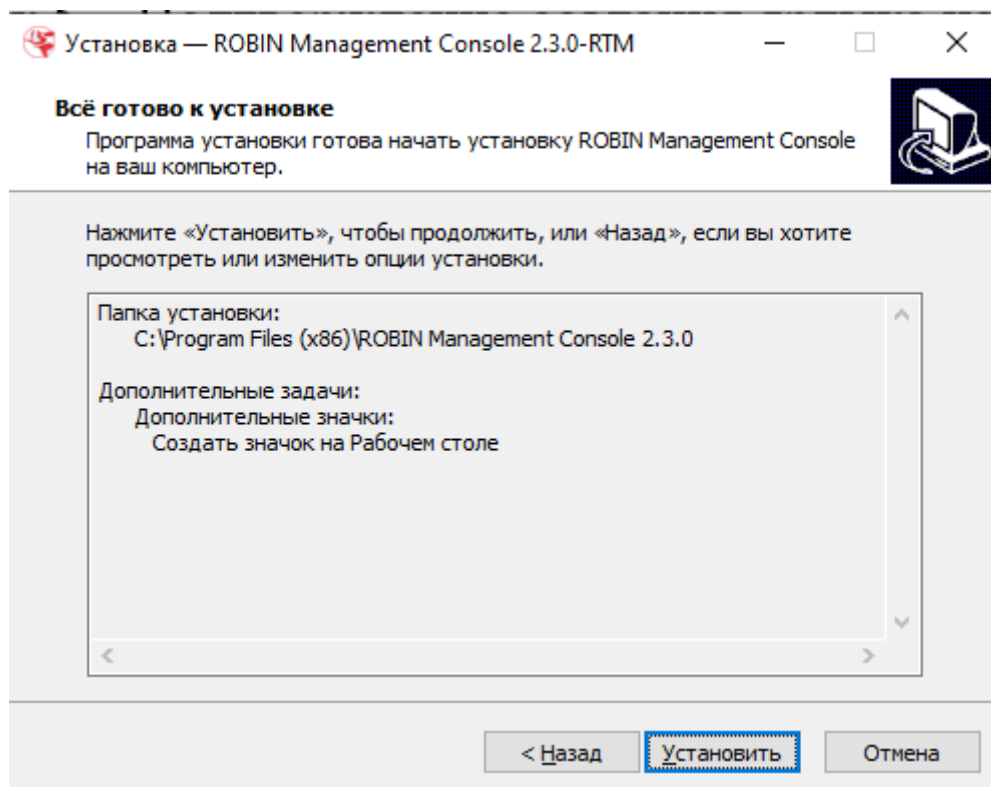
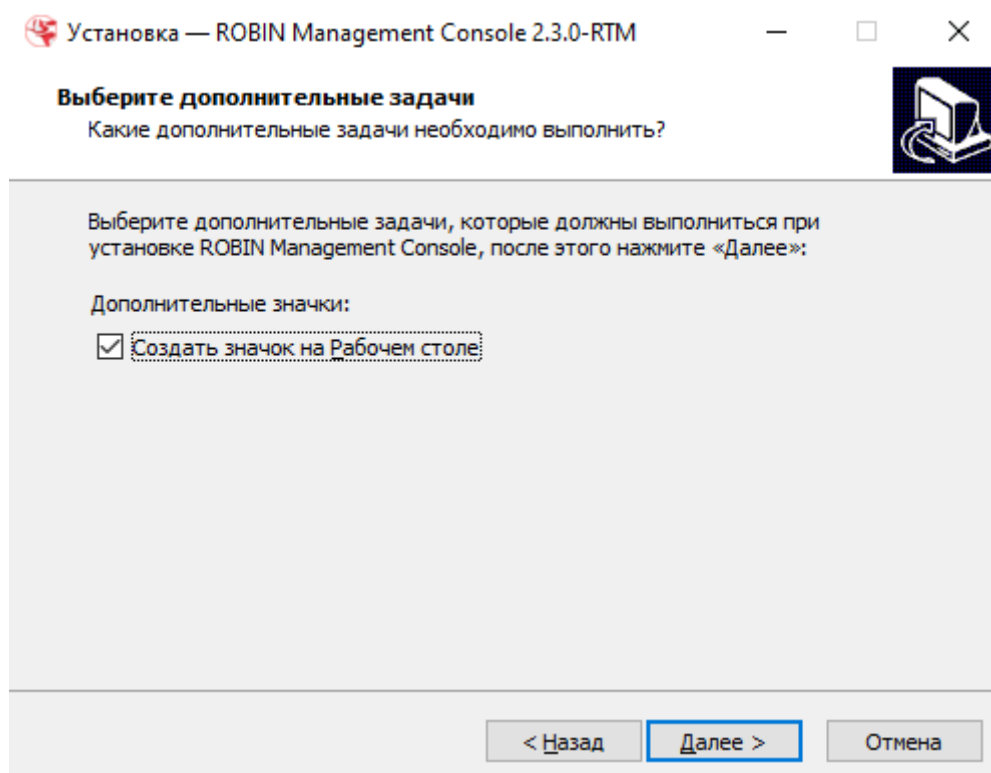
Внимание! Перед настройкой параметров рекомендуется закрыть все приложения Платформы. Для конфигурирования и настройки программы с помощью утилиты RobinLdbConfigUpdater.exe необходимо выполнить следующие действия:

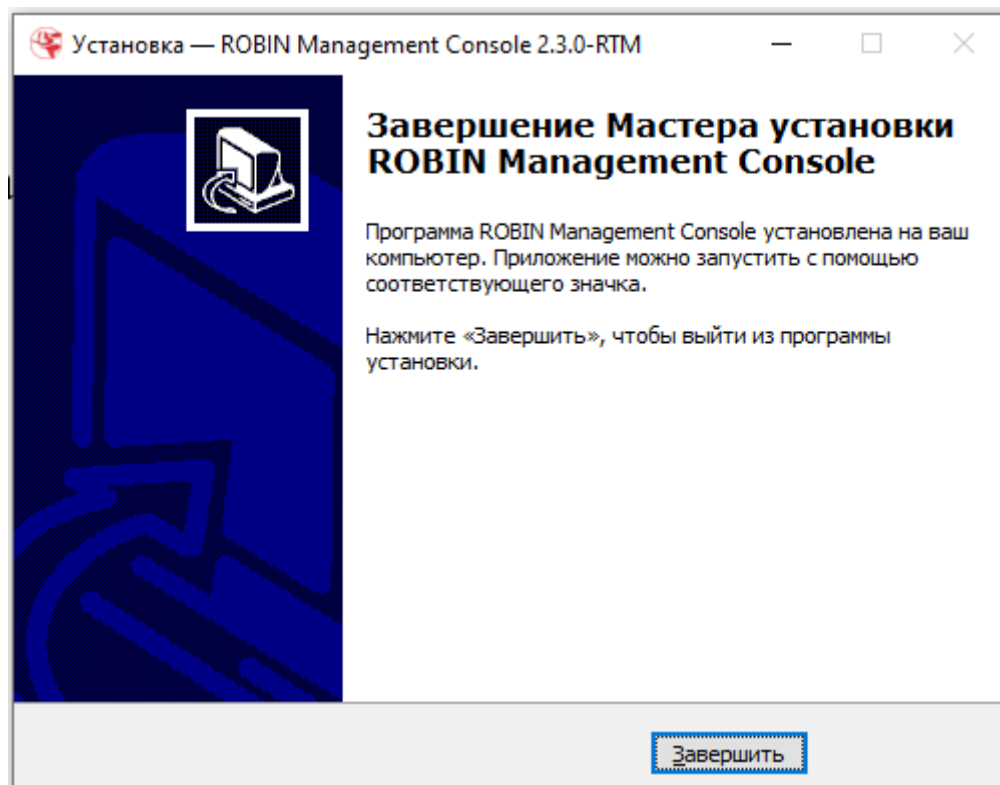
1. Запустить интерпретатор командной строки cmd.
2. Запустить утилиту: Robin Platform/2.0.0/Utils/RobinLdbConfigUpdater.exe

В случае запуска утилиты без параметров, в окне cmd будет указан список параметров для настройки ПО.

3. Для изменения конфигурации ПО, необходимо ввести команду с нужными опциями и задать значения этих опций, например: Robin Platform/2.0.0/Utils/RobinLdbConfigUpdater.exe -RobinServer=>dev-yandex.rpa-robin.ru» -LdapAuthMode =>Kerberos»

Таблица ниже содержит информацию об опциях утилиты конфигурирования ПО.





```
C:\Windows\System32\cmd.exe

-r, --RabbitPort
-l, --RabbitLogin
-u, --RabbitPassword
-d, --LdapBaseServerDn      dn=...,dn=...
-u, --LdapBaseUserDn       cn=...,cn=...
-m, --LdapAuthMode         User:Kerberos
-m, --LdapPort             LDAP port
-q, --CacheRobotDependencies
-t, --ClientToken
--help                     Display this help screen.
--version                  Display version information.

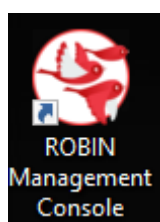
ERROR LIST :
CommandLine.MissingRequiredOptionError
```

Команда	Описание
--BaseDirectory	Базовая директория, относительно которой утилита будет искать конфигурационные файлы. По умолчанию это: "C:\Users\%Username%\AppData\Local\Programs\Robin Platform\2.0.0" и "C:\Program Files (x86)\ROBIN Management Console 2.3.0"
--InstallMode	Cloud
--ClientToken	Клиентский токен (только для облачных установок)
--RobinServer	Адрес балансировщика RMS сервера
--LdapPort	Порт соединения с LDAP сервером
--LdapBaseServerDn	База поиска. Определяет запись в каталоге, из которой выполняется поиск, инициированный клиентами LDAP.
--LdapAuthMode	Настройка авторизации пользователя. Параметр может принимать значение User или Kerberos. Если значение параметра равно User, то для авторизации пользователь должен ввести логин и пароль. Если значение параметра равно Kerberos, то для авторизации для авторизации используются логин и пароль учетной записи пользователя в Windows.
--LogStashHttpUri	URI сервера Logstash

## 5.6 Выполнение программы

### 5.6.1 Запуск программы

Для запуска программы необходимо совершить двойное нажатие ЛКМ по иконке на рабочем столе.




Для редакции Cloud авторизация программы осуществляется по ключу с портала ОФР, который вводится пользователем при установке. Для редакции Enterprise существует два варианта авторизации:

1. Если при настройке ПО для параметра LdapAuthMode было выбрано значение Kerberos, то окно для ввода пароля не отображается.
2. Если при настройке ПО для параметра LdapAuthMode было выбрано значение User, то отображается окно для ввода пароля. Пользователю необходимо ввести логин и пароль пользователя, зарегистрированного в LDAP и имеющего право работать с ПО RMC, и нажать на кнопку [Ok].

Примечание. Смена пароля пользователя для доступа к RMC производится Администратором Сервера управления идентификацией пользователей FreeIPA. Рисунок ниже иллюстрирует пример смены пароля в CentOS Identity Management.

LDAP

Для входа введите логин и пароль




Имя пользователя

Пароль

Ok Отмена

CENTOS IDENTITY MANAGEMENT

 Срок действия вашего пароля истёк. Пожалуйста, введите новый пароль.

Имя пользователя tn-user

Текущий пароль Текущий пароль

Новый пароль \* Новый пароль

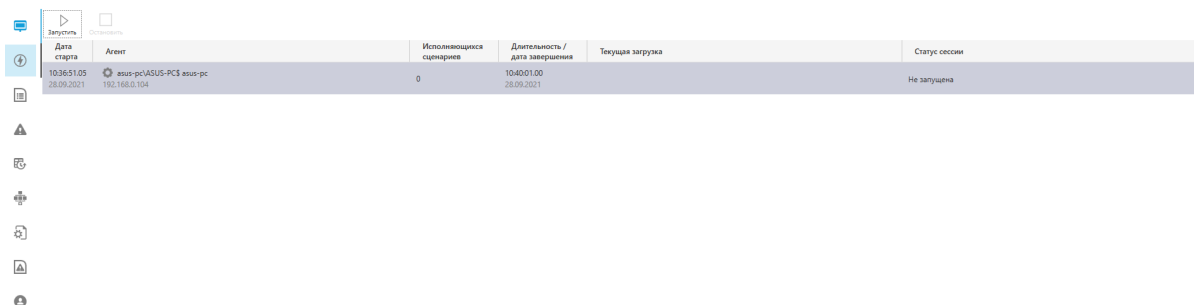
Проверить пароль \* Новый пароль

ОТР One-Time Password

Отменить Сбросить пароль и войти










## 5.6.2 Выполнение программы

Рисунок ниже иллюстрирует главное окно после запуска Программы.



Дата старта	Агент	Исполняющиеся сценарии	Длительность / дата завершения	Текущая загрузка	Статус сессии
10:36:51.03 28.09.2021	192.168.0.104 192.168.0.104	0	10:40:01.00 28.09.2021		Не запущена

В левой части окна расположены кнопки основного меню:

-  - агенты;
-  - исполняющиеся сценарии;
-  - история выполнения сценариев;
-  - ошибки компонентов системы;
-  - расписания;
-  - доступные сценарии;
-  - настройки;
-  - администрирование категорий событий системы;
-  - учетные записи.

В правой части – параметры функционирования роботов или платформы для соответствующего раздела.

### Раздел «Агенты»

Для перехода в раздел «Агенты» необходимо нажать на кнопку, расположенную в левой части интерфейса.



В данном разделе отображается информация о запуске программного приложения ROBIN Agent, установленного на стороне клиентского приложения. ROBIN Agent – прикладной сервис, расположенный на стороне клиента (Robin Studio или Robin Player), который принимает клиентские



запросы со стороны ROBIN Studio, ROBIN Player и ROBIN Management Console и транслирует их системе исполнения сценариев программных роботов Robin Executor.

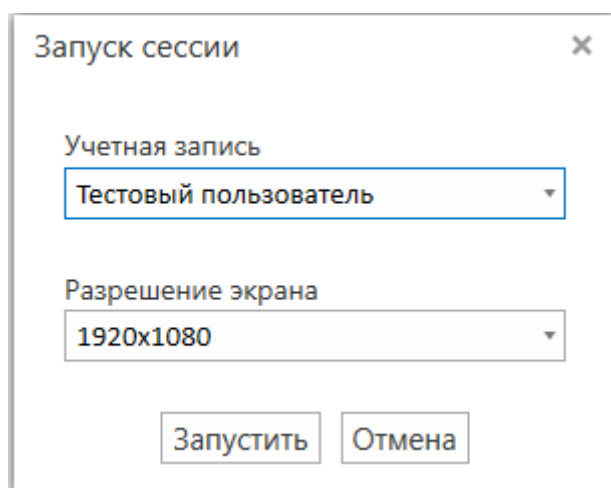
Программное обеспечение Robin Agent запускается при включении компьютера или входе в систему, где установлен ROBIN Agent или с помощью команды из RMC. При запуске агента значок в разделе «Агенты» принимает синий цвет, при остановке серый цвет.

Рисунок ниже иллюстрирует раздел «Агенты».

Дата старта	Агент	Исполняющийся сценарий	Длительность / дата завершения	Текущая загрузка	Статус сессии
13:56:41.08 22.10.2021	USERDAT-IBAS1MA\kkuzmin USERDAT-IBAS1MA 172.16.121.68	0	57m31s	Total CPU : 15.5376996% User CPU : 12.2304784%	Не запущена
14:47:19.03 22.10.2021	GOLOVLEVA-NB GOLOVLEVA-NB\AGolovleva 172.29.254.250	0	14:53:38.34 22.10.2021	Total CPU : 29.5625% User CPU : 13.265625%	Не запущена
21:59:02.62 20.10.2021	ad-ldap-client\vrubel ad-ldap-client 10.172.192.37	0	07:16:53.00 21.10.2021		Не запущена

Для запуска службы с помощью RDP-соединения на соответствующем хосте (запуска Robin Agent с помощью логина и пароля), необходимо нажать ЛКМ по соответствующей строке, а затем нажать на кнопку [Запустить], расположенную в левом верхнем углу интерфейса и ввести имя учетной записи, разрешение экрана для работы приложений с экраном (обязательна установка разрешения экрана при работе с «машинным зрением») и нажать на кнопку [Запустить].

Примечание. Создание учетной записи описано в разделе «Учетные записи» данного документа.



Для выключения RDP-сессии необходимо нажать на кнопку [Остановить], расположенную в левом верхнем углу интерфейса. Таблица ниже содержит информацию о данных, отображаемых в разделе «Агенты».

### Карточка агента (машины)

Для перехода к карточке агента (машины, необходимо совершить нажатие ЛКМ в соответствующей строке. Рисунок ниже иллюстрирует карточку машины. Рисунок отображает вкладку «Выполняется», на которой содержится информация о запущенных сценариях роботов на соответствующей машине.

В верхней части интерфейса содержится информация о машине, дата запуска агента, его версия, количество текущих исполняющихся сценариев, ПО, установленное на машине, аппаратное обеспечение машины и инфраструктурные требования. В нижней части расположены три вкладки – «Выполняется», «История» и «Запланировано».

Рисунок ниже иллюстрирует вкладку «История», на которой содержится информация о выполненных сценариях.

Рисунок ниже иллюстрирует вкладку «Запланировано», на которой находится информация о расписаниях запуска для данной машины.

Параметр	Описание
Дата старта	Дата и время запуска агента
Агент	Название агента
Исполняющихся сценариев	Количество выполняемых сценариев
Длительность/дата завершения	Длительность работы или дата и время отключения агента
Текущая загрузка	Текущая загрузка хоста (отображается загрузка процессора и оперативной памяти)
Статус сессии	Статус RDP-соединения с хостом. Статус может принимать значения: не запущен, запуск, запущен, остановка, неизвестно.

Карточка машины

USERDAT-IBAS1MA\kkuzmin (USERDAT-IBAS1MA)

Регистрация хоста в системе: 27.10.2021 12:48:58

Версия агента: null

Исполняющихся сценариев: 0

Учетная запись агента: USERDAT-IBAS1MA

Total CPU: 5,1168328%

Physical memory usage: 86,1836 mb

User CPU: 4,4148283%

TotalAllocatedMemorySize: 231,7282 mb

SurvivedMemorySize: 3,8907 mb

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

ИНФРАСТРУКТУРНЫЕ ВОЗМОЖНОСТИ

Выполняется История Запланировано

Время запуска	Сценарий	Время исполнения	Учетная запись плеера	Действия	Инициатор запуска
---------------	----------	------------------	-----------------------	----------	-------------------

Статус	Время запуска	Сценарий	Длительность работы	Время завершения	Учетная запись плеера	Инициатор запуска
✓	14:55:22.15 22.10.2021	R 240_rel_test 1.0.0	00:00:12	14:55:34.60 22.10.2021	USERDAT-IBAS1MA	IPA\admin
✓	13:51:41.42 22.10.2021	R 240_rel_test 1.0.0	00:00:05	13:51:46.51 22.10.2021	USERDAT-IBAS1MA	IPA\admin
✓	13:50:00.17 22.10.2021	R 240_rel_test 1.0.0	00:00:06	13:50:06.49 22.10.2021	USERDAT-IBAS1MA	IPA\admin

Название	Ближайшее время запуска	Описание	Сценарий	Всего запусков	Автор	Активен	Часовой пояс
test2	09:56:00.00 28.10.2021	В 09:56	R	23	IPA\admin		
test	00:00:00.00 01.12.2021	В 00:00, только в декабре	R	0	IPA\admin		

## Раздел «Исполняющиеся сценарии»

В данном разделе отображается информация о запущенных на выполнение сценариев роботов на данном «облаке». Для перехода в раздел «Исполняющиеся сценарии» необходимо нажать на кнопку в левой части интерфейса:



Количество запущенных в данный момент сценариев роботов отображается в виде цифры рядом со значком:



Рисунок ниже иллюстрирует раздел «Исполняющиеся сценарии».

Статус	Время запуска	Сценарий	Агент	Время исполнения	Учетная зап.	Журнал исполнения сценария
Выполняется	14:55:22.15 22.10.2021	240_rel_test 1.0.0	USERDAT-IBAS1MA\kkuzmin (USERDAT-IBAS1MA) 172.16.121.68	0m:1s	USERDAT-IBAS	Обновить

Таблица ниже содержит информацию о данных, отображаемых в разделе «Исполняющиеся сценарии».

В правой части интерфейса отображается «Журнал исполнения сценария», (информация о работе робота в части сообщений и параметров). Для просмотра журнала исполнения робота, необходимо нажать на ЛКМ по строке в центральной части интерфейса.

Таблица ниже содержит информацию о данных, отображаемых в разделе «Журнал исполнения сценария».

Для установки выполнения сценария робота необходимо нажать на кнопку «стоп», для паузы в выполнении сценария робота – на кнопку «пауза».

## Раздел «История выполнения сценариев»

В данном разделе отображается информация о выполнении сценариев роботов. Для перехода в раздел «История выполнения сценариев» необходимо нажать на кнопку в левой части интерфейса:

Рисунок ниже иллюстрирует раздел «История выполнения сценариев».

Таблица ниже содержит информацию о данных, отображаемых в разделе «История выполнения сценариев».

В правой части интерфейса отображается «Журнал исполнения сценария» (информация о работе робота в части сообщений и параметров). Для просмотра журнала работы робота, необходимо нажать на ЛКМ по строке в центральной части интерфейса.

Таблица ниже содержит информацию о данных, отображаемых в разделе «Журнал исполнения сценария».

Для обновления данных в Журнале исполнения сценария необходимо нажать на кнопку [Обновить]. Для версии Cloud существует возможность выгрузки данных раздела «История выполнения сценариев» необходимо в верхней части интерфейса с помощью «календаря» задать начальную и конечную дату для построения отчета и нажать на кнопку [Выгрузить отчет].

Далее необходимо указать папку для выгрузки отчета и нажать на кнопку [Сохранить].

Файл будет сохранен в указанную папку АРМа в формате \*.xlsx.

Параметр	Описание
Статус	Статус выполнения
Время запуска	Дата и время запуска робота
Сценарий	Наименование сценария робота
Агент	Хост, на котором запущен сервис Robin Agent
Время исполнения	Время работы робота
Учетная запись плеера	Учетная запись, использованная на хосте для запуска робота
Инициатор запуска	Имя пользователя, который авторизован в RMC и инициировал запуск робота

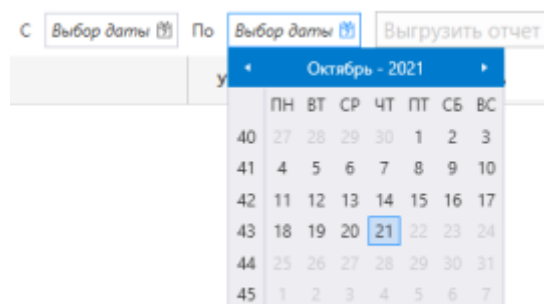
Параметр	Описание
Время	Дата и время события
Сообщение	Сообщение о выполнении действия, которое было внесено в сценарий
Параметр	Список параметров, передаваемых в выполняемое действие
Значение	Значение параметра, передаваемое в выполняемое действие

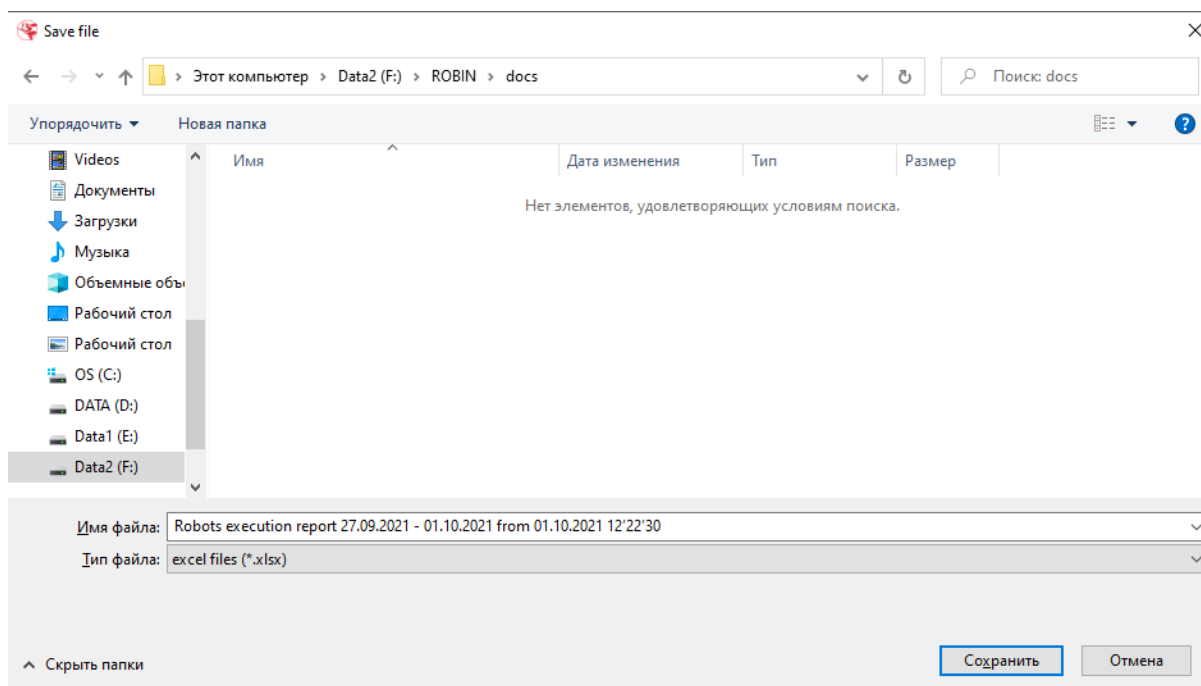


Статус	Время запуска	Сценарий	Длительность работы	Время завершения	Агент	ЖУРНАЛ ИСПОЛНЕНИЯ СЦЕНАРИЯ			
						Время	Сообщение	Параметр	Значение
✓	14:55:22.15 22.10.2021	240_rel_test 1.0.0	00:00:12	14:55:34.60 22.10.2021	(USERDATA-172.16.121.6)	14:55:22.23 22.10.2021	Сообщение 1 Сообщение 1 действие началось	DIALOG_TITLE MESSAGE_TEXT	test test
✓	14:48:57.79 22.10.2021	Golovleva_Test2 2.0.0	00:00:07	14:49:04.97 22.10.2021	(GOLOVLEV172.29.254.2)				
✓	14:42:54.49 22.10.2021	Golovleva_Test 1.0.0	00:00:35	14:43:29.73 22.10.2021	(GOLOVLEV172.29.254.2)				
✓	14:29:27.75 22.10.2021	Golovleva_Test2 1.0.0	00:00:12	14:29:40.39 22.10.2021	(GOLOVLEV172.29.254.2)				

Параметр	Описание
Статус	Статус выполнения сценария робота
Время запуска	Время запуска робота
Сценарий	Название сценария
Длительность работы	Продолжительность выполнения сценария робота
Время завершения	Время завершения работы сценария робота
Агент	Хост, на котором запущен сервис Robin Agent
Учетная запись плеера	Учетная запись, использованная на хосте для запуска робота
Инициатор запуска	Имя пользователя на удаленном компьютере, от имени которого был запущен робот

Параметр	Описание
Время	Дата и время события
Сообщение	Сообщение о выполнении действия, которое было внесено в сценарий
Параметр	Список параметров, передаваемых в выполняемое действие
Значение	Значение параметра, передаваемое в выполняемое действие





## Раздел «Ошибки компонентов системы»

В данном разделе содержится журнал ошибок, которые возникли при выполнении роботов. Для перехода в раздел «Ошибки компонентов системы» необходимо нажать на кнопку в левой части интерфейса:



Рисунок ниже иллюстрирует подраздел «Ошибки компонентов системы».

Тип	Дата	Код ошибки	Компонент системы	Хост	Текст ошибки	Операция
❗	10:45:55.28 21.10.2021	00001	OrchestratorBack		Ошибка обработки данных от агента	AGENT_SYSTEM_USAGE was received - message parsing
❗	10:45:55.28 21.10.2021	00001	OrchestratorBack		Ошибка обработки данных от агента	AGENT_SYSTEM_USAGE was received - message parsing
❗	10:45:15.27 21.10.2021	00001	OrchestratorBack		Ошибка обработки данных от агента	AGENT_SYSTEM_USAGE was received - message parsing
❗	10:45:05.28 21.10.2021	00001	OrchestratorBack		Ошибка обработки данных от агента	AGENT_SYSTEM_USAGE was received - message parsing
❗	10:44:55.27 21.10.2021	00001	OrchestratorBack		Ошибка обработки данных от агента	AGENT_SYSTEM_USAGE was received - message parsing

Таблица ниже содержит сведения об ошибках, отображаемых в разделе «Ошибки компонентов системы».

Примечание. Настройка отображения ошибок описана в разделе «Настройки».

Для перехода на следующую/предыдущую страницу или в конец/начало списка ошибок, необходимо использовать кнопки, расположенные в нижней части интерфейса.

Для загрузки отчета об ошибках необходимо нажать на кнопку [Сохранить в csv], расположенную в левой верхней части интерфейса, а затем указать папку на ARMe, в которую будет выгружен отчет в формате csv.

## Раздел «Доступные расписания»

Раздел «Расписания» предназначен для создания расписаний запуска роботов. Для перехода в раздел «Доступные расписания» необходимо нажать на кнопку в левой части интерфейса:

Рисунок ниже иллюстрирует раздел «Доступные расписания».

Параметр	Описание
Тип	Тип ошибки
Дата	Дата и время возникновения и записи ошибки в журнал.
Код ошибки	Код ошибки
Компонент системы	Компонент системы (источник), на котором произошла ошибка
Хост	Хост, на котором была зафиксирована ошибка
Текст ошибки	Описание ошибки
Операция	Операция сценария, при выполнении которой возникла ошибка



Название	Ближайшее время запуска	Агент	Описание	Сценарий	Всего запусков	Автор	Активен	Часовой пояс
Golovleva	00:00:00.00 01.04.2022	GOLOVLEVA-NB (172.29.254.250)	Каждые 10 минут, с 00:00 по 00:59, только в Апрель	R	0	IPA\training-user		
test2	09:56:00.00 23.10.2021	USERDAT-IBAS1M (172.16.121.68)	В 09:56	R	19	IPA\admin		
testp	15:11:00.00 22.10.2021	dc-win-server\test (10.172.192.157)	В 12:11	R	12	ROBIN\test		
0810	15:15:00.00 08.10.2021	DESKTOP-OLHVC1 (172.19.22.16)	Каждую минуту	R	4	IPA\admin		
testx	15:00:00.00 22.10.2021	dc-win-server\test (10.172.192.157)	Каждые 5 минут, с 12:00 по 12:59	R	120	ROBIN\test		
test	00:00:00.00 01.12.2021	USERDAT-IBAS1M (172.16.121.68)	В 00:00, только в Декабрь	R	0	IPA\admin		

Таблица ниже содержит сведения о расписаниях, отображаемых в разделе «Доступные расписания».

Параметр	Описание
Название	Название расписания
Ближайшее время запуска	Дата и время ближайшего запуска
Агент	Название и IP-адрес агента
Описание	Описание расписания
Сценарий	Название сценария программного робота
Всего запусков	Общее количество запусков
Автор	Создатель сценария
Активен	Переключатель для включения (крайнее правое положение) и отключения (крайнее левое положение) работы расписания запуска сценариев роботов
Часовой пояс	Часовой пояс, по времени которого будет запущен сценарий

Для добавления сценария необходимо нажать на кнопку [Создать] , а затем заполнить поля формы:

- «Название расписания» - название расписания;
- «Часовой пояс» - часовой пояс (указывается в соответствии с локальными настройками даты и времени);
- «Сценарий» - сценарий для запуска, при нажатии на кнопку [Выбор сценария] можно перейти к выбору сценариев роботов (раздел «Доступные сценарии»), для выбора сценария необходимо нажать на ЛКМ в соответствующей строке и нажать на кнопку [Выбрать];
- «Агент» - хост, на котором должен быть запущен сценарий робота (становится доступен только после выбора сценария робота);
- «Запуск» - дата и время запуска сценария, при нажатии на кнопку [Выбрать время запуска] появится доступ к выбору даты и времени запуска сценария робота.

Рисунок ниже иллюстрирует вкладку для выбора месяца для запуска сценария робота.

Для выбора месяца запуска сценария робота необходимо установить «флаг» «В любой месяц» или выбрать «Каждый номер месяца начиная с месяца». Также в разделе «Выбор месяца» пользователь может установить «флаг» рядом с названием месяца, или задать промежуток «Каждый месяц с названием месяца по названием месяца».

Рисунок ниже иллюстрирует вкладку для выбора дня для запуска сценария робота.

Для выбора дня запуска сценария робота необходимо установить «флаг» «В любой день» или установить «флаг» и выбрать периодичность «каждый порядковый день недели начиная с дня недели» или «каждое число месяца начиная с числа» или выбрать определенный день недели для запуска, установив «флаг» рядом с соответствующим названием дня недели или числом месяца. Пользователь также может установить «флаг» рядом с надписью «В последний день месяца», «В последний будний день месяца», «В последний название дня недели месяца», «количество дней до конца месяца».



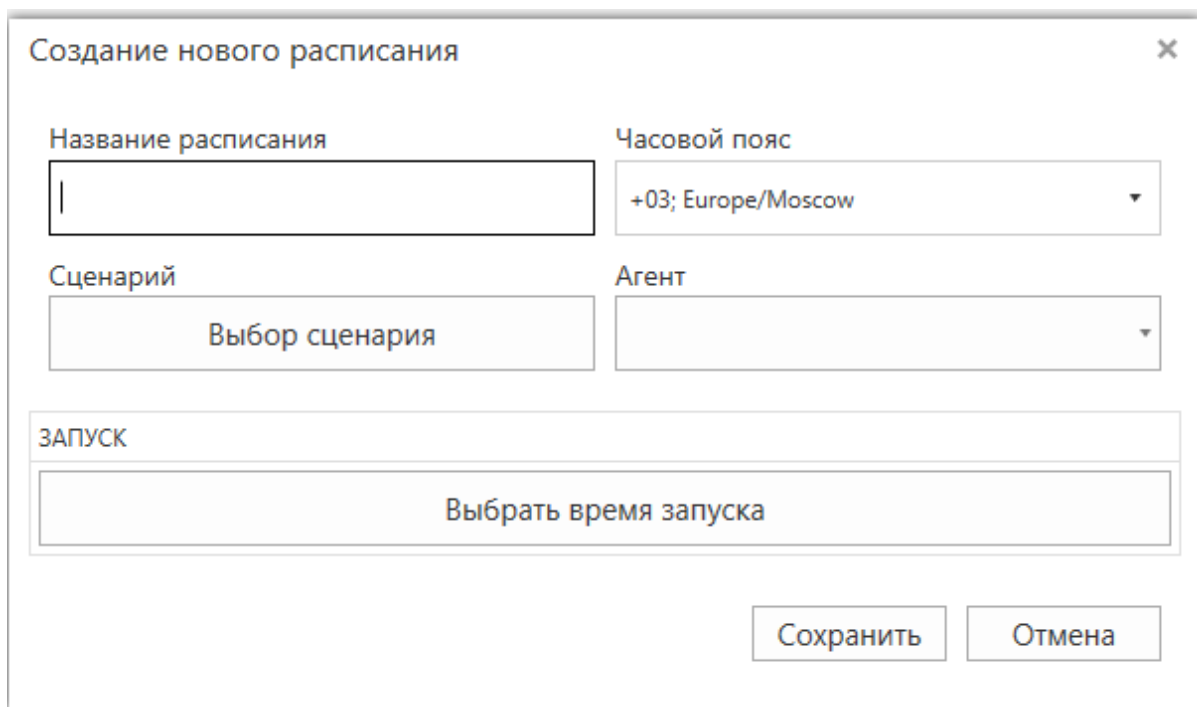


Рисунок ниже иллюстрирует вкладку для выбора часа для запуска сценария робота.

Для выбора часа запуска сценария робота необходимо установить «флаг» «В любой час» или «Каждый количество часов, начиная с часа» или в разделе «Определенные часы» установить «флаг» рядом с часом запуска, а также задать промежуток часов для каждого дня.

Рисунок ниже иллюстрирует вкладку для выбора минуты для запуска сценария робота.

Для выбора минуты запуска сценария робота необходимо установить «флаг» «В любую минуту» или «Каждые количество минут, начиная с минуты» или установить «флаг» рядом с минутой запуска, а также задать промежуток минут для каждого часа. Для сохранения настроек запуска необходимо нажать на кнопку [Выбрать] на любой из вкладок (день, час, месяц).

## Раздел «Доступные сценарии»

В разделе «Доступные сценарии» отображаются сценарии, доступные для запуска. Загрузка сценариев осуществляется через ОФР или с помощью ПО «ROBIN Studio». Для перехода в раздел «Доступные сценарии» необходимо нажать на кнопку в левой части интерфейса:

Рисунок ниже иллюстрирует раздел «Доступные сценарии».

Таблица ниже содержит сведения о сценариях, отображаемых в разделе «Доступные сценарии».

Для поиска робота по названию, необходимо ввести полное или часть название в поле [Сценарий] и нажать Enter. Для запуска робота необходимо нажать на ЛКМ по строке с названием робота и нажать на кнопку [Запуск], а затем выбрать хост, на котором будет запущен сценарий робота), при необходимости задать параметры запуска (выбрать уровень логирования) и нажать на кнопку [Запустить] или [Старт].

## Раздел «Настройки»

В данном разделе пользователь ПО может выбрать язык интерфейса. Выбор языка интерфейса осуществляется с помощью выпадающего меню.

Выбор времени запуска

×

МинутаЧасДеньМесяц

☒ В любой месяц

☐ Каждые  Месяц(ев), начиная с

☐ Определенные Месяцы

☐ Январь☐ Февраль☐ Март☐ Апрель☐ Май☐ Июнь☐ Июль☐ Август☐ Сентябрь☐ Октябрь☐ Ноябрь☐ Декабрь

☐ Каждый месяц с  по

В 00:00

Выбрать

Выбор времени запуска

×

Минута

Час

День

Месяц

☒ В любой день

☐ Каждый 

1

 День(дней), начиная с 

понедельник

☐ Каждый 

1

 День(дней), начиная с 

1го

 числа месяца.

☐ Определенный День Недели

☐ понедельник ☐ вторник ☐ среда ☐ четверг  
☐ пятница ☐ суббота ☐ воскресенье

☐ Определенный День Месяца

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11  
☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20 ☐ 21 ☐ 22  
☐ 23 ☐ 24 ☐ 25 ☐ 26 ☐ 27 ☐ 28 ☐ 29 ☐ 30 ☐ 31

☐ В последний день месяца

☐ В последний будний день месяца

☐ В последнее 

понедельник

 месяца

☐

1

 день(дней) до конца месяца

В 00:00

Выбрать

Выбор времени запуска

×

МинутаЧасДеньМесяц

☐ В любой час

☐ Каждый 

1

 Час(ов), начиная с 

0

☒ Определенные Часы

☒ 0☐ 1☐ 2☐ 3☐ 4☐ 5

☐ 6☐ 7☐ 8☐ 9☐ 10☐ 11

☐ 12☐ 13☐ 14☐ 15☐ 16☐ 17

☐ 18☐ 19☐ 20☐ 21☐ 22☐ 23

☐ Каждый час между 

0

 и 

0

В 00:00

Выбрать

5.6. Выполнение программы

246

Выбор времени запуска

Минута

Час

День

Месяц

☐ В любую минуту

☐ Каждые 

1

 Минут(у), начиная с 

0

☒ Определенные Минуты

☒ 0

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

☐ 6

☐ 7

☐ 8

☐ 9

☐ 10

☐ 11

☐ 12

☐ 13

☐ 14

☐ 15

☐ 16

☐ 17

☐ 18

☐ 19

☐ 20

☐ 21

☐ 22

☐ 23

☐ 24

☐ 25

☐ 26

☐ 27

☐ 28

☐ 29

☐ 30

☐ 31

☐ 32

☐ 33

☐ 34

☐ 35

☐ 36

☐ 37

☐ 38

☐ 39

☐ 40

☐ 41

☐ 42

☐ 43

☐ 44

☐ 45

☐ 46

☐ 47

☐ 48

☐ 49

☐ 50

☐ 51

☐ 52

☐ 53

☐ 54

☐ 55

☐ 56

☐ 57

☐ 58

☐ 59

☐ Каждая минута между 

0

 и 

0

В 00:00

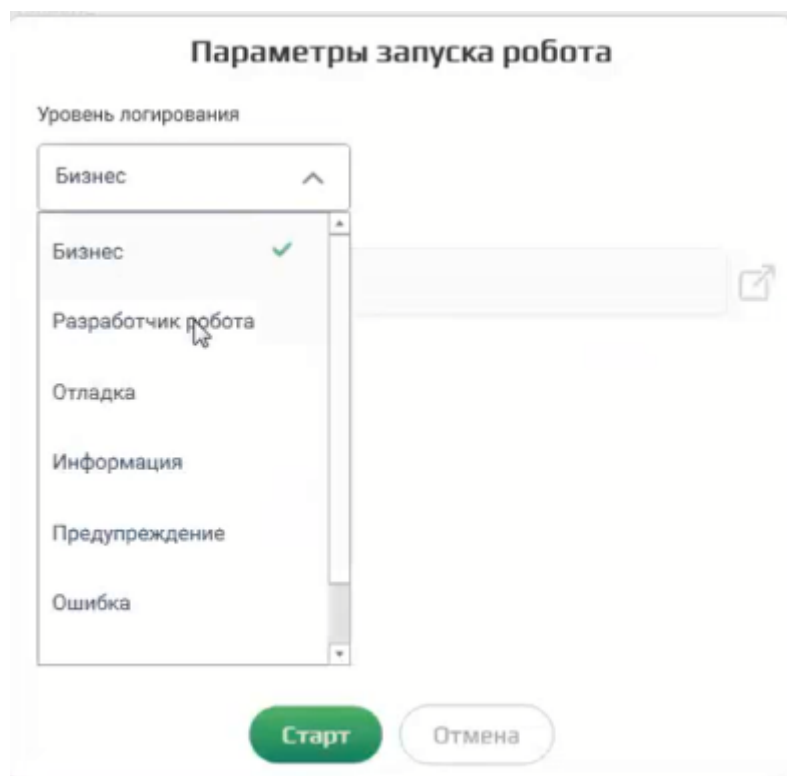
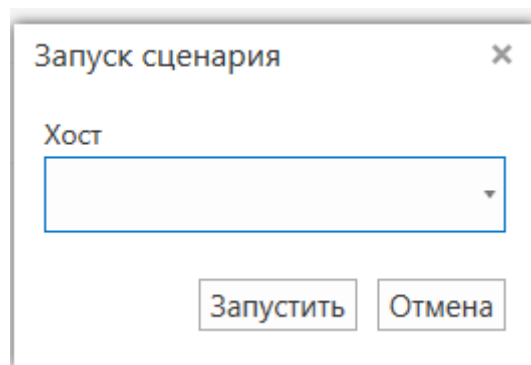
Выбрать

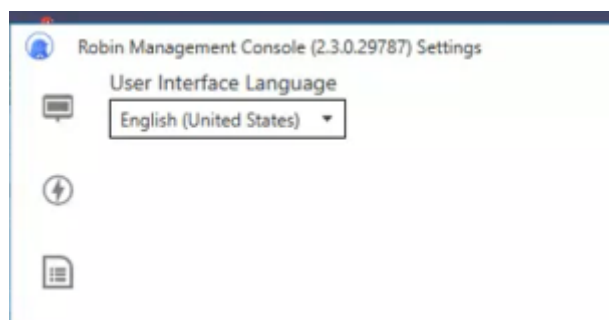
Название	Версия	Описание	Автор	Дата публикации
<a href="#">Интеграция с системой</a>	5.2.3	description(lang-ru=	Robin	9/28/2021 11:20:36 AM
<a href="#">Перенос исторических данных</a>	5.2.3	description(lang-ru=	Robin	9/28/2021 11:17:18 AM
<a href="#">Обработка звонков</a>	5.2.3	description(lang-ru=	Robin	9/28/2021 11:13:57 AM
<a href="#">Обработка входящих сообщений</a>	5.2.3	description(lang-ru=	Robin	9/28/2021 11:09:18 AM
<a href="#">Обработка заявок</a>	5.2.3	description(lang-ru=	Robin	9/28/2021 11:04:54 AM

5.6. Выполнение программы

247

Параметр	Описание
Название	Название сценария робота
Версия	Версия сценария робота
Описание	Описание робота
Автор	Наименование компании, разработавшей робота
Дата публикации	Дата размещения сценария робота на ОФР





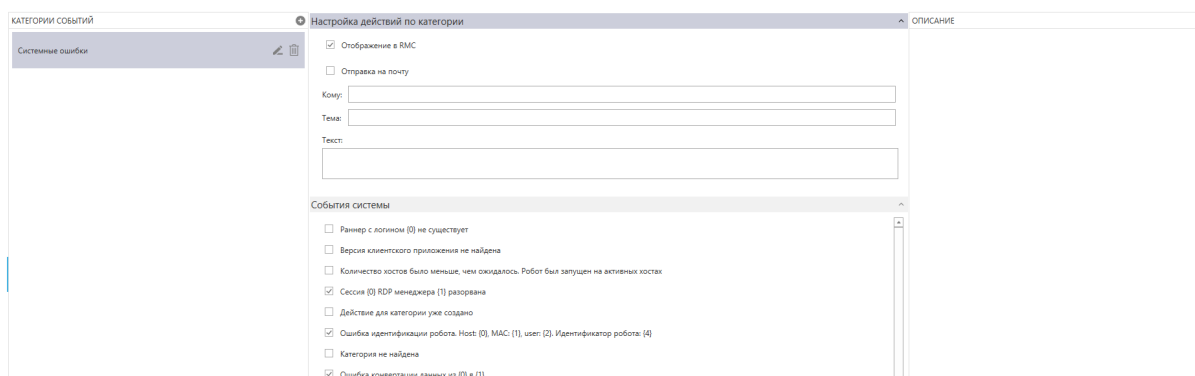
## Раздел «Администрирование категорий событий»

В разделе «Администрирование категорий событий» пользователь RMC может: \* создавать журналы событий; \* выполнить настройку содержания журнала событий, которые произошли при работе роботов и были зарегистрированы ROBIN Agent; \* указывать способ отображения событий для конечного пользователя – в RMC или через отправку сообщений на почту.

Для перехода в раздел «Администрирование категорий событий» необходимо нажать на кнопку в левой части интерфейса:



Рисунок ниже иллюстрирует раздел «Администрирование категорий событий».

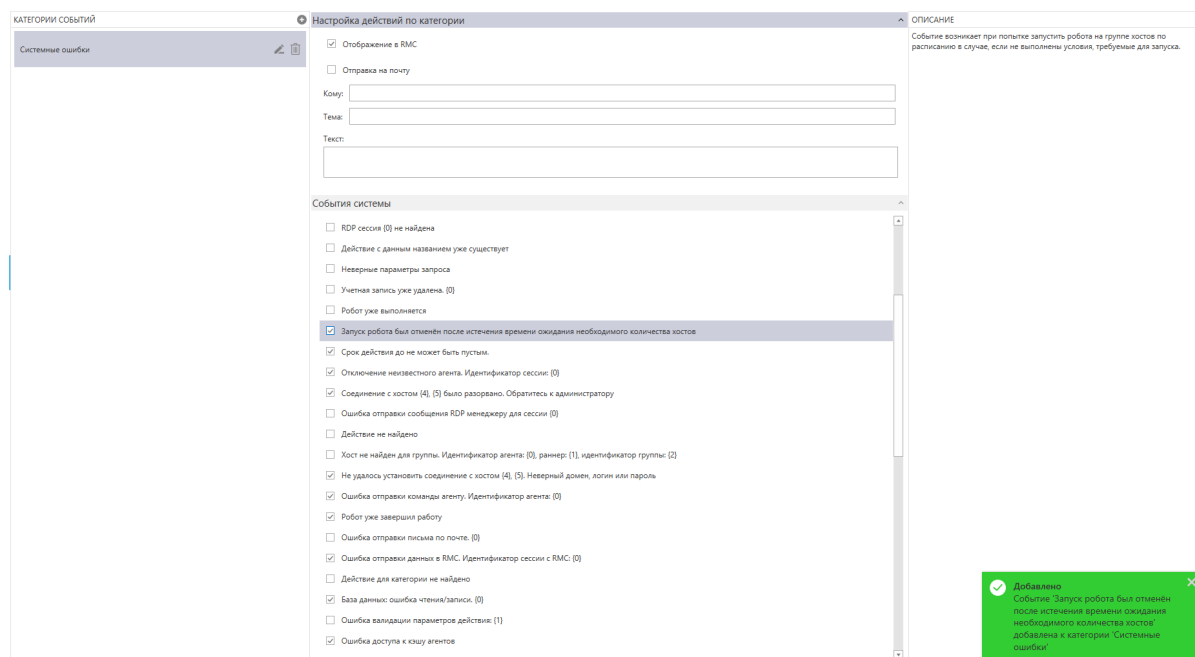


В левой части окна находится раздел «Категории событий», в котором пользователь может создать журнал с набором событий для отображения. Для создания журнала необходимо нажать на кнопку в виде плюса в строке «Категории событий», а затем ввести название журнала.

В средней части интерфейса находится раздел «События системы», в котором содержатся названия событий. В правой части – раздел «Описание», в котором находится описание соответствующего события.

Для добавления вида события в журнал, необходимо нажать на кнопку в виде карандаша, а затем поставить «флаг» рядом с названием события. В случае успешно добавления события в журнал, появится сообщение в правом нижнем углу интерфейса.

В верхней части интерфейса находится раздел «Настройка действий по категории», в котором пользователь может установить «флаги» для отображения выбранных событий в разделе RMC «Ошибки» (раздел «Ошибки компонентов системы») или отправки на электронную почту отчета о событиях. Для отправки на электронную почту отчета о событиях необходимо также ввести e-mail, название темы и текст сообщения в письме.



## Раздел «Учетные записи»

В разделе «Учетные записи» отображается информация о пользователях ПО.

Рисунок ниже иллюстрирует раздел «Учетные записи».

Название	Описание	Статус	Действительна до
Тестовый пользователь	Пользователь создан для тестирования	Недействительна	

Таблица ниже содержит сведения о пользователях, отображаемых в разделе «Учетные записи».

Параметр	Описание
Название	Имя пользователя
Описание	Описание пользователя
Статус	Статус пользователя. Статус может принимать значения «Действительна», «Недействительна», «Истек срок действия»
Действителен до	Срок действия учетной записи

Для добавления новой учетной записи необходимо нажать на кнопку [Создать], а затем заполнить форму:

Таблица ниже содержит информацию о добавлении учетной записи:

Для сохранения данных о пользователе необходимо нажать на кнопку [Сохранить].

Для редактирования пользователя необходимо нажать на ЛКМ по строке с информацией о пользователе, а затем нажать на кнопку [Редактировать], для удаления пользователя необходимо нажать на ЛКМ по строке с информацией о пользователе, а затем нажать на кнопку [Удалить].



Создать учетную запись

Название

Описание

Логин

Статус

Действительна

Пароль

Действительна до

Выбор даты

14

Сохранить

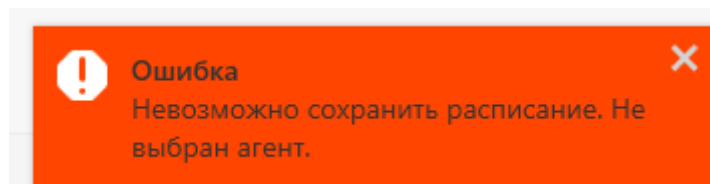
Отмена

Параметр	Описание
Название	Имя пользователя
Описание	Описание учетной записи
Логин	Логин учетной записи
Пароль	Пароль учетной записи
Статус	Статус пользователя. Статус может принимать значения «Действительна», «Недействительна», «Истек срок действия»
Действительна до	Срок действия учетной записи, выбирается с помощью «календаря»

## 5.7 Сообщения оператору

В процессе работы с Программой пользователь может получать в интерфейсе Программы сообщения об ошибках, являющиеся следствием некорректного ввода данных или ввода недопустимых данных. В этом случае пользователю следует ввести корректные данные или обратиться к системному администратору за консультацией.

В случае возникновения ошибки, связанной с работой приложения, сообщение появляется в правом нижнем углу интерфейса.



При возникновении подобной ошибки пользователю необходимо ознакомиться с сообщением, закрыть сообщение и исправить параметры, которые привели к ошибке или обратиться к системному администратору за консультацией.

---

## ROBIN Orchestrator. Руководство системного администратора

---

### 6.1 Аннотация

Данный документ предназначен для системного программиста (администратора) серверных компонент Автоматизированной Системы «ROBIN Orchestrator» версия 2.0 (далее – АС «ROBIN Orchestrator» или изделие) и содержит описание процессов установки и настройки «ROBIN Orchestrator», требования к общесистемному программному обеспечению (ОПО) и вычислительной технике (ВТ), процедуры резервного копирования изделия и восстановления после сбоев, а также диагностические сообщения системному программисту. Функции администратора заключаются в обслуживании АС «ROBIN Orchestrator», включая установку, настройку программных и аппаратных средств, установку информационного обеспечения, управление функциональностью и логикой работы.

Права, порядок доступа, а также результаты работы системного программиста (администратора) в отношении АС «ROBIN Orchestrator» определяет и контролирует администратор безопасности на протяжении всех этапов эксплуатации АС «ROBIN Orchestrator» - от ввода в эксплуатацию до вывода из эксплуатации - в строгом соответствии с требованиями руководящих документов Предприятия по информационной безопасности. В данном документе в разделе «Общие сведения о программе» указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы, а также требования к персоналу. В разделе «Структура программы» приведены сведения о структуре АС «ROBIN Orchestrator», ее составных частях, о связях между составными частями и о связях с другими программными средствами.

Раздел «Установка АС «ROBIN Orchestrator»» описывает условия и порядок развертывания Изделия. Раздел «Параметры запуска» описывает исполняемые файлы серверных компонентов АС «ROBIN Orchestrator» и параметры их запуска. В разделе «Настройка» приведено описание действий по настройке изделия. В разделе «Проверка и восстановление после сбоев» приведено описание действий администратора для проверки работоспособности Изделия с помощью диагностических команд. Также приведено описание действий администратора для восстановления системы после сбоев. В разделе «Сообщения системному программисту» приводится описание диагностических сообщений системному программисту.

## 6.2 Общие сведения

### 6.2.1 Назначение изделия

АС «ROBIN Orchestrator» - комплекс программных средств серверной части Платформы ROBIN, предназначенных для обеспечения взаимодействия компонентов Платформы ROBIN на всех этапах жизненного цикла программных роботов – создания, хранения, изменения, тестирования, эксплуатации.

### 6.2.2 Функциональные возможности изделия

Изделие реализует следующие функциональные возможности:

- выполнять процедуру идентификации и аутентификации пользователей;
- выполнять процедуру аутентификации запускаемых пользователем приложений и процессов;
- выполнять проверку прав доступа субъектов доступа к объектам доступа;
- обеспечивать транспорт служебных сообщений между приложениями и серверными компонентами Изделия;
- выполнять сбор и хранение логов клиентских приложений и серверных компонентов Изделия;
- обеспечивать запуск программных роботов на удаленном хосте по команде оператора или по расписанию;
- выполнять запросы пользователя на публикацию нового робота и его исходника в хранилище роботов и действий,
- вносить записи о новых роботах, как объектов доступа, в каталоги сервера идентификации FreeIPA,
- проксировать запросы к хранилищу роботов и действий на скачивание пакетов с готовыми роботами и их исходниками (при наличии у пользователя соответствующих прав на скачивание).

### 6.2.3 Исполнения изделия

Изделие имеет следующие варианты исполнений:

- АС «ROBIN Orchestrator. Standalone».
- АС «ROBIN Orchestrator. Enterprise».
- АС «ROBIN Orchestrator. Cloud».

АС «ROBIN Orchestrator. Standalone» - пробная версия, предназначенная для ознакомления пользователя с возможностями Изделия. В этом исполнении все серверные компоненты устанавливаются на одном сервере, имеют полный набор функций, но ограниченный период пользования.

АС «ROBIN Orchestrator. Enterprise» - версия, предназначенная для использования на Предприятии. Может быть развернута в пределах защищенного контура корпоративной сети на ресурсах Предприятия (On-Premise). В этом случае собственником, эксплуатантом и выгодоприобретателем является само Предприятие, приобретая права собственности на Изделие в результате его покупки у разработчика.

АС «ROBIN Orchestrator. Cloud» - облачная версия. Система поставляется на рынок как RPA-платформа (PaaS). При этом собственником Системы является Провайдер Системы, а эксплуатантом – контрагенты, заинтересованные в роботизации рутинных бизнес-процессов на своих Предприятиях. Права собственности на Систему Провайдер приобретает в результате ее покупки у разработчика. А контрагенты приобретают право на пользование услугами Системы путем заключения договоров с Провайдером Системы. В версиях АС «ROBIN Orchestrator. Enterprise» и АС «ROBIN

Orchestrator. Cloud» серверные компоненты могут быть развернуты на отдельных серверах как с образованием кластеров, так и без образования кластеров.

#### 6.2.4 Требования к вычислительной технике и общесистемному программному обеспечению

В зависимости от исполнения.

##### Аппаратная конфигурация в исполнении Standalone

Изделие в исполнении АС «ROBIN Orchestrator. Standalone» приобретает для целей оценки состава функциональных возможностей Изделия. Все серверные компоненты Изделия в исполнении АС «ROBIN Orchestrator. Standalone» размещаются на одном сервере со следующими минимальными требованиями к аппаратному обеспечению, указанными в таблице ниже.

Характеристика	Минимальные требования
Количество потоков процессора (vCPU - поток ядра)	8 vCPU
Количество оперативной памяти (RAM)	16 Gb
Емкость дисковой подсистемы (HDD)	250 Gb
Производительность дисковой подсистемы	125 IOPS

##### Аппаратная конфигурация в бескластерном исполнении

В этой конфигурации все серверные компоненты Платформы ROBIN разворачиваются на отдельных серверах со следующими минимальными требованиями к аппаратному обеспечению, указанные в таблице ниже.

##### Аппаратная конфигурация в кластерном исполнении

В этой конфигурации каждый серверный компонент Платформы ROBIN развернут в кластере для обеспечения отказоустойчивости и возможности дальнейшего масштабирования системы.

Минимальная аппаратная конфигурация в исполнении cluster может быть такой:

#### 6.2.5 Требования к персоналу (системному программисту)

Системный программист должен иметь как минимум высшее техническое образование. В перечень задач, выполняемых системным программистом, должны входить:

а) задача поддержания работоспособности технических средств, на которых устанавливается Система; б) задача установки (инсталляции) и поддержания работоспособности общесистемных программных средств – операционной системы, СУБД на узлах Системы; в) задача установки (инсталляции) и поддержания работоспособности на узлах Системы.

Системный программист должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных и технических средств, применяемых в системе.

	Сервер приложений	Число серверов	CPU	RAM, Gb	HDD, Gb
1	WildFly	1	4	8	50
2	PostgreSQL	1	4	4	50
3	FreeIPA	1	4	6	50
4	Sonatype Nexus	1	4	6	100
5	Rabbit MQ	1	4	4	50
6	Elasticsearch	1	4	4	150
7	Logstash	1	2	2	50
8	Kibana	1	4	4	50
9	Robin RDP-Manager	1	2	4	30
	<b>Bcero:</b>	<b>9</b>	<b>32</b>	<b>42</b>	<b>580</b>

Характеристика	Минимальные требования
Количество потоков процессора (CPU pCore- физическое ядро) (vCPU - поток ядра. 1pCore = 2 vCPU)	4 vCPU
Количество оперативной памяти (RAM)	2-6 Gb
Емкость дисковой подсистемы (HDD)	50-200 Gb
Производительность дисковой подсистемы	125 IOPS

Ресурс	CPU	RAM, Gb	HDD, Gb
Elasticsearch 1	4	4	200
Elasticsearch 2	4	4	200
Elasticsearch 3	4	4	200
FreeIPA LDAP 1	4	6	20
FreeIPA LDAP 2	4	6	20
HAProxy backup	4	4	20
HAProxy master	4	4	20
Kibana 1	4	4	20
Kibana 2	4	4	20
Logstash 1	2	2	20
Logstash 2	2	2	20
Nexus 1	4	6	100
Nexus 2	4	6	100
Nexus 3	4	6	100
PostgreSQL master	4	4	50
PostgreSQL slave	4	4	50
RabbitMQ 1	4	4	50
RabbitMQ 2	4	4	50
RabbitMQ 3	4	4	50
RDP Manager Server 1	2	4	20
RDP Manager Server 2	2	4	20
WildFly AppNode 1	4	6	50
WildFly AppNode 2	4	6	50
WildFly AppNode 3	4	6	50
<b>Всего:</b>	<b>88</b>	<b>108</b>	<b>1500</b>

Основными обязанностями администратора безопасности являются: \* модернизация, настройка и поддержка работоспособности комплекса технических средств (серверов, рабочих станций); \* установка, модернизация, настройка и поддержка работоспособности системного и базового программного обеспечения; \* установка, настройка и поддержка работоспособности прикладного программного обеспечения.

Права, порядок доступа, а также результаты работы системного программиста (администратора) в отношении серверных компонент АС «ROBIN Orchestrator» определяет и контролирует администратор безопасности на протяжении всех этапов эксплуатации Системы - от ввода в эксплуатацию до вывода из эксплуатации - в строгом соответствии с требованиями руководящих документов Предприятия по информационной безопасности.

## 6.3 Структура изделия

### 6.3.1 Состав изделия

В состав специального программного обеспечения (СПО) серверной части Платформы ROBIN входят:

- Robin Management Server (RMS) – централизованная служба, выступающая в роли оркестратора. Предоставляет программные интерфейсы (API), посредством которых с ним взаимодействуют клиентские приложения. Работает под управлением сервера приложений WildFly (JBoss Application Server).
- Robin Package Manager – сервис, предназначенный для выполнения клиентских запросов к хранилищу роботов. Работает под управлением сервера приложений WildFly (JBoss Application Server).
- Robin RDP-Manager – сервис, работающий на отдельном сервере. Предназначен для открытия и удержания RDP-сессий. Открытие RDP-сессии на удаленном хосте необходимо для робота, который работает с приложением, имеющим графический интерфейс, эмулируя работу пользователя.
- База данных для серверных приложений RMS – система хранения мета-информации о роботах и расписаниях их запуска. Работает под управлением СУБД PostgreSQL 10+.
- Каталоги сервера аутентификации и авторизации пользователей. Работает под управлением FreeIPA.
- Хранилище пакетов – система хранения артефактов роботов и действий. Работает под управлением Sonatype Nexus.
- Хранилище логов предназначено для хранения логов. Работает под управлением ELK-сервисов (Elasticsearch, Logstash и Kibana).
- Система транспорта сообщений - работает под управлением RabbitMQ.

В состав Платформы ROBIN на стороне конечного пользователя входят следующие программные компоненты:

- Robin Management Console (RMC) – клиентское приложение, предоставляющее пользователю графический интерфейс для централизованного управления запусками роботов на удаленных хостах,
- Robin Studio – клиентское приложение, предоставляющее пользователю графический интерфейс для разработки программных роботов,
- Robin Player – клиентское приложение, предоставляющее пользователю графический интерфейс для управления запусками программных роботов на исполнение сценария,
- Robin Agent – прикладной сервис, расположенный на стороне клиента (Robin Studio или Robin Player):



1. принимает через очереди сообщений программного брокера клиентские запросы со стороны Robin Studio, Robin Player и Robin Management Console,
2. транслирует клиентские запросы системе исполнения сценариев программных роботов Robin Executor,
3. отправляет в RMS через очереди сообщений программного брокера информационные сообщения о событиях в системе исполнения сценариев программных роботов Robin Executor.

### 6.3.2 Алгоритм работы изделия

1. Администратор выполняет установку серверных компонент АС «ROBIN Orchestrator» на требуемых серверах.
2. Администратор с помощью интерфейсов администрирования выполняет настройку серверов приложений и баз данных АС «ROBIN Orchestrator».

## 6.4 Установка

В данном разделе описываются условия и порядок развертывания (установка) серверных компонентов АС «ROBIN Orchestrator» в следующих вариантах исполнений:

- АС «ROBIN Orchestrator. Standalone»,
- АС «ROBIN Orchestrator. Enterprise»,
- АС «ROBIN Orchestrator. Cloud».

Во всех вариантах установка выполняется автоматически с помощью ansible-скриптов (Ansible-playbooks).

### 6.4.1 Порядок и условия развертывания

Администратор должен:

1. получить задание на развертывание серверных компонентов АС «ROBIN Orchestrator»;
2. подготовить ansible-хост, с которого будет выполняться установка;
3. обеспечить условия доступа к целевым серверам с ansible-хоста;
4. выполнить установку с помощью ansible-скриптов,
5. выполнить пост-установочные процедуры.

### 6.4.2 Задание на развертывание серверных компонентов Изделия

Перед началом работы администратор должен получить задание на установку серверных компонентов Изделия, в котором:

- должен быть указан вариант развертывания,
- должна быть схема физической инфраструктуры серверов с указанием ip-адресов всех серверов,
- архив ansible-deploy.tar.gz с дистрибутивом серверных компонентов АС «ROBIN Orchestrator» для выбранного варианта исполнения и ansible-скриптами вместе с конфигурационными файлами.

## Варианты развертывания

Возможны следующие варианты развертывания:

- «Установка Standalone» - установка на одном сервере,
- «Установка Clusterless» - установка на нескольких серверах без образования кластеров,
- «Установка Clusterful» - установка на нескольких серверах с образования кластеров.

### «Установка Standalone»

«Установка Standalone» выполняется для исполнения АС «ROBIN Orchestrator. Standalone». Технические характеристики сервера должны соответствовать требованиям, приведенным в таблице ниже.

Характеристика	Минимальные требования
Количество потоков процессора (vCPU - поток ядра)	8 vCPU
Количество оперативной памяти (RAM)	16 Gb
Емкость дисковой подсистемы (HDD)	250 Gb
Производительность дисковой подсистемы	125 IOPS

### «Установка Clusterless»

«Установку Clusterless» рекомендуется выполнять на Предприятии с небольшим объемом ресурсов и невысокой степенью эксплуатации программных роботов со следующими показателями назначения:

- одновременно работающих в системе пользователей – не более 1 000,
- одновременно работающих в системе программных роботов – не более 1 000,
- число транзакций в секунду для СУБД PostgreSQL – не менее 100 000,
- число запросов в секунду для WildFly – не менее 1000,
- число запросов в секунду для FreeIPA – не менее 100 000,
- число передаваемых сообщений в секунду через подсистему транспорта служебных сообщений – не менее 100 000.

Технические характеристики серверов должны соответствовать требованиям, приведенным в таблице:

### «Установка Clusterful»

«Установка Clusterful» выполняется, как правило, на Предприятии с достаточным объемом ресурсов и высокой степенью эксплуатации программных роботов для исполнения АС «ROBIN Orchestrator. Enterprise» или для исполнения АС «ROBIN Orchestrator. Cloud» со следующими показателями назначения:

- одновременно работающих в системе пользователей – более 10 000,
- одновременно работающих в системе программных роботов – более 10 000,
- число транзакций в секунду для СУБД PostgreSQL – не менее 100 000,
- число запросов в секунду для WildFly – не менее 1000,
- число запросов в секунду для FreeIPA – не менее 100 000,
- число передаваемых сообщений в секунду через подсистему транспорта служебных сообщений – не менее 100 000.

	Сервер приложений	Число серверов	CPU	RAM, Gb	HDD, Gb
1	WildFly	1	4	8	50
2	PostgreSQL	1	4	4	50
3	FreeIPA	1	4	6	50
4	Sonatype Nexus	1	4	6	100
5	Rabbit MQ	1	4	4	50
6	Elasticsearch	1	4	4	150
7	Logstash	1	2	2	50
8	Kibana	1	4	4	50
9	Robin RDP-Manager	1	2	4	30
	<b>Bcero:</b>	<b>9</b>	<b>32</b>	<b>42</b>	<b>580</b>

Технические характеристики серверов должны соответствовать требованиям, приведенным в таблице:

Характеристика	Минимальные требования
Количество потоков процессора (CPU pCore- физическое ядро) (vCPU - поток ядра. 1pCore = 2 vCPU)	4 vCPU
Количество оперативной памяти (RAM)	2-6 Gb
Емкость дисковой подсистемы (HDD)	50-200 Gb
Производительность дисковой подсистемы	125 IOPS

### 6.4.3 Порядок подготовки к установке

Все действия по установке серверных компонентов АС «ROBIN Orchestrator» производятся пользователем с правами root с выделенного управляющего сервера (ansible-хост), имеющего доступ по SSH ко всем управляемым серверам. На управляющем сервере должна быть установлена операционная система CentOS 7.

Порядок подготовки к установке серверных компонентов АС «ROBIN Orchestrator» заключается в следующем:

1. Выполнить установку необходимых пакетов на ansible-хост.
2. Распаковать архив ansible-deploy.tar.gz с дистрибутивом серверных компонентов АС «ROBIN Orchestrator»
3. Выполнить настройку конфигурационных параметров ansible-скриптов.

4. Сгенерировать сертификат для балансировщика и установить его в нужное место.
5. Выполнить настройку доступа по SSH ко всем управляемым серверам.
6. Открыть требуемые порты для балансировщика.

#### 6.4.4 Ansible-хост

##### Установка пакетов

Поскольку установка серверных компонентов AC «ROBIN Orchestrator» выполняется автоматически с помощью ansible-скриптов (ansible-playbooks), то прежде всего необходимо установить пакет ansible и другие необходимые для него пакеты:

```
yum install epel-release python-netaddr unzip yum install ansible
```

Выполнить установку дополнительного модуля jbosscli, находясь в директории скриптов:

```
mkdir -p /usr/share/ansible/plugins/modules/ sudo cp ./files/jbosscli.py /usr/share/ansible/plugins/modules/
```

##### Распаковка архива

Архив ansible-deploy.tar.gz с дистрибутивом серверных компонентов AC «ROBIN Orchestrator» необходимо распаковать в какой-нибудь директории, например: `cd /opt tar xzf ansible-deploy.tar.gz`

Структура файлов и папок в директории ansible-deploy:

```
[svolin@tmp-dev-ans3 ansible-deploy]$ ls -l
total 36
-rw-r--r--. 1 root root 352 Jul 21 18:06 ansible.cfg
drwxr-xr-x. 3 root root 4096 Dec 9 17:40 files
drwxr-xr-x. 2 root root 22 Dec 4 18:45 group_vars
-rw-r--r--. 1 root root 667 Dec 9 16:45 hosts2.ini
-rw-r--r--. 1 root root 631 Dec 8 17:53 hosts.ini
-rw-r--r--. 1 root root 913 Dec 2 20:38 packages.yaml
-rw-r--r--. 1 root root 3603 Dec 4 18:21 README.md
drwxr-xr-x. 17 root root 242 Dec 6 18:55 roles
drwxr-xr-x. 2 root root 117 Dec 4 18:45 scripts
-rw-r--r--. 1 root root 1798 Dec 8 16:32 start.yml
drwxr-xr-x. 2 root root 4096 Dec 9 17:47 templates
-rw-r--r--. 1 root root 377 May 3 2021 wf-restart.yaml
```

Описание файлов и директорий дистрибутива приведено в таблице:

Имя файла или директории	Назначение	Размещение
ansible.cfg	Задаёт локальные настройки Ansible в том числе файл инвентаризации	./ - корневой каталог скриптов
hosts.ini	Файл инвентаризации - задаёт целевые хосты, на которых будут устанавливаться серверные компоненты. Хосты могут быть объединены в группы.	./ - корневой каталог скриптов
start.yml	Главный ansible-скрипт, который устанавливает серверные компоненты	./ - корневой каталог скриптов
wf-restart.yml	Ansible-скрипт для рестарта службы wildfly.service	./ - корневой каталог скриптов
all.yml	Задаёт общие и специальные настройки (пароли, пути к хранилищам сертификатов, имена устанавливаемых пакетов и т.д.) серверных компонентов	./group_vars/
./files	Директория содержит пакеты с дистрибутивами общего и специального программного обеспечения, скрипты, публичные ключи и другие артефакты, необходимые для выполнения процедуры установки серверных компонент	./
./roles	Содержит ansible-роли с задачами по установке серверных компонент. Подробности о том, что такое ansible-роли, можно найти в документации по Ansible: <a href="https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html">https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html</a>	./
auditbeat.yml.j2	Jinja-шаблон для задачи по установке компонента auditbeat	./templates/
elasticsearch-single.yml.j2	Jinja-шаблон для задачи по установке Elasticsearch без кластера на одном сервере	./templates/
elasticsearch.yml.j2	Jinja-шаблон для задачи по установке Elasticsearch в кластере	./templates/
filebeat.yml.j2	Jinja-шаблон для задачи по установке компонента filebeat	./templates/
haproxy.cfg.j2	Jinja-шаблон для задачи по установке компонента HAProxy	./templates/
haproxy-rsyslog.conf	Jinja-шаблон для задачи по установке компонента HAProxy и службы rsyslog	./templates/
haproxy-standalone.cfg.j2	Jinja-шаблон для задачи по установке HAProxy без кластера на одном сервере	./templates/
kibana.yml.j2	Jinja-шаблон для задачи по установке компонента Kibana	./templates/
logstash.conf.j2	Jinja-шаблон для задачи по установке компонента Logstash	./templates/
nexus.service	Jinja-шаблон с настройками Nexus-сервиса	./templates/
nginx-repo.conf.j2	Jinja-шаблон с настройками Nginx-сервиса	./templates/
rdpmanager.application.properties.j2	Jinja-шаблон с настройками Nginx-сервиса RDP-менеджера	./templates/
rms2-infinispan.xml	Jinja-шаблон с параметрами распределенного кэша infinispan сервиса rms2	./templates/
rms2.properties	Jinja-шаблон с параметрами сервиса rms2	./templates/
rpm-infinispan.xml	Jinja-шаблон с параметрами распределенного кэша infinispan сервиса rpm (для wf-standalone.yml и wildfly.yml)	./templates/
rpm.properties	Jinja-шаблон с параметрами сервиса rpm (для задачи по установке WildFly)	./templates/

## Настройка конфигурационных параметров

Настройка конфигурационных параметров должна выполняться на основе данных, содержащихся в схеме физической инфраструктуры серверов.

### Инвентаризационный файл hosts.ini

Инвентаризационный файл hosts.ini (подробности [https://docs.ansible.com/ansible/2.3/intro\\_inventory.html](https://docs.ansible.com/ansible/2.3/intro_inventory.html)) содержит имя пользователя (ansible\_user), под учетной записью которого будет выполняться вход в операционную систему управляемых серверов, имена (алиасы) и адреса управляемых серверов (ansible\_host). Подключение ко всем управляемым серверам выполняется по протоколу SSH на порт 22 (по умолчанию). Если подключение по SSH выполняется по другому порту, то этот порт нужно указать после адреса сервера через символ «:».

Секция	Alias переменной	Имя переменной	Значение переменной
[all:vars]		ansible user	centos
[rabbit]	rbt1	ansible_host	10.172.192.27
[app]	wf1	ansible host	10.172.192.27
[elastic]	elastic1	ansible_host	10.172.192.27
[kibana]	kibana1	ansible host	10.172.192.27
[logstash]	logstash1	ansible host	10.172.192.27
[db]	db1	ansible_host	10.172.192.27
[pkg]	pkg1	ansible host	10.172.192.27
[rdp]	rdp1	ansible_host	10.172.192.27
[lb]	lb1	ansible host	10.172.192.27
[repository]	repo	ansible host	10.172.192.27
		ansible_connection	local

### Конфигурационный файл group\_vars/all.yaml

Конфигурационный файл group\_vars/all.yaml содержит общие параметры и параметры, специфичные для каждого управляемого сервера.

Секция	Имя переменной	Значение переменной	Назначение
Общие параметры	standalone_mode	true	Установка в режиме «во
	lb_address	testinstall. robin.demo	Адрес балансировщика
	local_install	true	Использовать установку
	repo_path	/opt/ansible-deploy/files/repo	Путь к директории с па
FreeIPA	domain	robin.demo	Домен LDAP
	base_dn	dc=robin,dc=demo	Base DN
	ldap_autoinit	true	Инициализировать LDA
	admin_pass	Qwerty123	Пароль пользователя ас
	dm_pass	Qwerty123	Пароль пользователя D
	ldap_use_ssl	false	Использовать SSL для п

Таблица 1 – продолжение с предыду

	ldap1_fqdn, ldap2_fqdn, ...	testinstall. robin.demo	Полные адреса серверов
DB	db_postgres_pass	Qwerty123	Пароль пользователя ро
	db_robin_pass	Qwerty123	Пароль пользователя ро
	db_users_pass	Qwerty123	Пароль пользователей ч
	flyway_ver	7.8.1	Версия пакета FlyWay
RabbitMQ	rabbit_admin_pass	Qwerty123	Пароль пользователя ад
	erlang_cookie	7KzILFz03fmS	Erlang cookie для связи
	rabbit_agent_pass	Qwerty123	Пароль пользователя аг
	rabbit_rmc_pass	Qwerty123	Пароль пользователя гр
	rabbit_rms_pass	Qwerty123	Пароль пользователя гр
	rabbit_rdp_pass	Qwerty123	Пароль пользователя ро
Nexus	nexus_ver	3.32.0-03	Версия пакета Nexus
	nexus_port	8081	Порт для доступа к Nex
	nexus_admin_pwd	Qwerty123	Устанавливаемый парол
Wildfly	wf_version	23.0.2	Версия пакета
	schema_version	16.0	Версия схемы, зависит о
	wf_admin_pass	Qwerty123	Пароль пользователя ад
	wf_slave_secret	RFB5N0NMRSEyVmc2KzI=	Хэш пользователя slave
HAProxy	lb_cert	/etc/ssl/certs/lb.pem	Путь, где будет хранит
Файлы/Пакеты	auditbeat_file	auditbeat-7.15.1-x86_64.rpm	Имя пакета auditbeat
	filebeat_rpm	filebeat-7.15.0-x86_64.rpm	Имя пакета filebeat
	haproxy_file	haproxy-2.4.7-1.el7.x86_64.rpm	Имя пакета HAProxy
	kibana_objects_file	kibana_objects-2112021320.ndjson	Имя файла с объектами
	ldif_file	ldif-2111242015.zip	Имя файла с форматом
	repository_war_file	repository-2112021320.war	
	rms_public_file	rms-public-2112021320.pub	
	rms2_war_file	rms2-2112021320.war	Имя файла с приложен
	rms2_war_file	rms2-2112021320.war	Имя файла с приложен
	robinrdpmanager_file	robin-rdp-manager-1.0-1.x86_64.rpm	Имя пакета с приложен
	rpm_war_file	rpm-2112021320.war	Имя пакета с приложен
	rpms_repo_file	rpms-repo-2112041757.tar.gz	
	sql_migrations_file	sql_migrations-2110041449.zip	
	wf_driver_pg_file	wf_driver_pg-2103231154.zip	
	wf_driver_rms2_file	wf_driver_rms2-2012181023.zip	
	wf_driver_logstash_file	wildfly-logstash-1.0.9-SNAPSHOT.zip	

После того как все конфигурационные параметры в файлах `./hosts.ini` и `./group_vars/all.yaml` будут заданы, необходимо будет выполнить: генерацию сертификата для балансировщика, настройку доступа по SSH, открытие портов, установку серверных компонентов.

### Генерация сертификата для балансировщика

Сертификат должен быть выпущен Удостоверяющим Центром, выбранным по усмотрению Заказчика. Сертификат быть создан также как самоподписанный. Актуальный сертификат для балансировщика необходимо положить в директорию `files` вместо `files/lb.pem`.

### Настройка доступа по SSH

Предпочтительным способом подключения к управляемым серверам по SSH является способ с использованием SSH-ключей и `ssh-agent`. Для этого необходимо:

1. На `ansible`-хосте сгенерировать пару SSH-ключей с помощью команды: `ssh-keygen -b 4096`
2. Установить публичный ключ на всех управляемых серверах с помощью утилиты `ssh-copy-id` или любым иным возможным способом, известным администратору.

3) Проверить SSH доступ с помощью команды: `ssh your_username@your_server_ip`. Если доступ по SSH ко всем управляемым серверам обеспечен, то можно переходить к следующему шагу.

### Открытие портов

Прежде всего должен быть открыт порт 22 на всех управляемых серверах для доступа к ним с ansible-хоста по SSH протоколу. В случае, если используется межсетевой экран, необходимо открыть требуемые порты для балансировщика: 80, 8081, 9080, 9443, 10990, 5671, 5672, 15672, 443, 389, 636.

### Установка серверных компонентов

Для установки серверных компонентов АС «ROBIN Orchestrator» необходимо выполнить: `ansible-playbook start.yaml`

## 6.5 Параметры запуска сервисов

### 6.5.1 Сервисы

Для обеспечения работоспособности серверных компонент АС «ROBIN Orchestrator» необходимо, чтобы на серверах были запущены следующие сервисы:

- auditbeat,
- elasticsearch,
- filebeat,
- haproxy,
- ipa.service,
- kibana,
- logstash,
- nginx.service,
- nexus.service,
- postgresql-11,
- rabbitmq-server.service,
- robin-xvfb,
- robin-rdp-manager,
- wildfly.service.

### 6.5.2 Автозапуск

Все сервисы во время установки серверных компонентов сконфигурированы на автозапуск – при загрузке операционных систем серверов все сервисы запускаются автоматически.

### 6.5.3 Штатный останов сервисов

Для штатного останова сервиса необходимо выполнить команду: `sudo systemctl stop <service_name>` где вместо `<service_name>` необходимо использовать имя соответствующего сервиса. Команда выполняется пользователем с правами root.



### 6.5.4 Штатный запуск сервисов

Для штатного запуска сервиса необходимо выполнить команду: `sudo systemctl start <service_name>` где вместо `<service_name>` необходимо использовать имя соответствующего сервиса. Команда выполняется пользователем с правами root.

### 6.5.5 Штатный перезапуск сервисов

Для штатного перезапуска сервиса необходимо выполнить команду: `sudo systemctl restart <service_name>` где вместо `<service_name>` необходимо использовать имя соответствующего сервиса. Команда выполняется пользователем с правами root.

### 6.5.6 Особенности выполнения команд по запуску, останову, перезапуску в различных исполнениях вариантах развертывания

В варианте развертывания «Установка Standalone» все команды по запуску, останову, перезапуску всех сервисов выполняются на одном сервере. В варианте развертывания «Установка Clusterless» все команды по запуску, останову, перезапуску сервиса выполняются на том сервере, на котором этот сервис установлен. В варианте развертывания «Установка Clusterful» все команды по запуску, останову, перезапуску сервиса выполняются на всех серверах кластера, в котором эти сервисы установлены.

## 6.6 Конфигурирование и настройка

### 6.6.1 Пост-установочные процедуры

После установки серверных компонентов AC «ROBIN Orchestrator» необходимо выполнить пост-установочные процедуры конфигурирования некоторых сервисов.

### 6.6.2 WildFly

#### rpm.properties

На каждом из серверов Wildfly в файле `/opt/wildfly/modules/rms2/main/properties/rpm.properties` найти параметр `nexus.nodes` и прописать в нем через запятую для всех автономных серверов Nexus их IP и соответствующие Nexus ID в формате:

```
nexus.nodes=ID1&IP1,ID2&IP2,...,IDN&IPN
```

где Nexus ID и IP имеют вид:

```
ID=XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX IP=http://127.0.0.1:8081
```

Nexus ID доступны на вкладке Nexus -> Support -> System Information -> node-id (доступно для автоматизации только в Pro версии нексуса). После внесения изменений в файл `rpm.properties` необходимо перезапустить службу Wildfly: `sudo systemctl restart wildfly.service`

#### Пользователи

Сгенерировать свои секреты и хэши для Wildfly и прописать их в конфиг (трижды запустить `add-user.sh`) согласно инструкции (опционально).

#### Пользователь управления

Далее необходимо создать пользователя управления, который сможет подключаться с помощью консоли администрирования или удаленно с помощью интерфейса командной строки. Чтобы добавить нового пользователя, используйте `add-user.sh` скрипт, который находится в каталоге `bin` WildFly:

```
sudo /opt/wildfly/bin/add-user.sh
```

Вас спросят, какого типа пользователя вы хотите добавить:

```
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
```

Выберите `a` и нажмите `Enter`. Далее скрипт предложит вам ввести данные нового пользователя:

- имя пользователя – ввести имя пользователя,
- пароль – ввести пароль,
- повторить пароль – повторно ввести пароль,
- добавить пользователя в группу управления „ManagementRealm“ – оставить без изменения,
- на вопрос: «Is this new user going to be used for one AS process to connect to another AS process?» ответить – `no`.

```
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username: admin
Password recommendations are listed below. To modify these restrictions edit the add-
user. properties configuration file.
- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, ad
ministrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 di
git(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list,
or leave blank for none)[ ]:
About to add user 'admin' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'admin' to file '/opt/wildfly-16.0.0.Final/standalone/configuration/mgmt-u
sers.properties'
Added user 'admin' to file '/opt/wildfly-16.0.0.Final/domain/configuration/mgmt-users
.properties'
Added user 'admin' with groups  to file '/opt/wildfly-16.0.0.Final/standalone/configu
ration/mgmt-groups.properties'
Added user 'admin' with groups  to file '/opt/wildfly-16.0.0.Final/domain/configurati
on/mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS process
?
e.g. for a slave host controller connecting to the master or for a Remoting connectio
n for server to server EJB calls.
yes/no? no
```

Новый пользователь будет добавлен в файлы свойств, используемые для аутентификации.

Далее необходимо добавить пользователя, от имени которого подчиненный хост-контроллер (slave) будет подключаться к мастеру (master), или для удаленного подключения для EJB- вызовов сервера к серверу. Чтобы добавить нового пользователя, используйте add-user.sh скрипт, который находится в каталоге bin WildFly:

```
sudo /opt/wildfly/bin/add-user.sh
```

Вас спросят, какого типа пользователя вы хотите добавить:

```
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
```

Выберите а и нажмите Enter. Далее скрипт предложит вам ввести данные нового пользователя:

- имя пользователя – ввести имя пользователя,
- пароль – ввести пароль,
- повторить пароль – повторно ввести пароль,
- добавить пользователя в группу управления „ManagementRealm“ – оставить без изменения,
- на вопрос: «Is this new user going to be used for one AS process to connect to another AS process?» ответить – yes.

Новый пользователь будет добавлен в файлы свойств, используемые для аутентификации. Будет сгенерирован хэш вида <secret value=»UGFuMjNkZWozNyNA» />, который нужно сохранить для дальнейшего добавления в конфиг.

### Конфигурирование slave-хостов

В файле /opt/wildfly/domain/configuration/host.xml на каждом из slave-серверов:

1. Задаем имя хоста. Оно может быть любым, но должно быть уникальным внутри домена:
2. Добавляем полученный хэш в ManagementRealm:
3. Указываем адрес контроллера домена «\${jboss.domain.master.address:192.168.1.1}» и имя пользователя «slave»:
4. Прописываем соответствующий адрес интерфейса public:

### 6.6.3 LDAP

Для всех клиентских приложений необходимо прописать актуальные параметры для подключения к LDAP серверу. Процедура конфигурирования и настройки клиентских приложений выполняется с помощью утилиты RobinLdbConfigUpdater.exe, которая входит в состав дистрибутивов для соответствующих редакций:

- Robin STUDIO:

Редакция Desktop, Редакция Enterprise, Редакция Cloud,

- Robin ROBOT:

Редакция Desktop, Редакция Enterprise, Редакция Cloud,

- Robin RMC:

```

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : slave
Password recommendations are listed below. To modify these restrictions edit the add-
user.properties configuration file.
- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, ad
ministrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 di
git(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list,
or leave blank for none)[ ]:
About to add user 'slave' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'slave' to file '/opt/wildfly-16.0.0.Final/standalone/configuration/mgmt-u
sers.properties'
Added user 'slave' to file '/opt/wildfly-16.0.0.Final/domain/configuration/mgmt-users
.properties'
Added user 'slave' with groups to file '/opt/wildfly-16.0.0.Final/standalone/configu
ration/mgmt-groups.properties'
Added user 'slave' with groups to file '/opt/wildfly-16.0.0.Final/domain/configurati
on/mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS process
?
e.g. for a slave host controller connecting to the master or for a Remoting connectio
n for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secret v
alue="UGFuMjNkZWw3NyNA" />

```

```
<host xmlns="urn:jboss:domain:16.0" name="slave-1">
```

```

    <security-realm name="ManagementRealm">
        <server-identities>
            <secret value="UGFuMjNkZWw3NyNA"/>
        </server-identities>
        ""
    </security-realm>

```

```

<domain-controller>
    <remote                                security-realm="ManagementRealm"
username="slave">
        <discovery-options>
            <static-discovery                name="primary"
protocol="${jboss.domain.master.protocol:remote+http}"
host="${jboss.domain.master.address:192.168.1.1}"
port="${jboss.domain.master.port:9990}"/>
        </discovery-options>
    </remote>
</domain-controller>

```

```
<interfaces>
  <interface name="management">
    <inet-address value=" 127.0.0.1"/>
  </interface>
  <interface name="public">
    <inet-address value="192.168.1.2"/>
  </interface>
</interfaces>
```

Редакция Enterprise, Редакция Cloud.

---

## Инструкция по созданию пользовательских действий на C#

---

### 7.1 Аннотация

Инструкция предназначена для разработчика пользовательских действий на платформе Robin с использованием языка программирования C# и содержит описание решения следующих задач:

- создание проекта действия,
- создание артефактов действия,
- тестирование и логирование действия,
- как интегрировать действие в ROBIN Studio.

### 7.2 Основные понятия

Таблица 1: Понятия и определения

Понятие	Описание
Действие	Функция (операция), повторно выполняемая пользователем с помощью компьютерной техники для решения своих повседневных задач, и реализованная в программном коде.
Контракт действия	Строковая константа заданного формата. Позволяет однозначно идентифицировать функциональный тип действия в системе.
Реализация действия	Программный код, реализующий функцию действия.
Версия реализации действия	Реализация действия может иметь несколько версий для каждого языка.
Платформа (тип) реализации действия	Язык программирования (Java, C#, Python), на котором реализовано действие.
Версия действия	Версия действия на пользовательском уровне. Скрывает от пользователя версию реализации и тип реализации действия.
Супер-действие	Чисто техническая сущность - создается в файле ListActions.xml, используется для регистрации пользовательских действий, их версий и платформ реализации в системе.

Продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Понятие	Описание
Определение действия	Обязательный артефакт действия, содержащий описание действия, а также его входных параметров и результата исполнения действия. Работа с описанием действия происходит только через интерфейс редактора действий. Этот файл зашифрован и хранится в пакете (архиве), его прочитать нельзя из файловой системы.
Описание реализации действия	Описание реализации действия хранится в файле <code>.robin-impinfo</code> . Файл хранится в корне проекта действия и является обязательным артефактом действия. Содержит следующую информацию: <ul style="list-style-type: none"> <li>• наименование контракта действия,</li> <li>• версия реализации действия,</li> <li>• тип платформы выполнения (Java, C#, Python),</li> <li>• версия схемы <code>ActionImpScheme_ActionImp.xsd</code>,</li> <li>• перечень интерфейсов и исключений Robin Action SDK, используемых в реализации действия</li> </ul> Более детальная информация содержится в схеме <code>ActionImpScheme_ActionImp.xsd</code> - см. Приложение 1.
Проект робота	Проект содержит функциональный план решения прикладной задачи с использованием функциональных возможностей действий. Задаёт последовательность выполнения действий с заданными наборами входных и выходных параметров для каждого действия из этой последовательности.
Проект действия	Проект, содержащий мета-информацию о действии и его пакеты
Maven-проект	Содержит код действия, реализованный на Java, а также его артефакты
.Net-проект	Содержит код действия, реализованный на C#, а также его артефакты
Python-проект	Содержит код действия, реализованный на Python, а также его артефакты
Артефакты действия	<ul style="list-style-type: none"> <li>• Пакет с исполняемым кодом действия</li> <li>• Пакет с исходником действия</li> <li>• Пакет с документацией действия для студии</li> <li>• XML-файл с определением действия</li> <li>• XML-файл с описанием реализации действия</li> <li>• Запись о действии в супер-действии в файле <code>ListActions.xml</code></li> <li>• Запись о супер-действии в файле <code>ListGroups.xml</code></li> </ul>

## 7.3 Общая схема по созданию и интеграции пользовательских действия с ROBIN Studio

### 7.3.1 Необходимое ПО

Для создания .net-действий на компьютере разработчика должно быть установлено следующее ПО:

- .net framework 4.8 sdk - <https://dotnet.microsoft.com/download/visual-studio-sdks>
- Visual Studio 2019 - <https://visualstudio.microsoft.com/ru/>
- Robin.SDK в составе:
  - набор стандартных библиотек и классов, необходимый для разработки действия на платформе Robin,

- Robin.ActionEditor - редактор действия,
- Robin.ActionLoader - утилита импорта пакетов с действиями в Робин Студию,
- ActionImpScheme\_ActionImp.xsd - XSD-схема описания реализации действия.

### 7.3.2 Общий порядок разработки пользовательского действия

При разработки .net-действия и его интеграции в Студию следует придерживаться следующего порядка:

1. До начала разработки действия или уже в процессе его разработки разработчик должен определиться с его контрактом и его артефактами:
  - наименование контракта
  - наименование действия
  - описание действия
  - описание входных параметров
  - описание выходных параметров
  - svg-иконка
2. Если разработчик определился с контрактом действия до его разработки, то он может сразу же приступить к созданию Проекта действия, а потом к его реализации в программном коде.
3. Если разработчик планирует определиться с контрактом действия в процессе его разработки, то он может приступить к созданию Проекта действия уже после завершения его реализации в программном коде.
4. С помощью Robin.ActionEditor разработчик:
  - заполняет форму для создания определения нового действия, в котором указаны:
    - наименование контракта
    - описание входных параметров
    - описание выходных параметров
  - выполняет сохранение определения действия в файл \*.robin-action (в процессе сохранения создается папка с проектом действия для размещения в нем пакетов и атрибутов действия)
5. В IDE разработчика:
  - создать новый проект на платформе .Net и реализовать в нем действие
  - создать файл с описанием реализации действия \*.robin-impinfo и разместить его в проекте вместе с кодом
  - создать пакеты со скомпилированным кодом, с транзитивными зависимостями и исходным кодом (только в Enterprise)
6. Далее разработчик должен импортировать из проекта программного кода в Проект действия следующие пакеты:
  - Пакет с реализацией действия
  - Пакеты с транзитивными зависимостями действия
  - Пакет с исходным кодом действия (только в Enterprise)
7. Добавить действие через существующее или новое супер-действие в ListActions.xml
8. Если было создано новое супер-действие, то его необходимо добавить в ListGroups.xml в подгруппу или группу супер-действий.



9. С помощью редактора действий `Robin.ActionEditor` создать дистрибутив - файл с расширением `*.robin-package`.
10. С помощью утилиты импорта `Robin.ActionLoader` выполнить публикацию пакетов в Робин Студию.

### 7.3.3 Контракт действия

До начала разработки действия (или уже в процессе его разработки) разработчик должен определиться с его контрактом. Для этого он должен:

- придумать или использовать уже существующее наименование вендора
- придумать или использовать уже существующую группу действий, которой будет принадлежать новое действие
- придумать наименование нового действия
- определиться с платформой, на которой разработчик предполагает создавать действие

Формат наименования контракта:

`<Vendor>.<Group>.<ActionName>`

- `<Vendor>` - наименование компании разработчика действия
- `<Group>` - наименование группы действия
- `<ActionName>` - название действия

Во всех названиях действий каждое слово выделяется заглавной буквой, например: `MyCustomAction`.

### 7.3.4 Создание Проекта действия

#### Структура проекта действия

Создание Проекта действия включает в себя создание структуры папок проекта и подготовку следующих пакетов:

- Пакет с определением действия (+ документация в html внутри)
- Пакет с реализацией действия
- Пакеты с транзитивными зависимостями действия
- Пакет с исходным кодом действия (только в Enterprise)
- `ListActions.xml` - файл со списком супер-действий
- `ListGroups.xml` - файл со списком групп

Создание действия начинается с его определения. Создание определения происходит с помощью `Robin.ActionEditor`, который, помимо создания пакета, создаёт ещё и проект с действием, имеющий особую структуру папок.

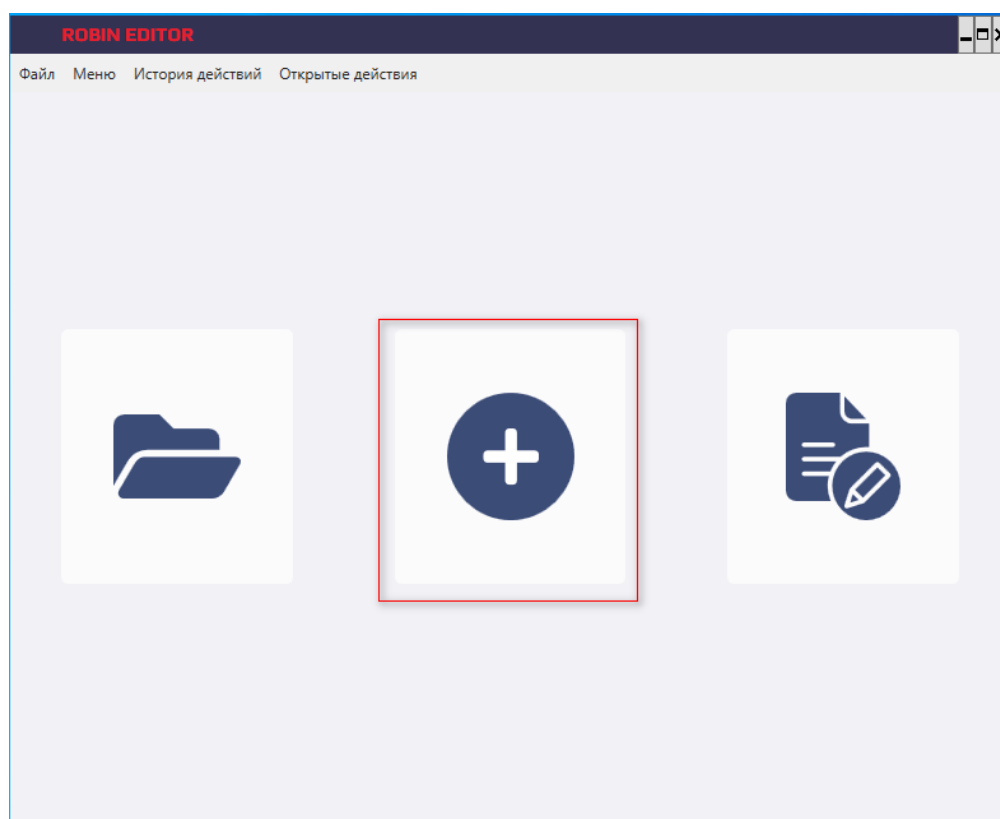
Имя папки с проектом действия задаётся в `Robin.ActionEditor` при создании проекта. Местоположение по умолчанию - папка `Robin Actions` в документах пользователя.

#### Создание проекта действия

Для создания Проекта действия необходимо выполнить следующие действия:


1. Запустить `Robin.ActionEditor`

2. Меню/Создать (нажать на «+»):



3. Заполнить форму:

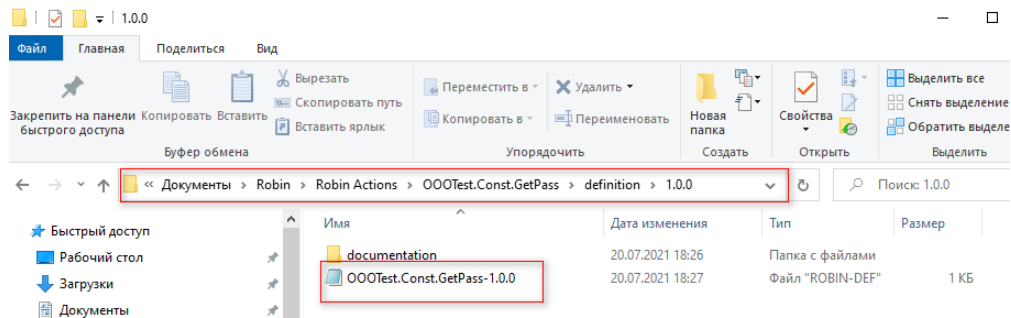
### Создание проекта действия

	Компания	Группа	Действие
Контракт действия:	<input type="text" value="000Test"/>	<input type="text" value="Const"/>	<input type="text" value="GetPass"/>
Версия	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Директория	<input type="text" value="C:\Users\serge\Documents\Robin\Robin Actions"/> 		
Платформа	<input checked="" type="checkbox"/> Net	<input type="checkbox"/> Java	<input type="checkbox"/> Python
<div><input type="button" value="Создать"/> <input type="button" value="Отмена"/></div>			

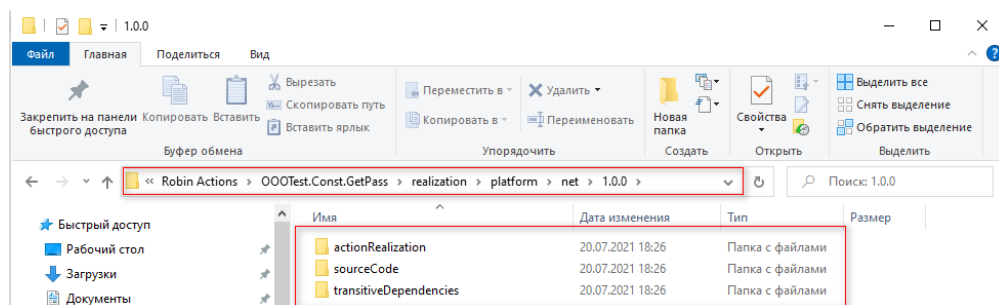
4. Нажать «Создать». В результате в папке `C:\Users\%USER%\Documents\Robin\Robin Actions\000Test.Const.GetPass` будет создан Проект действия со следующим списком артефактов:

- `000Test.Const.GetPass.action-project` - файл с Проект действиям
- `ListActions.xml` - XML-файл с описанием всех действий в проекте

- ListGroups.xml - XML-файл с описанием всех групп действий в проекте
- \definition\1.0.0\OOOTest.Const.GetPass-1.0.0.robin-def - шаблон XML-файла с описанием действия:



и папки для хранения файла с реализацией, исходников и зависимостей действия:



### Создание определения действия (.robin-action)

После формирования папки проекта `Robin.ActorEditor` позволяет пользователю создать определение действия. Для этого пользователь указывает наименование группы действий и название действия, которые будут потом использованы при формировании контракта действия. Наименование компании указано в конфигурации `Robin.ActorEditor`, а для варианта Enterprise - в LDAP. Начальная версия - 1.0.0.

`Robin.ActorEditor` формирует пакет с определением действия (архив), имеющий следующее название:

```
[компания] . [группа действий] . [действие] - [версия] .robin-def
```

Пример наименования пакета: `Robin.Dialogs.ShowMessage-1.0.0.robin-def`

		Описание	Допустимые значения
Контракт	Компания	Наименование компании, разработавшей определение действия	Буквы латинского алфавита, цифры, нижнее подчёркивание, тире
	Группа действий	Наименование группы, к которой относится действие	Буквы латинского алфавита, цифры, нижнее подчёркивание, тире
	Действие	Наименование действия	Буквы латинского алфавита, цифры, нижнее подчёркивание, тире
Версия	Версия	Версия определения действия в формате X.Y.Z	Цифры, точка

Чтобы создать XML-файл с определением действия, необходимо с помощью `Robin.ActorEditor` выполнить следующие действия:

1. Нажать Меню/Открыть
2. Выбрать файл с проектом действия - `MyCompany.CustomActions.CustomAction.action-project`:

## 7.3. Общая схема по созданию и интеграции пользовательских действия с ROBIN Studio277

3. В результате откроется форма, в которой нужно последовательно заполнить следующие разделы определения действия:

- Параметры
- Результат

4. Завершение работы с редактором:

- Нажать «Файл/Сохранить» для сохранения определения действия

Файл с определением действия `000Test.Const.GetPass-<x.y.z>.robin-def` будет сохранен в папку `C:\Users\<user>\Documents\Robin\Robin Actions\000Test.Const.GetPass\definition\<x.y.z>\`,

где `<x.y.z>` - версия действия, начальная версия - 1.0.0, `<user>` - имя пользователя на компьютере разработчика.

## Документация действия

Документация действия подготавливается разработчиком в виде html-документа и кладется в папку `C:\Users\<user>\Documents\Robin\Robin Actions\000Test.Const.GetPass\definition\<x.y.z>\documentation`

## Создание описания реализации действия (.robin-impinfo)

Файл с описанием реализации действия имеет расширение `.robin-impinfo`, а его имя формируется по следующему правилу:

```
<Vendor>.<GroupName>.<ActionName>-<ExecType>.robin-impinfo
```

где:

- `<Vendor>` - название вендора действия
- `<GroupName>` - название группы действий, к которой принадлежит действие
- `<ActionName>` - название действия
- `<ExecType>` - платформа, на которой реализовано действие

Файл хранится в проекте .Net-действия.

Далее приведены рекомендованные и обязательные элементы и их атрибуты для реализаций, а также их назначения в файле описания реализации действия:

Таблица 2: Элементы и атрибуты XML-реализаций

Имя элемента	Атрибуты элемента	Родитель элемента	Обязательность	Описание значения
ImplementationInfo			Да	Корневой элемент описания реализации действия
	contract		Да	<code>&lt;vendor-name&gt;.&lt;action-group&gt;.&lt;action-name&gt;</code>
	version		Да	Версия действия
	execType		Да	Тип реализации
	schemaVersion		Да	Версия xml-схемы описания реализации действия
Author		ImplementationInfo	Да	Автор
RequiredInterfaces		ImplementationInfo	Да	Обязательный элемент. Должен быть даже, если ни одного требуемого интерфейса нет

Продолжается на следующей странице

Таблица 2 – продолжение с предыдущей страницы

RequiredInterface		RequiredInterfaces	Нет	Полное название, включая namespace (или имя пакета, в зависимости от платформы) интерфейса, который требуется действию для его работы.
RequireUiThread		ImplementationInfo	Да	Флаг говорящий о том, что данному действию необходим для работы графический контекст. Если реализация действия взаимодействует с UI, данное значение должно true
Exceptions		ImplementationInfo	Да	Список исключений, которое может генерировать данное действие
ExceptionInfo		Exceptions	Нет	Исключение
DeprecatedVersions		ImplementationInfo	Нет	Список запрещенных версий реализаций данного действия
DeprecatedVersion		DeprecatedVersions	Нет	Запрещенная версия реализации действия
AssemblyName		ImplementationInfo	Нет	Для .NET - полное имя сборки. Элемент специфичен только для платформы .NET
PackageName		ImplementationInfo	Нет	Имя пакета содержащего реализацию действия
CommitHash		ImplementationInfo	Нет	Хеш комита из которого собрано действие
ListPermittedOS		ImplementationInfo	Нет	Список разрешенных версий ОС
PermittedOS		ListPermittedOS	Нет	Разрешенная версия ОС
TypePermittedOS		PermittedOS	Нет	Тип, представляющий информацию о релизах ОС
OSRelease		PermittedOS	Нет	Номер релиза
OSType		PermittedOS	Нет	Разрядность ОС
OSDistributorID		PermittedOS	Нет	Название дистрибутива

Особенности создания файла **.robin-impinfo** для каждой платформы описаны в соответствующих каждой платформе разделах по созданию действий

### 7.3.5 Создать действие

Подробные инструкции создания действия на языках программирования располагаются в разделе *Создание .Net-действия*

### 7.3.6 Импорт действия из .Net-проекта в проект действия

Далее разработчик подготавливает пакеты:

- Пакет с реализацией действия
- Пакеты с транзитивными зависимостями действия
- Пакет с исходным кодом действия (только в Enterprise)

и копирует их в структуру Проекта действия.

### 7.3.7 ListActions.xml

Для того чтобы добавить новое действие, нужно либо создать новое супер-действие в `ListActions.xml`, либо включить действие в уже существующее супер-действие.

Намерения пользователя	Действия пользователя
Добавление действия в существующее супер-действие	Сначала <code>ListActions.xml</code> выгружается из кэша студии с помощью утилиты импорта, выгруженный файл включает в себя только список супер-действий без конкретных реализаций. После этого пользователь добавляет туда новое действие
Добавление действия в новое супер-действие	У пользователя есть пример <code>ListActions.xml</code> , поставляемый в качестве примера в составе SDK. После этого пользователь добавляет туда новое действие

`ListActions.xml` содержит корневой XML-элемент `<ListSuperActions>`, внутри которого располагаются один или несколько XML-элементов `<SuperAction>`.

Разработчик для своего нового действия должен создать новый `<SuperAction>` или воспользоваться имеющимся, если он есть.

`<SuperAction>` имеет следующую структуру:

- заголовочную часть `<Titles/>`
- описательную часть `<Descriptions/>`
- список с контрактами `<ActionContracts/>`
- иконку в формате SVG `<IconSvg/>`

`<ActionContracts/>` содержит список контрактов с реализациями действий - `<ActionContract/>`

Если разработчик создал еще одну версию созданного ранее действия, то ему необходимо в файл `ListActions.xml` добавить новый XML-элемент `<ActionContract/>` следующего содержания:

```
<ActionContract contract="контракт" impVersion="версия контракта" superActionVersion="версия супер-действия"
exехType="платформа" />
```

#### Пример ListActions.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ListSuperActions
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:noNamespaceSchemaLocation="http://packages.itbs.it.ru:8083/repository/raw-robin-
artifacts/schemes/ListActions/1.2.0/ListActions.xsd"
  schemaVersion="1.0.0"
  version="1.2.0"
>
  <SuperAction actionID="Const.GetPass">
    <Titles>
      <Title lang="ru-RU">
        <content>Получить пароль</content>
      </Title>
      <Title lang="en-US">
        <content>Get pass</content>
      </Title>
    </Titles>
    <Descriptions>
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        <Description lang="ru-RU">
            <content>Получить пароль</content>
        </Description>
        <Description lang="en-US">
            <content>Get pass</content>
        </Description>
    </Descriptions>
    <ActionContracts>
        <ActionContract contract="000Test.Const.GetPass" impVersion="1.3.0"
↪superActionVersion="1" execType="net" />
    </ActionContracts>
    <IconSvg>
        <svg width="54" height="29" viewBox="0 0 54 29" fill="none" xmlns="http://www.w3.
↪org/2000/svg">
            <path fill-rule="evenodd" clip-rule="evenodd" d="M1.42595 28.3864L1.23442 28.
↪8482C1.47024 28.3864 1.47024 28.3864 1.47024 28.3864" />
        </svg>
    </IconSvg>
</SuperAction>
</ListSuperActions>

```

### 7.3.8 ListGroups.xml

Если было создано новое супер-действие, то его необходимо добавить в `ListGroups.xml` в подгруппу или группу супер-действий.

Намерения пользователя	Действия пользователя
Добавление супер-действия в существующую подгруппу существующей группы	Сначала <code>ListGroups.xml</code> выгружается из кэша студии с помощью утилиты импорта, выгруженный файл включает в себя только структуру групп и подгрупп без наполнения. После этого пользователь добавляет супер-действие туда, куда ему нужно
Добавление супер-действия в существующую группу	
Добавление супер-действия в новую подгруппу существующей группы	
Добавление супер-действия в новую подгруппу новой группы	У пользователя есть пример <code>ListGroups.xml</code> , поставляемый в качестве примера составе SDK. После этого пользователь добавляет туда новое супер-действие

`ListGroups.xml` содержит корневой XML-элемент `<ListGroups>`, внутри которого располагаются один или несколько XML-элементов `<Group>`.

Разработчик для своего нового действия должен создать новый элемент `<Group>` или воспользоваться имеющимся, если он есть.

`<Group>` имеет следующую структуру:

- заголовочную часть `<Titles/>`
- описательную часть `<Descriptions/>`
- иконку в формате SVG `<IconSvg>`
- список с контрактами `<Actions/>`

<Actions/> содержит список супер-действий - <Action/>

### Пример ListGroups.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ListGroups
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:noNamespaceSchemaLocation="http://packages.itbs.it.ru:8083/repository/raw-robin-
  ↪artifacts/schemes/ListGroups/1.1.0/ListGroups.xsd"
  schemaVersion="1.1.0"
  version="1.0.0"
>
  <Group name="Const">
    <Titles>
      <Title lang="ru-RU">
        <content>Константы</content>
      </Title>
      <Title lang="en-US">
        <content>Const</content>
      </Title>
    </Titles>
    <Descriptions>
      <Description lang="ru-RU">
        <content>Группа действий "Константы"</content>
      </Description>
      <Description lang="en-US">
        <content>"Const" actions group</content>
      </Description>
    </Descriptions>
    <IconSvg>
      <svg width="17" height="21" viewBox="0 0 17 21" fill="none" xmlns="http://www.w3.
      ↪org/2000/svg">
        <path d="M8.5 0L0 3.77778V9.44444C0 14.6861 3.62667 19.5878 8.5 20.7778C13.
        ↪3733 19.5878 17 14.6861 17 9.44444V3.77778L8.5 0ZM8.5 10.3794H15.1111C14.6106 14.2706 12.
        ↪0133 17.7367 8.5 18.8228V10.3889H1.88889V5.00556L8.5 2.06833V10.3794Z" fill="#3B4C77"/>
      </svg>
    </IconSvg>
    <Actions>
      <Action actionID="Const.GetPass" />
    </Actions>
  </Group>
</ListGroups>
```

### 7.3.9 Создать дистрибутив с действием

Теперь можно создать дистрибутив с действием. Для этого необходимо запустить Robin.ActionEditor и выполнить следующие действия:

- открыть Проект действия с пользовательским действием
- нажать на «Файл/Сохранить как пакет»

В результате этой операции в корне Проекта действия появится пакет с дистрибутивом - 000Test.Const.GetPass-1.0.0.robin-package

### 7.3.10 Импорт действия в ROBIN Studio

Для импорта действия в Робин Студию необходимо:

1. Запустить утилиту импорта



2. Выбираем подготовленный пакет для импорта - \*.robin-package
3. Нажимаем кнопку «импортировать»
4. Нажимаем кнопку «обновить кэш»
5. Запускаем Робин студию и находим добавленное действие
6. Используем действие

## 7.4 Создание .Net-действия

### Установка ПО

Для создания .net-действий на компьютере разработчика должно быть установлено следующее ПО:

- .Net framework 4.8 SDK и .NET Core 2.0 SDK и выше - <https://dotnet.microsoft.com/download/visual-studio-sdks>
- Visual Studio 2019 - <https://visualstudio.microsoft.com/ru/>

#### 7.4.1 Создание проекта действия

Перед созданием проекта требуется установить пакет с шаблоном действия, для этого нужно в командной строке выполнить

```
dotnet new -i <%AppData%\Local\Programs\Robin Platform\Robin SDK\2.0\TemplatePack\Internal.  
↪Build.ActionTemplate.1.0.1.nupkg>
```

После этого в списке шаблонов появится новый шаблон:

ASP.NET Core with React.j...	reactredux	[C#]	Web/MVC/SPA
Razor Class Library	razorclasslib	[C#]	Web/Razor/Library
ASP.NET Core Web API	webapi	[C#], F#	Web/WebAPI
ASP.NET Core gRPC Service	grpc	[C#]	Web/gRPC
ActionTemplate	action		Common/Library
dotnet gitignore file	gitignore		Config
global.json file	globaljson		Config
NuGet Config	nugetconfig		Config
Dotnet local tool manifes...	tool-manifest		Config
Web Config	webconfig		Config
Solution File	sln		Solution
Protocol Buffer File	proto		Web/gRPC

Для создания проекта нужно выполнить команду:

```
dotnet new action --force -o <путь, по которому будет создан проект> --company <имя компании> -  
↪-group <имя группы, к которой принадлежит действие> --action <название действия>
```

Все параметры должны быть написаны латинскими буквами и символами (не позволительны русские буквы и пробелы) и будут использованы для создания контракта в формате `company.group.action.version`. Также есть опциональный параметр `--actionVersion <версия действия>`, значение по умолчанию: 1.3.0.

После выполнения в указанной директории должны появиться следующие файлы:

- Action.sln - решение, содержащее проект с действием
- Action.csproj - проект действия
- Action.cs - исходный код действия
- Company.Group.Action-net.robin-impinfo - информация о реализации

Указанные при создании проекта параметры будут использоваться для генерации содержимого в файлах.

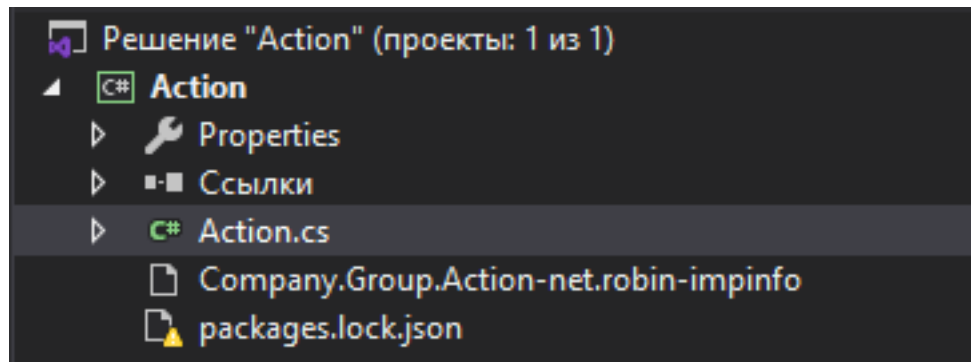
### 7.4.2 Описание проекта

Файл `Company.Group.Action-net.robin-impinfo` содержит информацию о реализации: название, издателя и версию. Ниже приведен пример созданного файла, сгенерированного с использованием в указанных в примере параметрах

```
<?xml version="1.0" encoding="UTF-8"?>
<ImplementationInfo version="$version$" contract="$company$.$group$.$action$" execType="net"
  schemaVersion="1.0.8">
  <Author>$company$</Author>
  <RequireUiThread>true</RequireUiThread>
  <RequiredInterfaces>
    <RequiredInterface>Robin.Engine.Services.Interfaces.IActionLogger</RequiredInterface>
  </RequiredInterfaces>
  <RequireUiThread>>false</RequireUiThread>
  <AssemblyName>$company$-Imp.$group$.$action$, Version=$version$.0, Culture=neutral,
  PublicKeyToken=null</AssemblyName>
  <PackageName>$company$-Imp.$group$.$action$</PackageName>
</ImplementationInfo>
```

- `version` - версия действия, ее необходимо поднять при создании новой версии.
- `execType` - платформа, на которой написана реализация
- `schemaVersion` - версия схемы с определением `impinfo`, сейчас используется 1.0.8
- `Author` - издатель действия

Решение `Action.sln` содержит следующую структуру файлов



Перед редактированием кода необходимо восстановить пакеты **Nuget**, используемые в проекте. Для этого, нажмите правой кнопкой мыши по решению и выберите «Восстановить пакеты **Nuget**». Если пакеты были восстановлены успешно, то сборка проекта завершится без ошибок. Подробнее о пакетах указано ниже.

### 7.4.3 Состав пакетов SDK

В состав Robin SDK входят **nuget**-пакеты, содержащие интерфейсы и классы, необходимые для написания действий. Ниже приведено краткое описание пакетов и их содержимого:

Наименование пакета	Версия пакета	Описание пакета
nuget.build.tasks.pack	5.3.1	Необходим для создания <b>nuget</b> пакетов с реализацией пользовательского действия. Должен быть использован во всех действиях.
robin.sdk.actions	1.4.0	Основной пакет для действий, в нем предоставлены: абстрактный класс <code>BaseRobinAction</code> , от которого должны наследоваться реализации действий и интерфейсы сервисов.

Для групп действий, где несколько действий работают с общими объектами, используются **nuget** пакеты с определением контекстов для их дальнейшей передачи действиям. Такие пакеты имеют наименование `robin.%group%.commonclasses`. При разработке действий, работающих с контекстами издателя, необходимо добавить в проект соответствующий пакет `CommonClasses`. Список пакетов указан ниже:

- `robin.xml.commonclasses 1.0.0`
- `robin.web.commonclasses 2.0.0`
- `robin.ftp.commonclasses 2.0.5`
- `robin.pdf.commonclasses 1.0.1`
- `robin.word.commonclasses 2.0.2`

Если тип, используемый в действии, не является нативным типом языка разработки, то необходимо включить в проект пакет с определением типа. Более подробно можно узнать в Приложении 2. Список пакетов, содержащих определение типов, указан ниже:

- `robin.type.filepath 2.0.0`
- `robin.type.folderpath 2.0.0`
- `robin.type.hotkeys 2.0.0`
- `robin.type.image 2.0.0`
- `robin.type.webelement 2.0.0`
- `robin.type.winelement 2.0.0`
- `robin.type.password 2.0.0`

#### 7.4.4 Написание реализации

Исходный код реализации расположен в файле `Action.cs`. Ниже приведен код, полученный после генерации проекта

```
using Robin.ActionSDK;
using Robin.Engine.Services.Interfaces;
using System.Collections.Generic;
using System;

namespace 000.Test
{
    public class Test : BaseRobinAction
    {
        public Test(IActionLogger logger) : base(logger) { }
        public override IDictionary<string, object> Execute(IDictionary<string, object>
↵parameters)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    {  
        return null;  
    }  
}
```

Класс `BaseRobinAction` - абстрактный класс для действия, от которого необходимо наследовать свое действие при разработке, находящийся в пространстве имен `Robin.ActionSDK`.

Интерфейс `IActionLogger` содержит интерфейс для работы с классами логирования, интерфейс находится в пространстве имен `Robin.Engine.Services.Interfaces`.

В конструкторе текущего класса необходимо указать все необходимые для работы сервисы, например, для работы с ресурсами необходимо указать в аргументах `IResourcesFolderService`.

При выполнении действия роботом вызывается метод `Execute`, на вход которому передается `Dictionary` с параметрами, указанными в действии, результаты действия возвращаются как результат выполнения метода `Execute`, при его отсутствии возвращается `null`, если действие должно возвращать результат, то необходимо написать следующий фрагмент кода.

```
IDictionary<string, object> result = new Dictionary<string, object>  
{  
    {"Result", resultValue}  
};  
return result;
```

#### 7.4.5 Добавление логирования

В конструкторе класса действия на вход был получен `IActionLogger logger`. Мы можем обратиться к сервису логирования через наследуемый атрибут `Logger`, сервис логирования имеет несколько уровней логирования:

- Debug
- Info
- Warning
- Error

При указании в студии у действия нужного уровня логирования, мы получим при выполнении логи от соответствующего уровня. Для создания логов в исходном коде необходимо вызвать

```
Logger.<Level>(<message>);
```

Например:

```
Logger.Error("Возникла ошибка при работе действия");  
Logger.Debug("Первый этап действия завершился успешно");
```

#### 7.4.6 Сборка проекта и создание Nuget пакета

При помощи утилиты `nuget` соберите пакет действия.

## 7.5 Приложение 1. Схема имплементации действия ActionImpScheme\_ActionImp.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.5.0">
  <xs:element name="ImplementationInfo">
    <xs:annotation>
      <xs:documentation>Корневой элемент описания реализации действия</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RequiredInterfaces">
          <xs:annotation>
            <xs:documentation>Обязательный тег. Должен быть даже если ни одного
↪требуемого интерфейса нет. (дабы отловить файлы в которые забыли это добавить)</
↪xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="RequiredInterface" type="xs:string" minOccurs="0
↪" maxOccurs="unbounded">
                <xs:annotation>
                  <xs:documentation>Полное название (включая неймспейс)
↪интерфейса, который требуется действию для его работы. (действие его запрашивает, помещая в
↪свой конструктор)</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="DeprecatedVersions" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Список запрещенных версий реализаций данного действия
↪</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DeprecatedVersion" type="xs:string" maxOccurs=
↪"unbounded">
                <xs:annotation>
                  <xs:documentation>Запрещенная версия реализации действия</
↪xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Author" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Автор последних изменений</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Build" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Номер билда</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="DateBuild" type="xs:dateTime" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Дата и время сборки действия</xs:documentation>
          </xs:annotation>

```

(continues on next page)

(продолжение с предыдущей страницы)

```

</xs:element>
<xs:element name="RequireUiThread" type="xs:boolean" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Флаг говорящий о том, что данному действию необходим
↪ для работы графический контекст</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="AssemblyName" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Для .NET - полное имя сборки</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="PackageName" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Имя пакета содержащего реализацию действия</
↪ xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="CommitHash" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Хеш комита из которого собрано действие</
↪ xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Exceptions" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Список исключений, которое может генерировать данное
↪ действие</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ExceptionInfo" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute name="TypeID" type="xs:string" use="required"
↪ ">
            <xs:annotation>
              <xs:documentation>Идентификатор типа исключения</
↪ xs:documentation>
            </xs:annotation>
          </xs:attribute>
          <xs:attribute ref="version" use="required">
            <xs:annotation>
              <xs:documentation>Версия описания типа исключения</
↪ xs:documentation>
            </xs:annotation>
          </xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ListPermittedOS" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PermittedOS" type="TypePermittedOS" maxOccurs=
↪ "unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute ref="version" use="required"/>

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        <xs:attribute name="contract" type="xs:string" use="required"/>
        <xs:attribute name="schemaVersion" type="xs:string" use="required"/>
        <xs:attribute name="execType" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
    <xs:attribute name="version" type="xs:string">
        <xs:annotation>
            <xs:documentation>Версия чего либо</xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:complexType name="TypePermittedOS">
        <xs:annotation>
            <xs:documentation>Тип представляющий информацию о релизах ОС</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="OSDistributorID" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Название дистрибутива</xs:documentation>
                    <xs:documentation>Позвоительно указывать перечисление дистрибутивов ОС
↪ через запятую (для линукса справедливо) Позвоительно вообще не указывать данный атрибут и
↪ это значить любой дистрибутив ОС</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="OSRelease" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Номер релиза</xs:documentation>
                    <xs:documentation>Позвоительно префиксом указывать ">=" что означает от
↪ данной версии и выше. Позвоительно указывать перечисление версий ОС через запятую.
↪ Позвоительно вообще не указывать данный атрибут и это значить любая версия ОС</
↪ xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="OSType" default="64" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>разрядность ОС</xs:documentation>
                </xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:enumeration value="32"/>
                        <xs:enumeration value="64"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

## 7.6 Приложение 2. Типы в SDK Robin

### 7.6.1 Тип Robin.Boolean

Сериализация типа: Да

Нативный тип .NET: System.Boolean

Презентационное значение объекта: Получается с помощью toString()

Пример: true

### 7.6.2 Тип Robin.Collection

Сериализация типа: Да

Нативный тип .NET: `List<object>`

Презентационное значение объекта: Количество записей в коллекции. Получается с помощью `toString()`

Пример: Записей = 5 - выдает число элементов (записей) в коллекции.

### 7.6.3 Тип Robin.DataTable

Сериализация типа: Да

Нативный тип .NET: `System.Data.DataTable`

Презентационное значение объекта: Получается с помощью `toString()`

Пример: Столбцов = 45, Строк = 74 - выдает число столбцов и строк в таблице.

### 7.6.4 Тип Robin.DateTime

Сериализация типа: Да

Нативный тип .NET: `System.DateTime`

Презентационное значение объекта: Получается с помощью `toString()`

Пример: Дата и время в формате dd-MM-yyuu HH:mm:ssXXX: 22-07-2020 10:17:34 +03:00

### 7.6.5 Тип Robin.Dictionary

Сериализация типа: Да

Нативный тип .NET: `System.Collections.Generic.Dictionary<string, object>`

Презентационное значение объекта: Получается с помощью `toString()`

Пример: Записей = 68 - выдает число элементов (записей) в словаре.

### 7.6.6 Тип Robin.Exception

Сериализация типа: Нет

DTO-тип .NET: `Robin.SDK.Types.Interfaces.RobinException` из Nuget-пакета `Robin.SDK.Actions.1.3.0`

Презентационное значение объекта: Получается с помощью `toString()`

Пример: Поля структуры ошибки, выведенные каждое на новой строке:

- Тип: ...
- Сообщение: ...
- Источник: ...
- Дополнительная информация: ключ=значение, ключ=значение
- Вложенное исключение: есть/нет



### 7.6.7 Тип Robin.FilePath

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinFilePath из Nuget-пакета Robin.Type.FilePath.2.0.0

Презентационное значение объекта: Получается с помощью toString()

Пример: C://Temp/1.txt - строка внутри обёртки FilePath

### 7.6.8 Тип Robin.FolderPath

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinFolderPath из Nuget-пакета Robin.Type.RobinFolderPath.2.0.0

Презентационное значение объекта: Получается с помощью toString()

Пример: C://Temp/1.txt - строка внутри обёртки FolderPath

### 7.6.9 Тип Robin.HotKeys

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinHotKeys из Nuget-пакета Robin.Type.RobinHotKeys.2.0.0

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.10 Тип Robin.Image

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinImage из Nuget-пакета Robin.Type.RobinImage.2.0.0

Презентационное значение объекта: Получается с помощью toString()

Пример: C://Temp/1.png - строка внутри обёртки Image

### 7.6.11 Тип Robin.LocationPoint

Сериализация типа: Да

Нативный тип .NET: System.Drawing.Point

Презентационное значение объекта: Получается с помощью toString()

Пример: (24, 32) - координаты точки в формате (x, y)

### 7.6.12 Тип Robin.Numeric

Сериализация типа: Да

Нативный тип .NET: double

Презентационное значение объекта: Получается с помощью toString()

Пример: 28.5

### 7.6.13 Тип Robin.Object

Сериализация типа: Нет

Нативный тип .NET: object

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.14 Тип Robin.Password

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinPassword из Nuget-пакета Robin.Type.RobinPassword.2.0.0

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.15 Тип Robin.RectangleSize

Сериализация типа: Да

Нативный тип .NET: System.Drawing.Size

Презентационное значение объекта: Получается с помощью toString()

Пример: 240 x 320 - размер прямоугольной области в формате W x H

### 7.6.16 Тип Robin.Selenium

Сериализация типа: Нет

Нативный тип .NET: object

Презентационное значение объекта: Отсутствует

Пример: Экземпляр Web-драйвера

### 7.6.17 Тип Robin.String

Сериализация типа: Да

Нативный тип .NET: string

Презентационное значение объекта: Сама строка и является презентационным значением объекта

Пример: Строка объекта

### 7.6.18 Тип Robin.WebElement

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinWebElement из Nuget-пакета Robin.Type.RobinWebElement.2.0.0

Презентационное значение объекта: Выводится одно из полей структуры. Какое поле вывести указано в поле activeAttribute этой же структуры.

Пример:

### 7.6.19 Тип Robin.WinElement

Сериализация типа: Да

DTO-тип .NET: Robin.Type.RobinWinElement из Nuget-пакета Robin.Type.RobinWinElement.2.0.0

Презентационное значение объекта:

Пример:

### 7.6.20 Тип Robin.XmlDoc

Сериализация типа: Да

Нативный тип .NET: System.Xml.XmlDocument

Презентационное значение объекта: XML представление xml документа(при презентации пользователю в виде одной строки - обрезается (программой презентующей), но есть возможность посмотреть полный текст.

Пример:

### 7.6.21 Тип Robin.Email

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.22 Тип Robin.FTP

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.23 Тип Robin.MQ

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.24 Тип Robin.PDF

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.25 Тип Robin.SpreadSheets

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Отсутствует

Пример:

### 7.6.26 Тип Robin.Word

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Экземпляр Word

Пример:

### 7.6.27 Тип Robin.XML

Сериализация типа: Нет

Нативный тип .NET: нет соответствия

Презентационное значение объекта: Отсутствует

Пример:

---

## Скрипты по автоматическому развертыванию RMS. Руководство администратора

---

### 8.1 Аннотация

Инструкция описывает назначение скриптов автоматической установки оркестратора и других серверных компонент Платформы Робин в редакции standalone, а также действия системного администратора по их настройке и запуску.

### 8.2 Назначение

Скрипты:

- выполняют установку и конфигурирование следующего серверного ПО:
  - PostgreSQL 10+ - система управления базами данных,
  - ELK-сервис (Elasticsearch, Logstash и Kibana) – сервис, предназначенный для сбора и хранения логов от прикладных сервисов Платформы Робин, а также для выполнения клиентских запросов с целью построения отчетов.
  - FreeIPA - интегрированное решение по идентификации и аутентификации пользователей, задания политик доступа и аудита для сетей на базе Linux и Unix,
  - Sonatype Nexus – интегрированная платформа, с помощью которой разработчики могут проксировать, хранить и управлять зависимостями Java (Maven), образами Docker, Python, Ruby, NPM, Bower, RPM-пакетами, gitlfs, Apt, Go, Nuget, а также распространять свое программное обеспечение,
  - Rabbit MQ – программный брокер сообщений на основе стандарта AMQP, обеспечивает асинхронную передачу сообщений между прикладными службами Робин Платформы.
  - WildFly (JBoss Application Server) – сервер приложений Java EE,
- выгружают серверные компоненты ( j2ee-приложения) RMS на сервер WildFly :
  - rms2.war
  - rpm.war
- загружают схему базы данных RMS,

- загружают схему базы FreeIPA.

## 8.3 Состав и описание функциональных возможностей ROBIN Platform 2.0

Платформа Робин 2.0 представляет собой комплекс программных средств, предназначенных для решения задач в области роботизации бизнес-процессов, а именно:

- осуществлять разработку новых программных роботов и действий,
- выполнять сценарии программных роботов.

В состав Платформы Робин входят следующие программные компоненты:

- на стороне клиента:
  - ROBIN Management Console (RMC) – клиентское приложение, предоставляющее пользователю графический интерфейс для централизованного управления запусками роботов на удаленных хостах,
  - ROBIN Studio – клиентское приложение, предоставляющее пользователю графический интерфейс для разработки программных роботов,
  - ROBIN Player – клиентское приложение, предоставляющее пользователю графический интерфейс для управления запусками программных роботов на исполнение сценария,
  - ROBIN Executor – прикладной сервис, расположенный на стороне клиента (ROBIN Studio или ROBIN Player), система исполнения сценариев программных роботов,
  - ROBIN Agent – прикладной сервис, расположенный на стороне клиента (ROBIN Studio или ROBIN Player), выполняет функцию проксирования клиентских запросов другим прикладным сервисам RMS или RMC через очереди программного брокера сообщений,
- на стороне серверной части приложений:
  - ROBIN Management Server (RMS) – централизованная служба, выступающая в роли оркестратора, выполняет функцию делегирования клиентских и сервисных запросов прикладным сервисам Робин Платформы,
  - База данных RMS - система хранения мета-информации о роботах и пользователях с использованием СУБД PostgreSQL 10+,
  - Хранилище роботов – файловое хранилище (на базе Sonatype Nexus), предназначенное для хранения артефактов роботов и действий:
    - \* пакеты с исходниками роботов,
    - \* пакеты с исполняемым кодом и документацией роботов,
    - \* зависимости (библиотеки), необходимые для выполнения исполняемого кода программных роботов
  - Сервер идентификации пользователей - корпоративное хранилище сведений о пользователях/компаниях, роботах, действиях и правах доступа пользователей/компаний к роботам и действиям на базе FreeIPA-сервера.
  - Robin RDP-Manager – сервис, предназначенный для открытия/закрытия RDP-сессии на удаленном хосте для запуска локальных прикладных служб клиентских приложений,
  - Robin Package Manager – сервис, выполняет функцию проксирования клиентских и сервисных запросов к хранилищу роботов,

Работу серверных и клиентских компонент ROBIN Platform обеспечивают следующие инфраструктурные сервисы (сервера приложений):

- WildFly (JBoss Application Server) – сервер приложений Java EE,

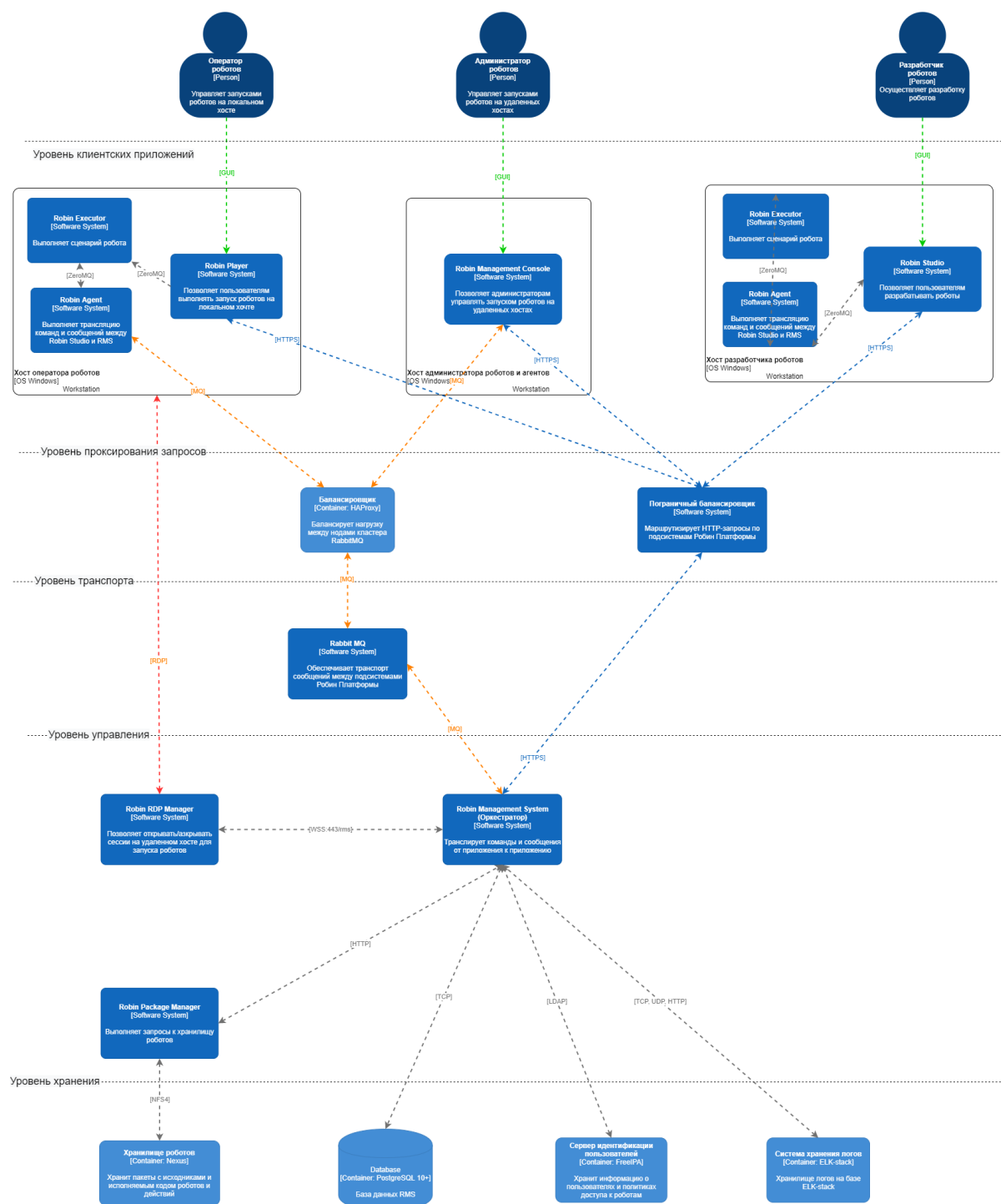


Рис. 1: Функциональная архитектура Робин Платформы.

- PostgreSQL 10+ - система управления базами данных,
- FreeIPA - интегрированное решение по идентификации и аутентификации пользователей, задания политик доступа и аудита для сетей на базе Linux и Unix,
- Sonatype Nexus – интегрированная платформа, с помощью которой разработчики могут проксировать, хранить и управлять зависимостями Java (Maven), образами Docker, Python, Ruby, NPM, Bower, RPM-пакетами, gitlfs, Apt, Go, Nuget, а также распространять свое программное обеспечение,
- Rabbit MQ – программный брокер сообщений на основе стандарта AMQP, обеспечивает асинхронную передачу сообщений между прикладными службами Робин Платформы.
- HAProxy - серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов,
- ELK-сервис (Elasticsearch, Logstash и Kibana) – сервис, предназначенный для сбора и хранения логов от прикладных сервисов Платформы Робин, а также для выполнения клиентских запросов с целью построения отчетов.

Хребет функциональной архитектуры Платформы Робин реализован через 2 архитектурных шаблона проектирования - шаблон хореографии и шаблон оркестрации.

В рамках *шаблона хореографии* взаимодействуют следующие компоненты Платформы Робин:

- Robin-агенты,
- RMC - консоль управления роботами,
- RMS - сервер управления.

Поддержку транспорта сообщений и механизма очередей сообщений между компонентами выполняет RabbitMQ по протоколу AMPQ.

*Шаблон оркестрации* реализован между клиентскими приложениями и серверными компонентами через Robin Management Server (RMS) – централизованную службу, которая выполняет функцию делегирования клиентских и сервисных запросов прикладным сервисам Робин Платформы по протоколу HTTPS. RMS обрабатывает запросы от:

- RMC
- Robin Studio
- Robin Player

Сам RMS состоит из следующих J2EE-приложений:

- rms2.war - выполняет аутентификацию клиентов, открывает и закрывает RDP-сессии на агентских машинах,
- rpm.war - сохраняет пакеты роботов и их исходники в хранилище роботов, добавляет записи о них в каталог FreeIPA.

## 8.4 Состав и назначение скриптов

### 8.4.1 Архитектура

Для автоматической установки RMS используется Ansible. Ansible — система управления конфигурациями, написанная на языке программирования Python, с использованием декларативного языка разметки для описания конфигураций. Используется для автоматизации настройки и развертывания программного обеспечения.

Скрипты представляют собой набор конфигурационных файлов и плейбуков для системы Ansible. В конфигурационных файлах (\*.cfg, \*.ini, group\_vars/\*, templates/\*) задаются параметры настроек скриптов. Сами скрипты представляют собой набор плейбуков (сценариев) для функциональных



модулей Ansible. Каждый плейбук содержит одно или несколько заданий для решения одной или нескольких взаимосвязанных задач по установке и настройке ресурса. Задание имеет имя задания, имя Ansible-модуля, который должен исполнять задание, параметры задания. Параметры задания задают в том числе и целевое состояние настраиваемого ресурса.

### 8.4.2 Конфигурационные файлы (параметры настройки сервисов)

Таблица 1. Конфигурационные файлы плейбуков.

Имя файла	Назначение	Размещение
ansible.cfg	Задаёт локальные настройки Ansible	./ - корневой каталог скриптов
hosts.ini	Задаёт целевые хосты, на которых будут устанавливаться приложения. Хосты могут быть объединены в группы.	./ - корневой каталог скриптов
all.yaml	Задаёт параметры (пароли и ip) доступа к БД пользователей и служб Робин.	./group_vars/
app.yaml	Задаёт параметры WildFly (версии, пароль, ip, пути к исполняемым файлам)	./group_vars/
db.yaml	Задаёт версию FlyWay	./group_vars/
ldap.yaml	Задаёт локальный домен FreeIPA	./group_vars/
pkg.yaml	Задаёт версию Nexus	./group_vars/
rabbit.yaml	Задаёт пароли доступа к системе сообщений Rabbit	./group_vars/
elastic-search-single.yml.j2	Jinja-шаблон для elastic.yaml (установка Elasticsearch без кластера)	./templates/
elasticsearch.yml.j2	Jinja-шаблон для elastic.yaml (установка Elasticsearch в кластере)	./templates/
kibana.yml.j2	Jinja-шаблон для kibana.yaml	./templates/
logstash.conf.j2	Jinja-шаблон для logstash.yaml	./templates/
nexus.service	Jinja-шаблон для nexus.yaml (команды запуска Nexus-сервиса)	./templates/
orchestrator-client.cer	Сертификат безопасности оркестратора с открытым ключом для шифрования данных при обращении клиента к серверу	./templates/
orchestrator-client.jks	Java-хранилище с закрытым ключом оркестратора используется для расшифровки данных, отправленных клиентом.	./templates/
rms2-infinispan.xml	Jinja-шаблон с настройками параметрами распределенного кэша infinispan сервиса rms2	./templates/
rms2.properties	Jinja-шаблон с настройками параметрами сервиса rms2	./templates/
rpm-infinispan.xml	Jinja-шаблон с настройками параметрами распределенного кэша infinispan сервиса rpm для wf-standalone.yaml и wildfly.yaml	./templates/
rpm.properties	Jinja-шаблон с настройками параметрами сервиса rpm для wf-standalone.yaml и wildfly.yaml	./templates/

### 8.4.3 Конфигурационные скрипты (плейбуки)

#### db.yaml

Скрипт db.yaml:

- выполняет установку пользовательского репозитория пакетов rpm для ОС RedHat 7,
- выполняет установку пакетов для СУБД PostgreSQL 11:

- python-psycopg2
- postgresql11
- postgresql11-server
- postgresql11-contrib
- инициализирует базу данных,
- устанавливает сервис postgresql-11 в автозапуск
- устанавливает пароль для пользователя postgres,
- задает сетевые настройки для клиентов СУБД PostgreSQL,
- создает пользователя user\_robin, его пароль и права,
- создает БД robin\_2,
- создает пользователей user\_orch, user\_quartz, user\_rpm, их пароли и роли,
- выполняет команды запуска сервиса или перезапуска,
- выполняет проверку работоспособности сервиса с помощью пользовательских SQL-запросов,
- создает схему базы данных robin\_2:
  - выполняет распаковку архива flyway-commandline-{{ flyway\_ver }}-linux-x64.tar.gz
  - устанавливает flyway,
  - распаковывает архив с sql-скриптами для flyway,
  - запускает задание на исполнение скриптов для создания схемы базы данных robin\_2.

### elastic.yaml

Скрипт elastic.yaml:

- выполняет установку пользовательского репозитория пакетов ELK для ОС RedHat 7,
- устанавливает пакеты:
  - java-1.8.0-openjdk
  - elasticsearch
  - jq
- устанавливает настройки JVM (jvm.options)
- устанавливает лимит виртуальной памяти (vm.max\_map\_count)
- подгружает один из Jinja-шаблонов - elasticsearch-single.yml.j2 или elasticsearch.yml.j2 в зависимости от параметра elastic\_cluster
- устанавливает сервис elastic в автозапуск

### freeipa.yaml

Скрипт freeipa.yaml:

- задает имя хоста ldap1 для FreeIPA
- отключает ipv6
- устанавливает пакеты:
  - ipa-server
  - ipa-server-trust-ad

- ipa-server-dns
- bind
- bind-dyndb-ldap
- создает директорию /opt/ldif
- извлекает в нее схемы для сервера идентификации из архива ldif.zip
- импортирует схемы в БД FreeIPA
- устанавливает парольную политику для пользователей FreeIPA.

### kibana.yaml

Скрипт kibana.yaml:

- выполняет установку пользовательского репозитория пакетов ELK для ОС RedHat 7,
- устанавливает пакеты:
  - kibana
  - jq
- создает системного пользователя kibana без домашней директории
- создает директорию /opt/kibana
- создает директорию /var/run/kibana для логов
- копирует настройки из Jinja-шаблона kibana.yml.j2 в /etc/kibana/kibana.yml
- устанавливает сервис kibana в автозапуск

### logstash.yaml

Скрипт logstash.yaml:

- выполняет установку пользовательского репозитория пакетов ELK для ОС RedHat 7,
- устанавливает пакет logstash,
- копирует настройки из Jinja-шаблона logstash.yml.j2 в /etc/logstash/conf.d/base.conf
- устанавливает сервис logstash в автозапуск

### nexus.yaml

Скрипт nexus.yaml:

- создает системного пользователя nexus,
- выполняет распаковку архива nexus-{{ nexus\_ver }}-unix.tar.gz
- меняет имя директории, а которую был распакован архив
- проверяет существование директории /opt/sonatype-work
- рекурсивно меняет собственника директории /opt/nexus и всех ее поддиректорий
- рекурсивно меняет собственника директории /opt/sonatype-work и всех ее поддиректорий
- модифицирует файл nexus.rc для запуска сервиса от имени пользователя nexus
- копирует Jinja-шаблон nexus.service в директорию /etc/systemd/system и меняет собственника на root
- устанавливает сервис nexus в автозапуск

**rabbit.yaml**

Скрипт rabbit.yaml:

- выполняет установку пользовательских репозиторий rabbitmq\_erlang и rabbitmq\_server для ОС RedHat 7,
- устанавливает пакеты:
  - python-requests
  - erlang
  - rabbitmq-server
- создает директорию /etc/systemd/system/rabbitmq-server.service.d
- создает конфигурационный файл /etc/systemd/system/rabbitmq-server.service.d/limits.conf, в котором задает параметр LimitNOFILE
- устанавливает сервис rabbitmq-server.service в автозапуск
- подключает плагины rabbitmq\_management, rabbitmq\_event\_exchange
- создает пользователя admin, задает для него пароль и дает полные права на конфигурирование, чтение и запись на хосте /,
- создает пользователей agent, rmc, rms, rdp, задает для них пароли и дает полные права на конфигурирование, чтение и запись на хосте /,
- создает файл .erlang.cookie в /var/lib/rabbitmq,
- перезапускает сервис rabbitmq-server.service,
- задает параметры политики для rabbitmq,
- задает обменники и их типы для управления очередями:
  - agents.exchange - тип direct,
  - rdp.exchange - тип fanout,
  - rmc.exchange - тип topic
- создает очереди:
  - mirror.agents.queue
  - mirror.rdp.queue
  - mirror.events.queue
- задает правила маршрутизации сообщений для обменников.

**wf-standalone.yaml**

Скрипт wf-standalone.yaml:

- устанавливает пакеты:
  - java-1.8.0-openjdk
  - unzip
- создает системного пользователя wildfly без домашней директории и без входа в систему
- устанавливает переменную окружения wildfly\_home для домашней директории /opt/wildfly
- устанавливает значения параметров системы:
  - net.core.rmem\_max - максимальный размер буфера получения данных через сокетные соединения в байтах

- net.core.wmem\_max - максимальный размер буфера отправки данных сокетные соединения в байтах
- распаковывает архив wildfly-{{ wf\_version }}.Final.tar.gz в директорию /opt,
- создает директорию /opt/wildfly-{{ wf\_version }}.Final, если не создана, и устанавливает ее собственника owner:wildfly и group:wildfly,
- создает символическую ссылку /opt/wildfly на директорию /opt/wildfly-{{ wf\_version }}.Final,
- создает директории:
  - /opt/wildfly/modules
  - /opt/wildfly/certificate/server
  - /opt/wildfly/certificate/client
 и устанавливает права доступа к ним,
- устанавливает postgresql драйвер для java-приложений оркестратора в директорию /opt/wildfly/modules/rms2/main,
- копирует содержимое ресурсных файлов rms2.properties и rpm.properties в файл /opt/wildfly/modules/rms2/main/properties
- копирует настройки infinispn из rms2-infinispn.xml и rpm-infinispn.xml в файл /opt/wildfly/modules/rms2/main/properties
- копирует j2ee-приложения rms2.war и rpm.war в /opt
- копирует сертификат безопасности оркестратора orchestrator-client.cer в директорию /opt/wildfly/certificate/client/
- копирует хранилище с закрытым ключом оркестратора orchestrator-client.jks в директорию /opt/wildfly/certificate/server/
- модифицирует mgmt-groups.properties и mgmt-users.properties
- устанавливает значения параметров в файле /opt/wildfly/standalone/configuration/standalone.xml:
  - jboss.bind.address.management
  - jboss.bind.address
  - jboss.socket.binding.port-offset
- устанавливает настройки для логирования сервера в файле /opt/wildfly/standalone/configuration/standalone.xml,
- рекурсивно меняет собственника директории /opt/wildfly и всех ее поддиректорий на owner:wildfly и group:wildfly
- создает директорию /etc/wildfly и устанавливает собственника owner:wildfly и group:wildfly
- создает wildfly.conf в /etc/wildfly и добавляет в него настройки
- копирует скрипт запуска сервера launch.sh в директорию /opt/wildfly/bin
- копирует wildfly.service в /usr/lib/systemd/system
- задает параметры запуска сервера JAVA\_OPTS для JVM
- создает файл /var/run/wildfly и устанавливает собственника owner:wildfly и group:wildfly
- устанавливает сервис wildfly.service в автозапуск
- перезапускает сервис wildfly.service
- выполняет команды:
  - по настройке базы данных (драйвера и источники данных)
  - указывает URL доступа к сертификатам

- выполняет настройку слушателей подключений по протоколам HTTP/HTTPS
- задает имя JNDI-источника данных PackageManagerDS2 для таймера
- разворачивает на сервере j2ee-приложения:
  - \* rms2.war
  - \* rpm.war

## 8.5 Порядок работы со скриптами

### 8.5.1 Инициализация параметров

Перед запуском скриптов должны быть заданы значения параметров конфигурационных файлов и Jinja-шаблонов.

### 8.5.2 Инициализация параметров конфигурационных файлов

#### hosts.ini

В этом файле необходимо задать ip-адреса серверов, на которые устанавливаются инфраструктурные сервисы (сервера приложений):

```
[ldap]
ldap1 ansible_host=192.168.1.147

[rabbit]
rbt1 ansible_host=192.168.1.147

[app]
wf1 ansible_host=192.168.1.147

[elastic]
elastic1 ansible_host=192.168.1.147

[kibana]
kibana1 ansible_host=192.168.1.147

[logstash]
logstash1 ansible_host=192.168.1.147

[db]
db1 ansible_host=192.168.1.147

[pkg]
pkg1 ansible_host=192.168.1.147
```

Поскольку данная инструкция описывает процесс установки оркестратора в режиме standalone, то все сервисы устанавливаются на одном хосте.

#### all.yaml

В этом файле необходимо задать только:

- ip-адрес сервера базы данных RMS,
- пароли доступа к БД пользователей и служб Робин
- флаг, указывающий на то, создавать или не создавать кластер для Elastic:

```

---
#DB
db_postgres_pass: Qwerty123
db_robin_pass: Qwerty123
db_users_pass: Qwerty123
db_ip: 192.168.1.147
flyway_ver: 7.8.1

# Elasticsearch
elastic_cluster: false

```

### app.yaml

В этом файле необходимо задать только:

- версию схемы
- и пароль админа WildFly

```

---
wf_version: 23.0.2
schema_version: 16.0
wf_admin_pass: Qwerty123

```

### ldap.yaml

В этом файле необходимо задать значения локального домена сервера FreeIPA, корневого DN, пароли администратора и менеджера каталогов:

```

---
domain: robin.demo
root_dn: dc=robin,dc=demo
admin_pass: Qwerty123
dm_pass: Qwerty123

```

### rabbit.yaml

В этом файле необходимо задать пароли администратора, Robin Agent, RMC, RMS, RDP, а также значение erlang\_cookie (секрет для аутентификации между нодами RabbitMQ):

```

---
rabbit_admin_pass: Qwerty123
erlang_cookie: 7KzILFz03fmS

rabbit_agent_pass: Qwerty123
rabbit_rmc_pass: Qwerty123
rabbit_rms_pass: Qwerty123
rabbit_rdp_pass: Qwerty123

```

## 8.5.3 Инициализация параметров Jinja-шаблонов

### elasticsearch-single.yml.j2

В этом файле необходимо задать:

- cluster.name - имя кластера,

- node.name - имя узла elasticsearch сервера,
- discovery.type - мода обнаружения узлов elasticsearch сервера,
- network.host - сетевой адрес elasticsearch сервера,
- path.data - директория, где лежат библиотеки elasticsearch сервера,
- path.logs - директория, где лежат логи elasticsearch сервера

### kibana.yml.j2

В этом файле необходимо задать:

- server.host - сетевой адрес kibana сервера,
- path.data - директория, где лежат данные kibana сервера

### logstash.conf.j2

В этом файле необходимо задать:

- input - описание источников сообщений,
- filter - правила обработки сообщений,
- output - правила формирования индексов сообщений.

Как составлять правила описано в оригинальной документации на logstash - <https://www.elastic.co/guide/en/logstash/current/configuration.html>

## 8.5.4 Запуск и результат выполнение скриптов

Для выполнения установки серверных компонент оркестратора RMS нужно выполнить скрипт `orcherstrator.sh`, который последовательно запустит все плейбуки:

```
ansible-playbook db.yaml
ansible-playbook elastic.yaml
ansible-playbook freeipa.yaml
ansible-playbook kibana.yaml
ansible-playbook logstash.yaml
ansible-playbook nexus.yaml
ansible-playbook rabbit.yaml
ansible-playbook wf-standalone.yaml
```

После выполнения скрипта на целевом сервере будут установлены и запущены следующие сервисы:

- postgresql-11
- elastic
- freeipa
- kibana
- logstash
- nexus.service
- rabbitmq-server.service
- wildfly.service

Пользователю через клиентские приложения Робин Платформы (Robin Player, Robin Studio, Robin Management Console) будут доступны приложения оркестратора:

- rms2.war



- rpm.war

## 8.6 Ссылки

1. <http://robingit.itbs.it.ru/devops/ansible-deploy> - скрипты
2. <https://docs.ansible.com/ansible/latest/index.html> - Документация по Ansible

---

## Сервер авторизации FreeIPA. Руководство администратора

---

### 9.1 Введение

В примерах используются:

- инструменты IPA CLI (устанавливаются при установке FreeIPA)
- веб-интерфейс FreeIPA (устанавливается при установке FreeIPA)
- инструменты ldap-utils (ldapadd, ldapmodify)
- Apache Directory Studio:  
<https://directory.apache.org/studio/downloads.html>  
[https://nightlies.apache.org/directory/studio/2.0.0.v20200411-M15/userguide/ldap\\_browser/](https://nightlies.apache.org/directory/studio/2.0.0.v20200411-M15/userguide/ldap_browser/)

### 9.2 Установка сервера FreeIPA и настройка доверительных отношений с AD

FreeIPA - это интегрированное решение для идентификации и аутентификации для сетевых сред Linux / UNIX.

В случае, когда с системой ROBIN будут работать пользователи Active Directory, необходимо настроить доверительные отношения между доменами FreeIPA и AD.

Для отказоустойчивости FreeIPA будет работать в кластере из двух мастер-реплик. В дальнейшем при необходимости количество реплик можно увеличить.

Установка и последующая интеграция *FreeIPA* с существующим AD:

[https://www.freeipa.org/page/Active\\_Directory\\_trust\\_setup](https://www.freeipa.org/page/Active_Directory_trust_setup)

Рекомендации по установке:

[https://www.freeipa.org/page/Deployment\\_Recommendations](https://www.freeipa.org/page/Deployment_Recommendations)

Настройка репликации:

[https://www.freeipa.org/page/V4/Replica\\_Setup](https://www.freeipa.org/page/V4/Replica_Setup)

Настройка брандмауэра для доменов и доверия Active Directory:

<https://docs.microsoft.com/ru-ru/troubleshoot/windows-server/identity/config-firewall-for-ad-domains-and-trusts>

### Замечания по установке

1. Синхронизировать время с контроллером домена AD, с которым планируется интегрироваться. Для этого настроить сервер *FreeIPA* на синхронизацию с тем же NTP сервером, что и AD  
[https://docs.fedoraproject.org/en-US/fedora/rawhide/system-administrators-guide/servers/Configuring\\_NTP\\_Using\\_ntpd/](https://docs.fedoraproject.org/en-US/fedora/rawhide/system-administrators-guide/servers/Configuring_NTP_Using_ntpd/)
2. *FreeIPA* должен иметь собственный основной домен, т.е. не быть поддоменом AD. Домен *FreeIPA* для AD является отдельным лесом
3. Создаваемая область *Kerberos* (название домена с большой буквы) не должна конфликтовать с существующими областями
4. Устанавливать вместе с интегрированным DNS сервером (нужен флаг `-setup-dns` при установке)
5. Перед настройкой доверительных отношений между доменами AD и *FreeIPA* необходимо чтобы серверы DNS видели друг друга. Для этого на контроллере домена AD необходимо выполнить:

```
C:\> nslookup
> set type=srv
> _ldap._tcp.ad_domain
> _ldap._tcp.ipa_domain
> quit
```

На IPA сервере:

```
# dig SRV _ldap._tcp.ipa_domain
# dig SRV _ldap._tcp.ad_domain
```

6. Создать одностороннее доверие леса *FreeIPA* конкретному домену или лесу AD на ваше усмотрение

## 9.3 Пошаговая инструкция по установке кластера FreeIPA для Centos 7

### 9.3.1 Установка основного сервера FreeIPA

1. Залогиниться под пользователем `root`
2. Обновить установленные в ОС пакеты:  
`yum update -y`
3. Установить полное доменное имя хоста:  
`hostnamectl set-hostname <ipa_hostname_1.ipa_domain>`
4. Прописать в `/etc/hosts` адреса основного и резервного серверов FreeIPA:

```
vi /etc/hosts

<main_ip> <ipa_hostname_1.ipa_domain> <ipa_hostname_1>
<reserve_ip> <ipa_hostname_2.ipa_domain> <ipa_hostname_2>
```

5. Отключить назначение IPv6 адресов:

```
vi /etc/sysctl.d/ipv6.conf

#Disable IPv6

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.<interface0>.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 0
```

6. Установить необходимые пакеты для установки FreeIPA и настройки доверительных отношений с AD
 

```
yum install -y ipa-server ipa-server-trust-ad ipa-server-dns bind bind-dyndb-ldap
```
7. Установить сервер IPA
 

```
ipa-server-install -a <admin_password> -p <dm_password> --domain=<ipa_domain>
--realm=<IPA_DOMAIN> --mkhomedir --setup-dns --no-forwarders -U
```

где:

- **admin\_password** - пароль учетной записи *admin*
  - **dm\_password** - пароль учетной записи *Directory Manager*
  - **domain** - название домена IPA должно отличаться от существующего домена AD и не быть поддоменом AD или наоборот.
  - **realm** - название домена в верхнем регистре
- 8. Получить билет администратора IPA:
 

```
kinit admin
```
- 9. Проверить, что пользователи IPA доступны для системных служб:

```
id admin
getent passwd admin
```

Если вышеуказанные команды не работают, перезапустите службу **sssd** и попробуйте их снова.

### 9.3.2 Настройка доверительных отношений с AD

1. Запустить **ipa-adtrust-install** (необходимо запустить на каждом мастер-сервере IPA):

```
ipa-adtrust-install --netbios-name=<ipa_netbios> -a <admin_password>
```

где **ipa\_netbios** - первый компонент доменного имени (например, если домен - **ipadomain.example.com**, NetBIOS - **IPADOMAIN**)

2. Вендор FreeIPA рекомендует отключить **firewalld** и включить **iptables**.  
Отключить **firewalld**:

```
systemctl stop firewalld
systemctl mask firewalld
```

Установить **iptables**:

```
yum install -y iptables-services
systemctl enable iptables
```

Исправить конфигурацию с целью открыть необходимые порты:

```
vi /etc/sysconfig/iptables

*filter
```

(continues on next page)

(продолжение с предыдущей страницы)

```

:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT

-A INPUT -s ad_ip_address -p tcp -m multiport --dports 389,636 -m state --state NEW,
↪ESTABLISHED -j REJECT

-A INPUT -p tcp -m multiport --dports 80,88,443,389,636,88,464,53,123,138,139,445 -m
↪state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m multiport --dports 88,464,53,123,138,139,389,445 -m state --state NEW,
↪ESTABLISHED -j ACCEPT
-A INPUT -p udp -j REJECT
-A INPUT -p tcp -j REJECT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

```

Перезагрузить сервис:

```
service iptables restart
```

3. Установить NTP и настроить синхронизацию с ntp-сервером. Требуется открыть порт 123 (указан в конфигурации iptables)

```
yum install -y ntp
```

4. При необходимости указать свои ntp-сервера в конфиге

```
vi /etc/ntp.conf
```

5. Запустить службу ntpd и включить автозапуск:

```
systemctl start ntpd
systemctl enable ntpd
```

6. Проверить временную зону. Должна совпадать с зоной сервера AD:

```
timedatectl | grep "Time zone"
```

и исправить, если отличается:

```
timedatectl set-timezone Europe/Moscow
```

7. Настроить DNS. Сервер AD и сервер IPA должны видеть друг друга по доменному имени. На контроллере домена AD добавить сервер условной пересылки для домена IPA:

```
C:\> dnscmd 127.0.0.1 /ZoneAdd <ipa_domain> /Forwarder <main_ip>
```

На сервере IPA в сетевых настройках указать в DNS1 - IP основного сервера IPA, DNS2 - IP резервного

```
vi /etc/sysconfig/network-scripts/ifcfg-ens192
```

8. Перезагрузить сеть:

```
service network restart
```

и добавить сервер условной пересылки для домена AD:

```
ipa dnsforwardzone-add <ad_domain> --forwarder=<ad_ip_address> --forward-policy=only
```

9. Установить доверительные отношения с доменом AD (потребуется ввести пароль):

```
ipa trust-add --type=ad <ad_domain> --admin <admin> --password
```

- **ad\_domain** - название домена AD, с которым необходимо установить доверительные отношения
- **admin** - учетная запись входящая в группу «Администраторы домена»

Команда создаст одностороннее доверительное отношение IPA домену AD. Вариант установки доверительных отношений по общему паролю:

```
ipa trust-add --type=ad ad_domain --trust-secret
```

Команда запросит общий пароль, который генерируется в AD. Так же возможно создать доверительные отношения через web-интерфейс IPA во вкладке IPA Server - Trusts или в оснастке «Домены и доверие» на контроллере домена AD.

10. Получить список доверенных доменов леса со стороны AD

```
ipa trust-fetch-domains <ad_domain>
```

### 9.3.3 Установка резервного сервера FreeIPA

1. Залогиниться под пользователем **root**
2. Обновить установленные в ОС пакеты:  
`yum update -y`
3. Установить полное доменное имя хоста:  
`hostnamectl set-hostname <ipa_hostname_2.ipa_domain>`
4. Прописать в `/etc/hosts` адреса основного и резервного серверов FreeIPA:

```
vi /etc/hosts

<main_ip> <ipa_hostname_1.ipa_domain> <ipa_hostname_1>
<reserve_ip> <ipa_hostname_2.ipa_domain> <ipa_hostname_2>
```

5. Отключить назначение IPv6 адресов:

```
vi /etc/sysctl.d/ipv6.conf

#Disable IPv6

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.<interface0>.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 0
```

6. Отключить **firewalld**:

```
systemctl stop firewalld
systemctl mask firewalld
```

7. Установить **iptables**:

```
yum install -y iptables-services
systemctl enable iptables
```

8. Исправить конфигурацию с целью открыть необходимые порты:

```
vi /etc/sysconfig/iptables

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
# -A INPUT -s ad_ip_address -p tcp -m multiport --dports 389,636 -m state --state NEW,
↪ESTABLISHED -j REJECT
-A INPUT -p tcp -m multiport --dports 80,88,443,389,636,88,464,53,123,138,139,445 -m
↪state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m multiport --dports 88,464,53,123,138,139,389,445 -m state --state NEW,
↪ESTABLISHED -j ACCEPT
-A INPUT -p udp -j REJECT
-A INPUT -p tcp -j REJECT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

9. Перезагрузить сервис:

```
service iptables restart
```

10. Установить NTP и настроить синхронизацию с ntp-сервером. Требуется открыть порт 123 (указан в конфигурации iptables)

```
yum install -y ntp
```

11. При необходимости указать свои ntp-сервера в конфиге

```
vi /etc/ntp.conf
```

12. Запустить службу ntpd и включить автозапуск:

```
systemctl start ntpd
systemctl enable ntpd
```

13. Проверить временную зону. Должна совпадать с зоной сервера AD:

```
timedatectl | grep "Time zone"
```

и исправить, если отличается:

```
timedatectl set-timezone Europe/Moscow
```

14. В сетевых настройках указать в DNS1 - IP основного сервера IPA, в DNS2 - IP резервного:

```
vi /etc/sysconfig/network-scripts/ifcfg-ens192
```

15. Перезагрузить сеть

```
service network restart
```

16. Установить необходимые пакеты для установки реплики FreeIPA и настройки доверительных отношений с AD

```
yum -y install ipa-server ipa-server-trust-ad ipa-server-dns bind bind-dyndb-ldap
```

17. Зарегистрировать хост на основном сервере IPA:

```
ipa-client-install --domain=<ipa_domain> --realm=<IPA_DOMAIN> --server=<ipa_hostname_
↪1.ipa_domain>
```

В ходе установки ответить на вопросы - `yes` и ввести логин/пароль администратора IPA

Proceed with fixed values and no DNS discovery? [no]: yes

18. На основном сервере IPA добавить хост `<ipa_hostname_2>` в группу

`ipaservers:`

```
ipa hostgroup-add-member ipaservers --hosts <ipa_hostname_2.ipa_domain>
```

19. На резервном сервере запустить:

```
ipa-replica-install
```

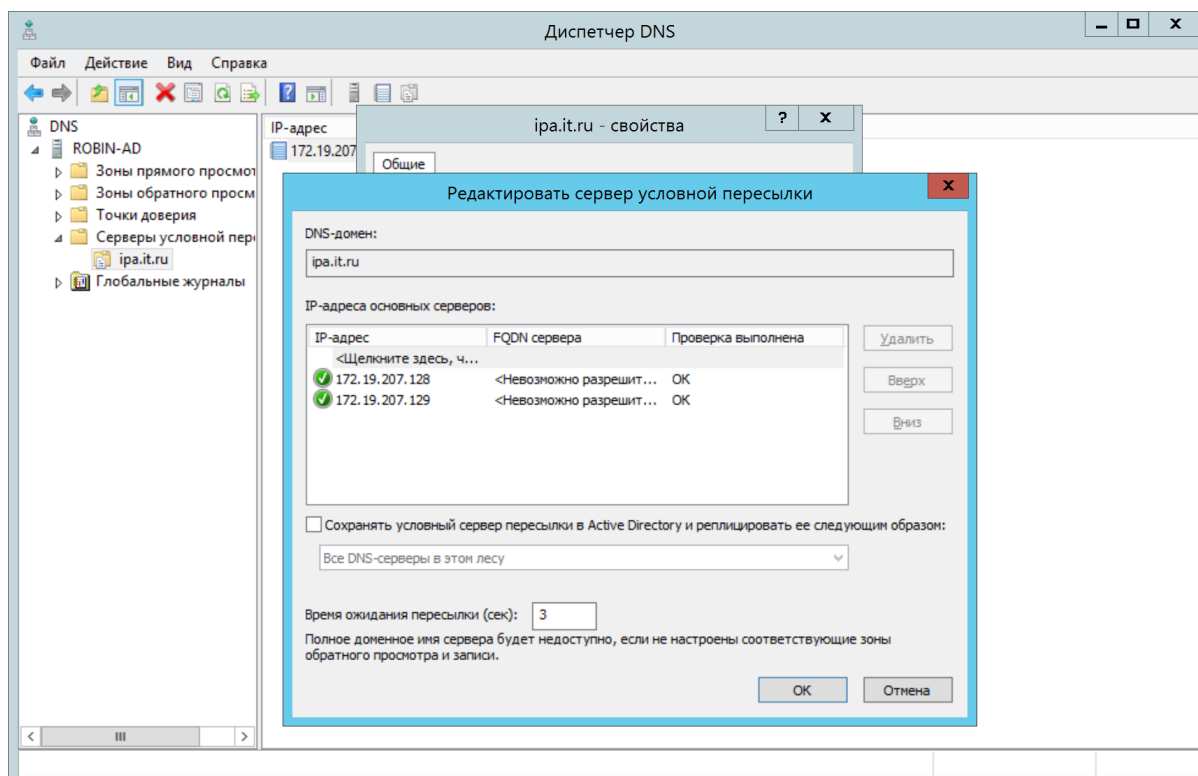
Игнорируем ошибку Reverse DNS

20. На резервном сервере установить DNS:

```
ipa-dns-install
```

Отвечаем «по» на вопросы про серверы условной пересылки

На контроллере домена AD добавить IP адрес резервного сервера в серверы условной пересылки для домена IPA:



21. На резервном сервере установить CA сервер

```
ipa-ca-install
```



22. На резервном сервере запустить ipa-adtrust-install

```
ipa-adtrust-install --netbios-name=<ipa_netbios> -a <admin_password>
```

## 9.4 Импорт каталога ROBIN

После установки и настройки сервера *FreeIPA*, в его каталог импортируется ветка ROBIN, используемая приложениями системы. Импорт осуществляется выполнением инструкций, содержащихся в LDIF-файлах.

Для импорта и последующих изменений каталога используется ldap-браузер [Apache Directory Studio](#).

1. В LDIF-файлах по всему тексту заменяется корневая запись домена «dc=domain,dc=ru» на соответствующую вашему домену FreeIPA:

- (a) С помощью PowerShell в Windows:

```
cd {директория с ldif-файлами} ``$dn={корневой DN FreeIPA} ``foreach ($file in Get-ChildItem *.ldif) { (Get-Content $file) -Replace 'dc=domain,dc=ru', $dn | out-file $file }
```

- (b) В Linux:

```
cd {директория с ldif-файлами}
dn={корневой DN FreeIPA}
find . -type f -name '*.ldif' -exec sed -i -r 's/dc=domain,dc=ru/'$dn'/g' {} \;
```

2. LDIF-файлы поочередно импортируются в порядке нумерации

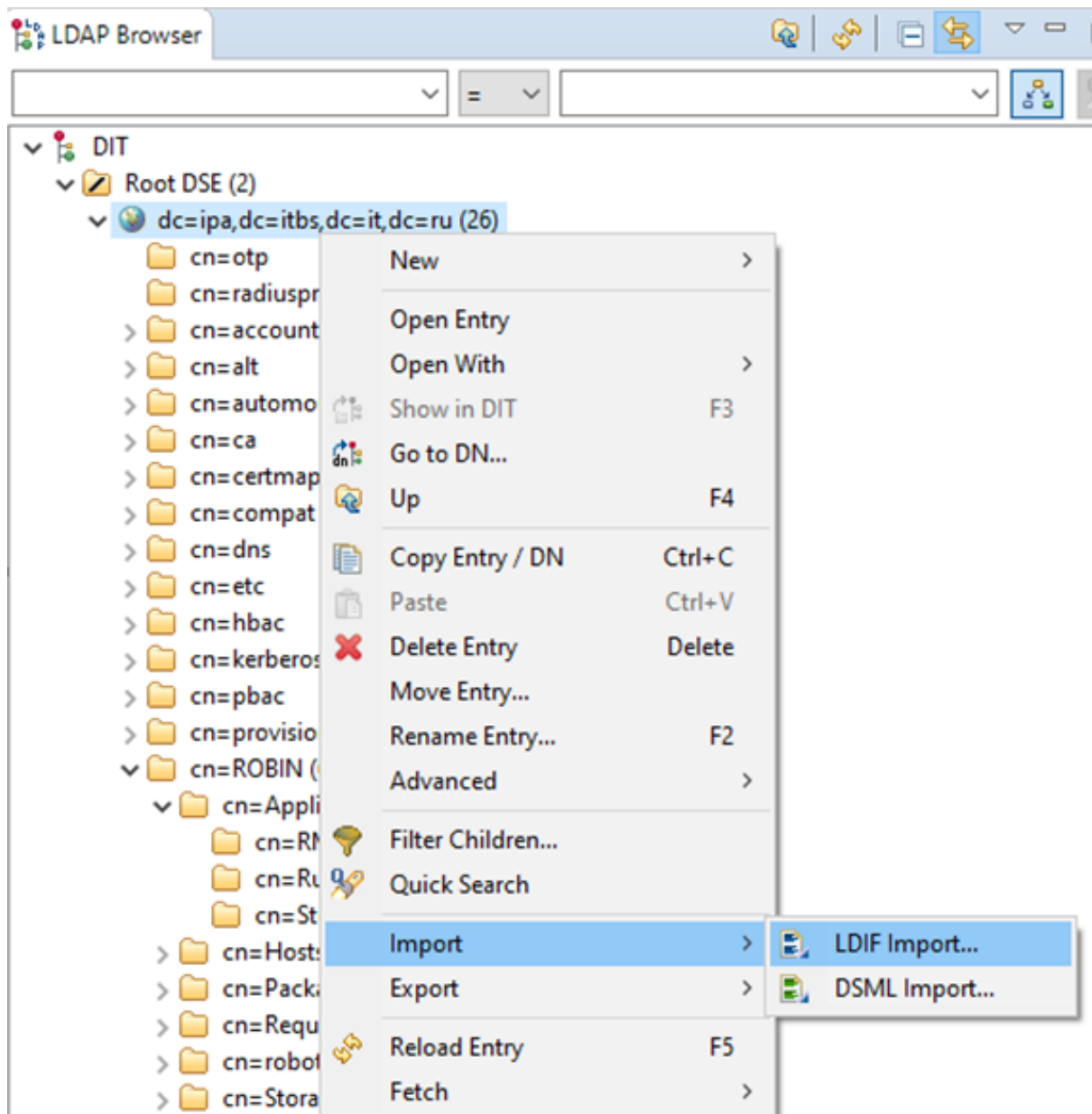
## 9.5 Синхронизация пользователей корпоративного AD-каталога, смена паролей, настройка парольных политик

### 9.5.1 Цель

Обеспечить работу пользователей из корпоративного AD-каталога с клиентскими приложениями Платформы Робин с установлением и без установления доверенных отношений между корпоративным доменом AD и доменом FreeIPA.

### 9.5.2 Описание проблемы

Для того чтобы пользователи могли работать с клиентскими приложениями (Robin Player, Robin Studio, Robin Management Console) Робин Платформы, они должны быть авторизованы сервером FreeIPA. Для выполнения процедуры авторизации пользователей их учетные записи должны храниться на сервере авторизации FreeIPA. FreeIPA-сервер может работать с пользователями из корпоративного AD-каталога только путем их переопределения в БД FreeIPA. Для этого устанавливаются доверительные отношения (трасты) между доменами AD и IPA. Процедура переопределения пользователей и создания трастов выполняется штатными средствами FreeIPA и описана в разделе *Настройка доверительных отношений с AD* этого документа. Однако создание трастов между доменами AD и IPA предъявляет определенные требования к согласованности версий AD-сервера и FreeIPA-сервера [1], которые компания, владеющая AD-каталогом, не всегда может обеспечить. Поэтому целесообразно получить такое решение, которое бы обеспечивало работу пользователей из корпоративного AD-каталога с клиентскими приложениями Платформы Робин как с установлением так и без установления доверенных отношений между корпоративным доменом AD и доменом FreeIPA.



### 9.5.3 Сервис рекуррентной репликации пользователей

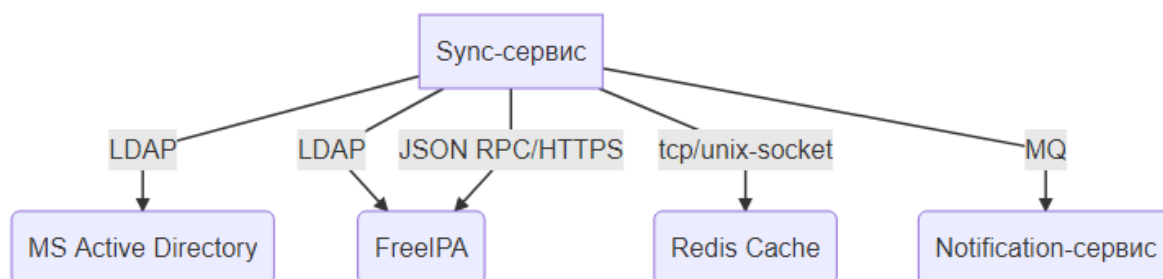
Сервис рекуррентной репликации пользователей обеспечивает работу пользователей из корпоративного AD-каталога с клиентскими приложениями Платформы Робин. При этом он может работать как с установленными доверенными отношениями между корпоративным доменом AD и доменом FreeIPA так и без них.

### 9.5.4 Функциональная архитектура сервиса

#### Функциональные компоненты

Сервис рекуррентной репликации состоит из следующих компонент:

- Sync-сервис - сервис репликации,
- Notification-сервис - обработчик очереди сообщений,
- Redis Cache - сервис кэширования репликационных состояний,
- MS Active Directory - служба каталогов Майкрософт корпоративной сети Заказчика,
- FreeIPA - сервис идентификации пользователей Платформы Робин.



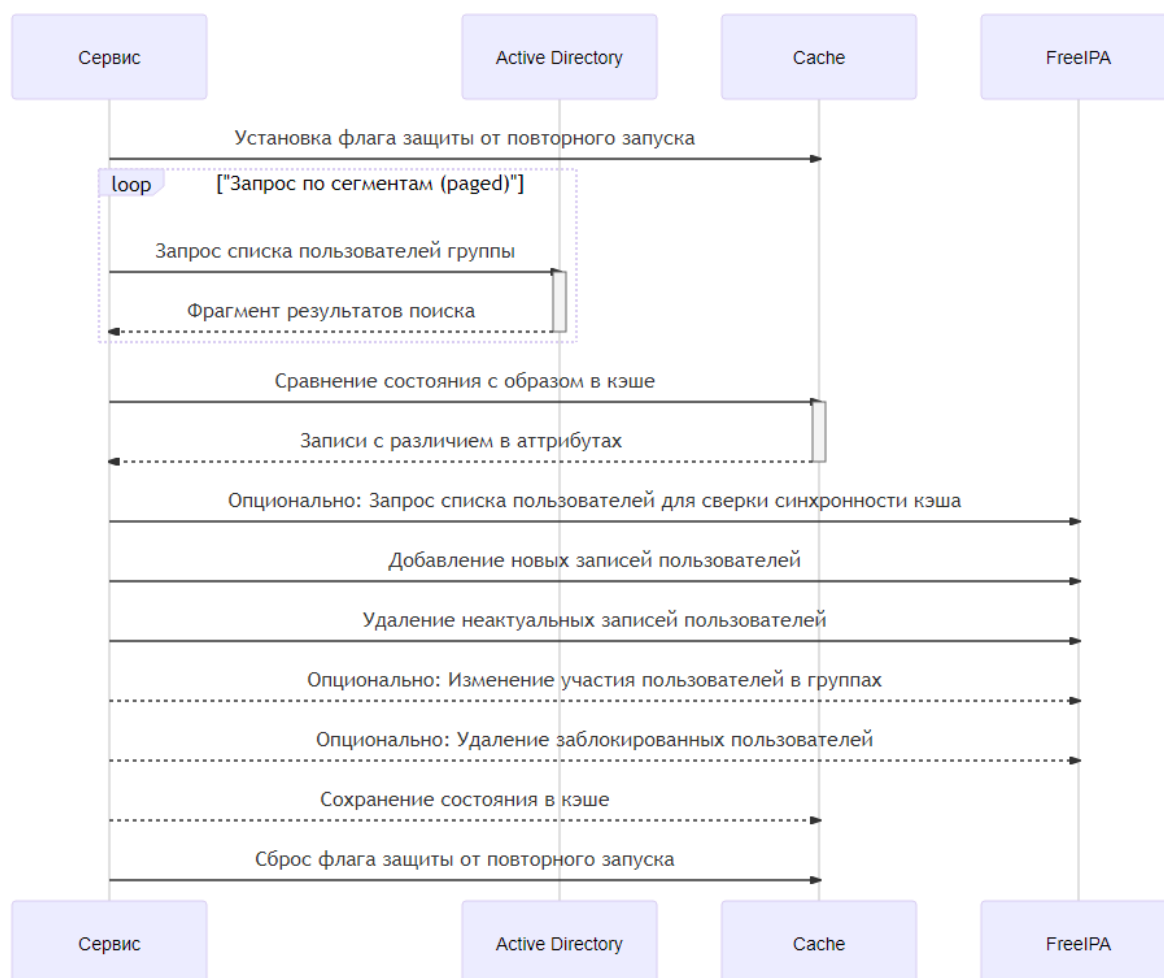
#### Диаграмма основных действий Sync-сервиса

Sync-сервис:

1. В начале проверяет состояние флага защиты от повторного запуска: если он установлен, то сервис прекращает работу, поскольку предыдущий запуск сервиса не завершил свой цикл работ. В этом случае стоит проанализировать расписание запуска сервиса - не слишком ли часто он вызывается?
2. Если флаг защиты от повторного запуска не установлен, то Sync-сервис устанавливает его, тем самым блокируя возможные запуски сервиса до завершения текущего.
3. Выполняет запрос к службе каталогов для получения списка пользователей. Запрос выполняется и обрабатывается по-фрагментно.
4. Выполняет сравнение результата поиска с образом кэша.
5. При необходимости выполняет запрос списка пользователей для сверки синхронности кэша.
6. Если изменений нет, то выполняется переход к п.8.
7. Если есть изменения, то выполняется процесс репликации - сервис отрабатывает изменения на сервере FreeIPA:
  - добавляет новых пользователей или создает их переопределения в зависимости от значения параметра `create_user_overrides`;
  - удаляет неактуальные записи пользователей;

- удаляет из домена FreeIPA пользователей, заблокированных в домене AD;
- устанавливает участие пользователей в группах;
- сохраняет текущее состояние репликации в кэше для снижения нагрузки на сервера при проверках и репликации;
- в случае регистрации новых пользователей отправляет сообщения сервису оповещения (notification-сервис).

8. По окончании своей работы выполняет сброс флага защиты от повторного запуска.



### Диаграмма основных действий Notification-сервиса

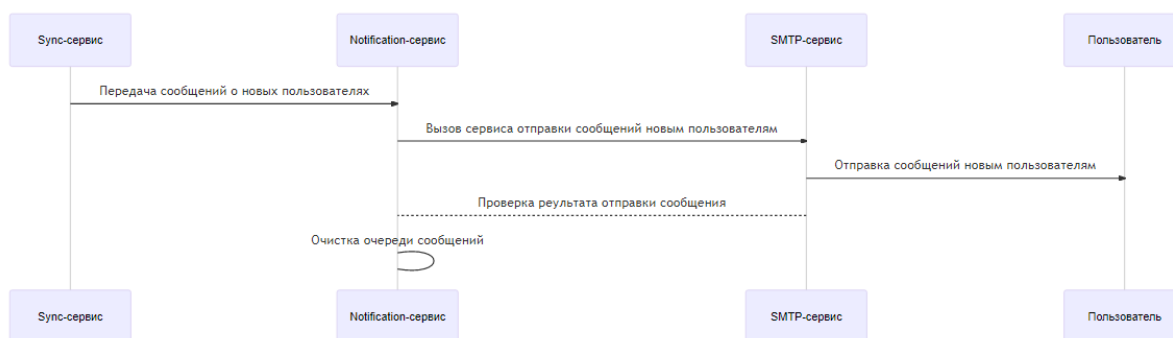
Notification-сервис:

1. Принимает сообщения от Sync-сервиса о регистрации новых пользователей.
2. Формирует по шаблону сообщения для новых пользователей.
3. Вызывает сервис отправки сообщений новым пользователям.

## 9.5.5 Установка программного обеспечения

### Требования к программному обеспечению

Для корректной работы сервиса репликации на сервере должны быть установлены:



- python 3.6.2+
- redis
- FreeIPA версии 4.9.2

## Установка Python

Установить Python версии 3.6 из официального репозитория для CentOS 7 (может потребоваться добавление репозитория epel):

```
sudo yum install python36
```

Обеспечить наличие менеджера пакетов Python:

```
python3 -m ensurepip
```

Обновить менеджер пакетов до актуальной версии:

```
sudo python3 -m pip install --upgrade pip
```

## Установка Redis

Установка Redis рекомендуется только из официальных репозиториях Cent OS (EPEL и т.п.). Для этого надо установить один из официальных репозиториях расширений CentOS 7, например:

```
sudo yum install -y epel-release
```

Далее установить Redis:

```
sudo yum install -y redis
```

## Установка vim

Установить текстовый редактор vim:

```
sudo yum install vim
```

## Установка сервиса репликации

Для работы сервиса не требуется полномочий root-пользователя. Для установки рекомендуется создать некорневую пользовательскую учетную запись с домашней директорией:

```
sudo adduser robin-active-ipa
```

После создания пользователя, перейти в домашнюю директорию (далее называемой корневой директорией сервиса):

```
cd /home/robin-active-ipa
```

Скопировать в нее предоставленный архив приложения и распаковать его.

```
cp <путь к архиву> .
tar -xvf robin-active-ipa.tar
```

Создать виртуальное окружение

```
python3 -m venv venv
```

Активировать виртуальное окружение для последующих шагов установки

```
source venv/bin/activate
```

Обновить менеджер пакетов pip:

```
python3 -m pip install --upgrade pip
```

Установить предоставленные пакеты wheel в режиме оффлайн:

```
tar -xvf wheelhouse.tar
python3 -m pip install -r requirements.txt --no-index --find-links wheelhouse
```

## 9.5.6 Настройка программного обеспечения

### Настройка имен сервиса FreeIPA

При установке требуется проверить, и, при необходимости обеспечить корректное разрешение доменного имени удаленного хоста FreeIPA, если это не обеспечивается настройками `dns` в сети (внимание, корректный доступ к функциям API FreeIPA по ip-адресу невозможен):

```
sudo vim /etc/hosts
```

Добавить правило в файл:

```
<ip address> <freeipa domain name>
```

Проверить корректность разрешения имени:

```
ping <freeipa domain name>
```

### Настройка Redis

Настроить Redis с привязкой к локальному адресу. Допустима настройка с применением unix-socket.

```
sudo vim /etc/redis.conf
```

Проверить наличие единственной привязки сервиса для протокола IPv4:

```
...
bind 127.0.0.1
...
```

## 9.5.7 Настройка Sync-сервиса

### Переменные окружения и настройки

Для хранения настроек можно использовать файл `json` (рекомендуется), переменные среды окружения (для простых конфигураций), директории с ограниченным доступом (`secrets`) для хранения учетных записей и паролей для доступа к серверам (например, `/var/run/`)

### Параметры настроек доступа к службе MS Active Directory

Для доступа к AD серверу в настройках должен быть указан его адрес (`ad_server`), имя пользователя (`ad_bind_user`), от имени которого сервис будет выполнять запросы к AD серверу и его пароль (`ad_bind_pwd`).

Пользователи в домене AD могут входить в одну или несколько глобальных групп. Синхронизация пользователей происходит через их вхождение в ту или иную глобальную группу. Имена наблюдаемых групп, пользователи которых подлежат синхронизации с пользователями в домене FreeIPA, задаются в параметре `ad_target_group_dn_list`.

Для выполнения поиска в учетных записях AD должен быть задан базовый DN поиска - `ad_search_base_dn`.

#### Атрибут `ad_server`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_AD_SERVER`

**Тип:** `string`

**Описание:** Адрес хоста Microsoft Active Directory (имя или ip).

#### Атрибут `ad_bind_user`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_AD_BIND_USER`

**Тип:** `string`

**Описание:** Имя пользователя Microsoft Active Directory.

#### Атрибут `ad_bind_pwd`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_AD_BIND_PWD`

**Тип:** `string`

**Описание:** Пароль пользователя Microsoft Active Directory.

#### Атрибут `ad_search_base_dn`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_AD_SEARCH_BASE_DN`

**Тип:** `string`

**Описание:** Базовый DN поиска пользователей в локальном домене Microsoft Active Directory.

**Атрибут `ad_target_group_dn_list`****Имя переменной среды (альтернатива):** `ROBIN_A2I_AD_TARGET_GROUP_DN`**Тип:** `array[string]`**Описание:** Наблюдаемые группы в Microsoft Active Directory. При использовании в переменной окружения - строка JSON заключенная в одинарные кавычки. Имеет более низкий приоритет, чем `ROBIN_A2I_GROUPS_MATCHING`. При работе по этому списку пользователи создаются без принадлежности к какой-либо дополнительной группе**Параметры настроек FreeIPA сервера**

Для доступа к FreeIPA серверу в настройках должно быть указано его доменное имя (`freeipa_server`), имя пользователя (`freeipa_bind_user`), от имени которого сервис будет выполнять запросы к FreeIPA серверу, DN пользователя (`freeipa_bind_user_dn`) и его пароль (`freeipa_bind_pwd`).

**Атрибут `freeipa_bind_user`****Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_BIND_USER`**Тип:** `string`**Описание:** Имя пользователя FreeIPA (CN).**Атрибут `freeipa_bind_user_dn`****Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_BIND_USER_DN`**Тип:** `string`**Описание:** Имя пользователя FreeIPA (DN).**Атрибут `freeipa_bind_pwd`****Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_BIND_PWD`**Тип:** `string`**Описание:** Пароль пользователя FreeIPA.**Параметры настроек реплицированных пользователей**

Пользователи в FreeIPA располагаются в двух областях:

- основной - для реплицированных пользователей в случае отсутствия доверенных отношений между AD и FreeIPA.
- дополнительной - для переопределений пользователей в случае установления доверенных отношений между AD и FreeIPA.

Область для реплицированных пользователей задается параметром `freeipa_user_search_base` - DN для родительского узла / контейнера пользователей. Область для переопределений задается параметрами `freeipa_trust_id_view_dn` и `freeipa_trust_id_view_cn`.

В основной области располагаются также и не модифицируемые сервисом пользователи, зарегистрированные администратором FreeIPA. Эти пользователи указываются в параметре `freeipa_readonly_users` и не попадают в область действия sync-сервиса.



**Важно.** Если пользователи либо перепределения, созданные при работе системы, удаляются либо изменяются через административный интерфейс пользователя FreeIPA, то действия администратора системы имеют более высокий приоритет и при автоматической репликации изменений такие записи обновляться не будут. При необходимости продолжения автоматических обновлений этих записей необходимо осуществить ручной сброс кэша репликации:

```
cd /home/robin-active-ipa
./cache-reset --conf=config.json
```

#### Атрибут `freeipa_user_search_base`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_USER_SEARCH_BASE`

**Тип:** string

**Описание:** DN для родительского узла / контейнера пользователей.

#### Атрибут `freeipa_trust_id_view_dn`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_TRUST_ID_VIEW_DN`

**Тип:** string

**Описание:** DN для ID View

#### Атрибут `freeipa_trust_id_view_cn`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_TRUST_ID_VIEW_CN`

**Тип:** string

**Описание:** CN для ID View.

#### Атрибут `freeipa_readonly_users`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_FREEIPA_READONLY_USERS`

**Тип:** array[string]

**Описание:** Пользователи FreeIPA, которые не должны изменяться при репликации. Указываются короткие имена (uid в терминологии FreeIPA). При использовании в переменной окружения - строка JSON заключенная в одинарные кавычки.

#### Отображение атрибутов пользователя при репликации

При репликации пользователей AD во FreeIPA транслируются не все их атрибуты, а только их часть. (см. Таблицу 1).

**Таблица 1. Отображение атрибутов пользователя при репликации.**

Active Directory	FreeIPA (user)	FreeIPA (id override)
<code>'userPrincipalName'</code>		<code>ipaOriginalUID, ip aAnchorUUID<sup>3</sup></code>
<code>name, sAMAccountName</code>	<code>uid</code>	<code>uid</code>
<code>givenname</code>	<code>givenname</code>	
<code>cn</code>	<code>cn</code>	
<code>name</code>	<code>sn</code>	
<code>mail</code>	<code>mail</code>	
<code>memberOf</code>	<code>memberOf, memberof_group</code>	<code>memberOf, memberof_group</code>

## Параметры настройки работы Sync-сервиса

Сервис позволяет:

- создавать пользователей либо их переопределения (параметр `create_user_overrides`) во FreeIPA из пользователей отдельной группы либо групп, выбранной внутри MS Active Directory (параметр `ad_target_group_dn_list`),
- отображать группы Active Directory на указанные группы во FreeIPA, а также отображать участие пользователей в группах соответствии с заданными правилами (параметр `groups_matching`),
- удалять пользователей (либо переопределения) FreeIPA, для которых уже не существует оригинала в MS Active Directory (параметр `remove_unrelated_freeipa_anchors`),
- отслеживать статус постоянной блокировки пользователей в Active Directory и удалять пользователей FreeIPA, заблокированных в домене AD (параметр `remove_disabled_users`),
- при необходимости автоматически создавать non-posix группы во FreeIPA и помещать в них пользователей FreeIPA на основе таблицы сопоставления «группа Active Directory - группа FreeIPA» (параметр `groups_matching`),
- опционально отслеживать изменения участия пользователя в группах Active Directory и актуализировать участия пользователей FreeIPA в локальных группах на основе таблицы сопоставления (параметр `groups_matching`),
- опционально создавать локальные группы, указанные в таблице сопоставления, если они отсутствует во FreeIPA (параметр `allow_create_local_groups`),
- хранить предыдущее состояние репликации в кэше для снижения нагрузки на сервера при проверках и репликации (см. настройки Redis Cache),
- для режима с созданием пользователей во FreeIPA создавать очереди сообщений для отправлять пользовательские уведомлений о регистрации с данными учетной записи (см. настройки SMTP-сервера),
- параметр `search_page_size` позволяет ограничивать размер фрагмента результатов для итерационного поиска, единиц записей, а также включать режим «постраничного» итерационного запроса для каталогов Active Directory с большим количеством записей (для решения проблемы с ограничением количества записей в результате запроса в Active Directory).

**Примечание.** Выбор способа репликации учетных записей (пользователь либо переопределение пользователя), настраиваемого значением атрибута `create_user_overrides`, осуществляется при установке сервиса и изменение этого режима в ходе эксплуатации не предусмотрено.

### Атрибут `create_user_overrides`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_CREATE_USER_OVERRIDES`

<sup>3</sup> Первичное значение при создании переопределения, во внутреннем представлении FreeIPA идентификатор `ipaAnchorUUID` может заменяться на ссылку `SID`.

**Тип:** boolean (true/false)

**Описание:** Создавать во FreeIPA переопределения пользователей (idoverride) вместо пользователей (user), включено по умолчанию. В переменных среды окружения допустимо указание (yes/no). Значение по умолчанию равно true.

**Атрибут** freeipa\_request\_timeout\_sec

**Имя переменной среды (альтернатива):** ROBIN\_A2I\_FREEIPA\_REQUEST\_TIMEOUT\_SEC

**Тип:** integer

**Описание:** Макс время ожидания ответа FreeIPA (сек).

**Атрибут** remove\_unrelated\_freeipa\_anchors

**Имя переменной среды (альтернатива):** ROBIN\_A2I\_REMOVE\_UNRELATED\_FREEIPA\_ANCHORS

**Тип:** boolean (true/false)

**Описание:** Удаление пользователей FreeIPA, для которых не найден оригинал в Active Directory. В переменных среды окружения допустимо указание (yes/no).

**Атрибут** remove\_disabled\_users

**Имя переменной среды (альтернатива):** ROBIN\_A2I\_REMOVE\_DISABLED\_USERS

**Тип:** boolean (true/false)

**Описание:** Удаление пользователей FreeIPA, заблокированных в Active Directory. В переменных среды окружения допустимо указание (yes/no).

**Атрибут** groups\_matching

**Имя переменной среды (альтернатива):** ROBIN\_A2I\_GROUPS\_MATCHING

**Тип:** Объект JSON

**Описание:** Сопоставление групп Active Directory и FreeIPA. Ключи объекта JSON - DN групп Active Directory, значения - DN группы FreeIPA.

**Пример значения параметра “groups\_matching“:**

```
{ "CN=fake-group-blue,CN=Users,DC=robin,DC=it,DC=ru": "cn=freeipa-blue,cn=groups,cn=accounts,
↪dc=ipa,dc=it,dc=ru",
  "CN=fake-group-green,CN=Users,DC=robin,DC=it,DC=ru": "cn=freeipa-green,cn=groups,cn=accounts,
↪dc=ipa,dc=it,dc=ru" }
```

**Атрибут** allow\_create\_local\_groups

**Имя переменной среды (альтернатива):** ROBIN\_A2I\_ALLOW\_CREATE\_LOCAL\_GROUPS

**Тип:** boolean (true/false)

**Описание:** Разрешить создание групп во FreeIPA если они отсутствуют. В переменных среды окружения допустимо указание (yes/no).

**Атрибут** `search_page_size`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_SEARCH_PAGE_SIZE`

**Тип:** integer

**Описание:** Ограничение на размер фрагмента результатов для итерационного поиска, единиц записей.

**Атрибут** `log_file`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_LOG_FILE`

**Тип:** string

**Описание:** Путь к файлу журнала (относительный, если находится внутри директории сервиса, полный - если расположен во внешних директориях).

**Параметры настроек Redis Cache**

Сервис позволяет :

- хранить предыдущее состояние репликации (пользователи, группы) в кэше для снижения нагрузки на сервера при проверках и репликации
- сверять содержимое кэша и FreeIPA (обратная сверка) - параметр `sync_cache_with_freeipa`.

Параметры настройки режима работы Redis Cache-сервиса:

- параметр `redis_uri` задает адрес (`схема://хост:порт`) сервера Redis. Возможные схемы: `redis://`, `unix://`
- параметр `process_lock_expiration_sec` задает интервал времени автоматической очистки флага блокировки (сек),
- параметр `cache_expiration_seconds` задает время сброса кэша синхронизации в секундах, отключено - 0,

**Атрибут** `redis_uri`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_REDIS_URI`

**Тип:** string

**Описание:** Адрес (`схема://хост:порт`) сервера Redis. Возможные схемы: `redis://`, `unix://`.

**Атрибут** `process_lock_expiration_sec`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_PROCESS_LOCK_EXPIRATION`

**Тип:** integer

**Описание:** Интервал автоматической очистки флага блокировки (сек).

**Атрибут** `cache_expiration_seconds`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_CACHE_EXPIRATION_SECONDS`

**Тип:** integer

**Описание:** Время сброса кэша синхронизации в секундах, отключено - 0

**Атрибут** `sync_cache_with_freeipa`**Имя переменной среды (альтернатива):** `ROBIN_A2I_SYNC_CACHE_WITH_FREEIPA`**Тип:** boolean (true/false)**Описание:** Сверять содержимое кэша и FreeIPA (обратная сверка). В переменных среды окружения допустимо указание (yes/no).**Параметры настроек SMTP-сервера**

SMTP-сервис используется обработчиком очереди сообщений от Sync-сервиса для отправки пользовательских уведомлений. Для настройки SMTP-сервиса используются следующие параметры:

**Таблица 2. Параметры настройки работы SMTP-сервиса.**

Атрибут настройки	Имя переменной среды (альтернатива)	Тип	Описание
<code>smtp_host</code>	<code>ROBIN_A2I_SMTP_HOST</code>	string	Адрес (имя/ip) шлюза пересылки почты
<code>smtp_port</code>	<code>ROBIN_A2I_SMTP_PORT</code>	integer	Порт шлюза пересылки почты.
<code>smtp_user</code>	<code>ROBIN_A2I_SMTP_USER</code>	string	Имя учетной записи для доступа к шлюзу пересылки почты
<code>smtp_pwd</code>	<code>ROBIN_A2I_SMTP_PWD</code>	string	Пароль учетной записи для доступа к шлюзу пересылки почты

**Параметры настроек Notification-сервис**

Notification-сервис обрабатывает сообщения от Sync-сервиса о новых пользователях и отправляет им уведомления по электронной почте. Шаблон текста письма:

```

Уважаемый пользователь!
Для вас зарегистрирована учетная запись в домене linux со следующими данными:
Имя пользователя: $uname
Пароль: $pwd
P.S. Это письмо сформировано автоматически, просьба не отвечать на него.

```

параметры шаблона:

- `$uname` - имя нового пользователя в системе,
- `$pwd` - пароль нового пользователя в системе.

**Важно.** Уведомления пользователям по электронной почте отправляются только в режиме создания пользователей (user) во FreeIPA. В режиме с созданием переопределений (представлений ID) отправка сообщений не применяется. Для отправки сообщений пользователям необходимо, чтобы атрибут `mail` в записи пользователя Active Directory содержал актуальное значение действующего адреса пользователя.

**Атрибут** `email_ntf_user_new_template`**Имя переменной среды (альтернатива):** `ROBIN_A2I_EMAIL_NTF_USER_NEW_TEMPLATE`**Тип:** string**Описание:** Путь к файлу шаблона почтового сообщения для новых пользователей.

**Атрибут** `email_ntf_user_new_subject`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_EMAIL_NTF_USER_NEW_SUBJECT`

**Тип:** `string`

**Описание:** Тема почтового сообщения для новых пользователей.

**Атрибут** `smtp_ntf_sender`

**Имя переменной среды (альтернатива):** `ROBIN_A2I_SMTP_NTF_SENDER`

**Тип:** `string`

**Описание:** Отправитель почтового сообщения для новых пользователей (поле «from», видимое в сообщении).

### Пример шаблона файла конфигурации

Создать файл конфигурации в формате `json` на основе описания атрибутов из раздела «Переменные окружения и настройки».

Файл может быть создан в корневой директории сервиса, в этом случае при указании имени файла в аргументах вызова сервиса можно использовать только имя файла без полного пути (`--conf=config.json`).

```
[admin@host ~] vim config.json
```

следующего содержания:

```
{
  "ad_server": "<хост Active Directory>",
  "ad_bind_user": "<имя>",
  "ad_bind_pwd": "<пароль>",
  "ad_search_base_dn": "CN=Users,DC=robin,DC=it,DC=ru",
  "ad_target_group_dn_list": [
    "CN=freeipa-sync-test,CN=Users,DC=robin,DC=it,DC=ru"
  ],
  "freeipa_server": "<имя хоста FreeIPA>",
  "freeipa_bind_user_dn": "uid=<имя>,cn=users,cn=accounts,dc=ipa,dc=it,dc=ru",
  "freeipa_bind_user": "<имя>",
  "freeipa_bind_pwd": "<пароль>",
  "freeipa_readonly_users": [
    "admin@robin.it.ru",
    "user1@robin.it.ru",
    "test@robin.it.ru"
  ],
  "freeipa_trust_id_view_dn": "cn=Default Trust View,cn=views,cn=accounts,dc=ipa,dc=it,dc=ru",
  "freeipa_trust_id_view_cn": "Default Trust View",
  "freeipa_user_search_base": "cn=users,cn=accounts,dc=ipa,dc=it,dc=ru",
  "remove_unrelated_freeipa_anchors": false,
  "remove_disabled_users": false,
  "groups_matching": {
    "CN=fake-group-blue,CN=Users,DC=robin,DC=it,DC=ru": "cn=freeipa-blue,cn=groups,
    cn=accounts,dc=ipa,dc=it,dc=ru",
    "CN=fake-group-green,CN=Users,DC=robin,DC=it,DC=ru": "cn=freeipa-green,cn=groups,
    cn=accounts,dc=ipa,dc=it,dc=ru"
  },
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"redis_uri": "redis://127.0.0.1:6379",
"search_page_size": 50,
"allow_create_local_groups": true,
"sync_cache_with_freeipa": false,
"smtp_host": "127.0.0.1",
"smtp_port": 2500,
"smtp_user": "",
"smtp_pwd": "",
"email_ntf_user_new_template": "notifyacc.html",
"create_user_overrides": false,
"log_file": "events.log"
}

```

## 9.5.8 Запуск и остановка сервисов

### Запуск Redis

Разрешить автоматический запуск Redis и запустить сервис

```

sudo systemctl start redis.service
sudo systemctl enable redis

```

### Настройка таблицы cron для запуска сервисов sync и notify

Проверить и скопировать путь к установленному интерпретатору Python.

```
which python3
```

Перейти к редактированию таблицы cron

```
crontab -e
```

Добавить задания для периодического запуска сервиса и для обработчика очереди сообщений (с указанием пути к ранее созданному файлу конфигурации, в примере - файл находится в корневой директории сервиса):

```

<* * * * * > /home/robin-active-ipa/sync.sh --conf=config.json # Активация сверки и репликации
<* * * * * > /home/robin-active-ipa/notifications.sh --conf=config.json # Активация сверки и
↪репликации

```

**Примечание.** Интервалы выполнения задач указываются в соответствии с синтаксисом таблиц cron, в примере график времени выполнения заключен в угловые скобки. Интервалы запуска sync.sh выбираются с учетом необходимой оперативности обновления данных и с учетом рационального ограничения нагрузки на сервера, интервалы обработки очереди сообщений - порядка 15-30 минут.

## 9.5.9 Задание парольной политики

Управлять парольными политиками пользователей администратор FreeIPA может через веб-интерфейс FreeIPA или через командную строку с помощью команд `pwpolicy-add` и `pwpolicy-mod` [2].

Задать парольную политику можно также с помощью модуля Ansible - `community.general.ipa_pwpolicy` [3].

Пример задания для этого модуля:

```
- name: Modify the global password policy
  community.general.ipa_pwpolicy:
    maxpwdlife: '90'
    minpwdlife: '1'
    historylength: '8'
    minclasses: '3'
    minlength: '16'
    maxfailcount: '6'
    failinterval: '60'
    lockouttime: '600'
    ipa_host: ipa.example.com
    ipa_user: admin
    ipa_pass: topsecret
```

### 9.5.10 Смена паролей

Смену пароля пользователь выполняет самостоятельно в своем клиентском приложении (Robin Player, Robin Studio, Robin Management Console). Процедура смены пароля описана в руководстве оператора соответствующего клиентского приложения.

### 9.5.11 Ссылки

1. Active Directory trust setup. - [https://www.freeipa.org/page/Active\\_Directory\\_trust\\_setup](https://www.freeipa.org/page/Active_Directory_trust_setup)
2. Defining Password Policies - [https://docs.fedoraproject.org/en-US/Fedora/18/html/FreeIPA\\_Guide/user-pwdpolicy.html](https://docs.fedoraproject.org/en-US/Fedora/18/html/FreeIPA_Guide/user-pwdpolicy.html)
3. Manage FreeIPA password policies - [https://docs.ansible.com/ansible/latest/collections/community/general/ipa\\_pwpolicy\\_module.html](https://docs.ansible.com/ansible/latest/collections/community/general/ipa_pwpolicy_module.html)

## 9.6 Создание групп и добавление в группы

### 9.6.1 Создание записи

На примере рассмотрено создание записи, являющейся группой записей о пакетах с роботами *Robots For Test*.

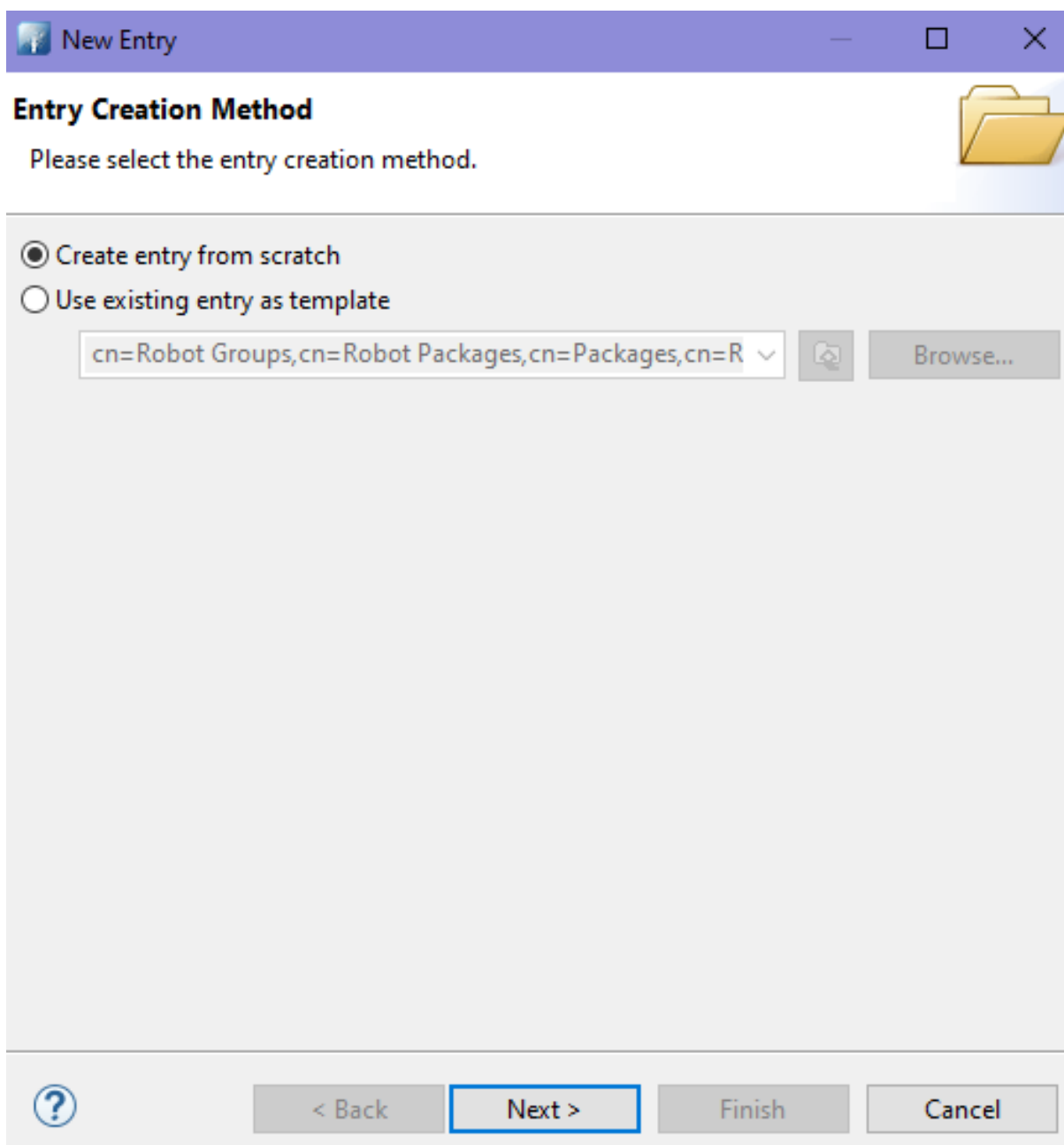
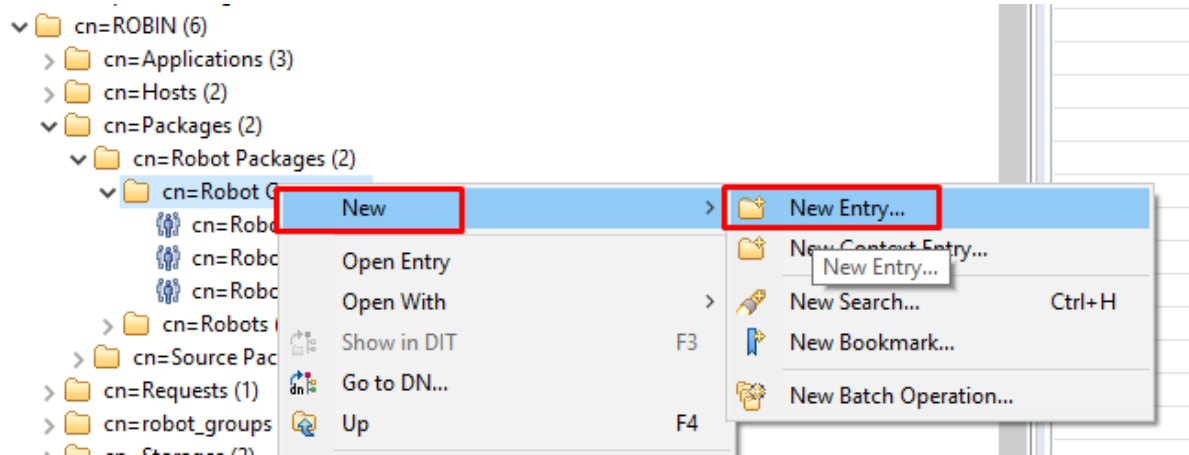
**Замечание:** Создание группы пользователей будет описано позже в специальном разделе.

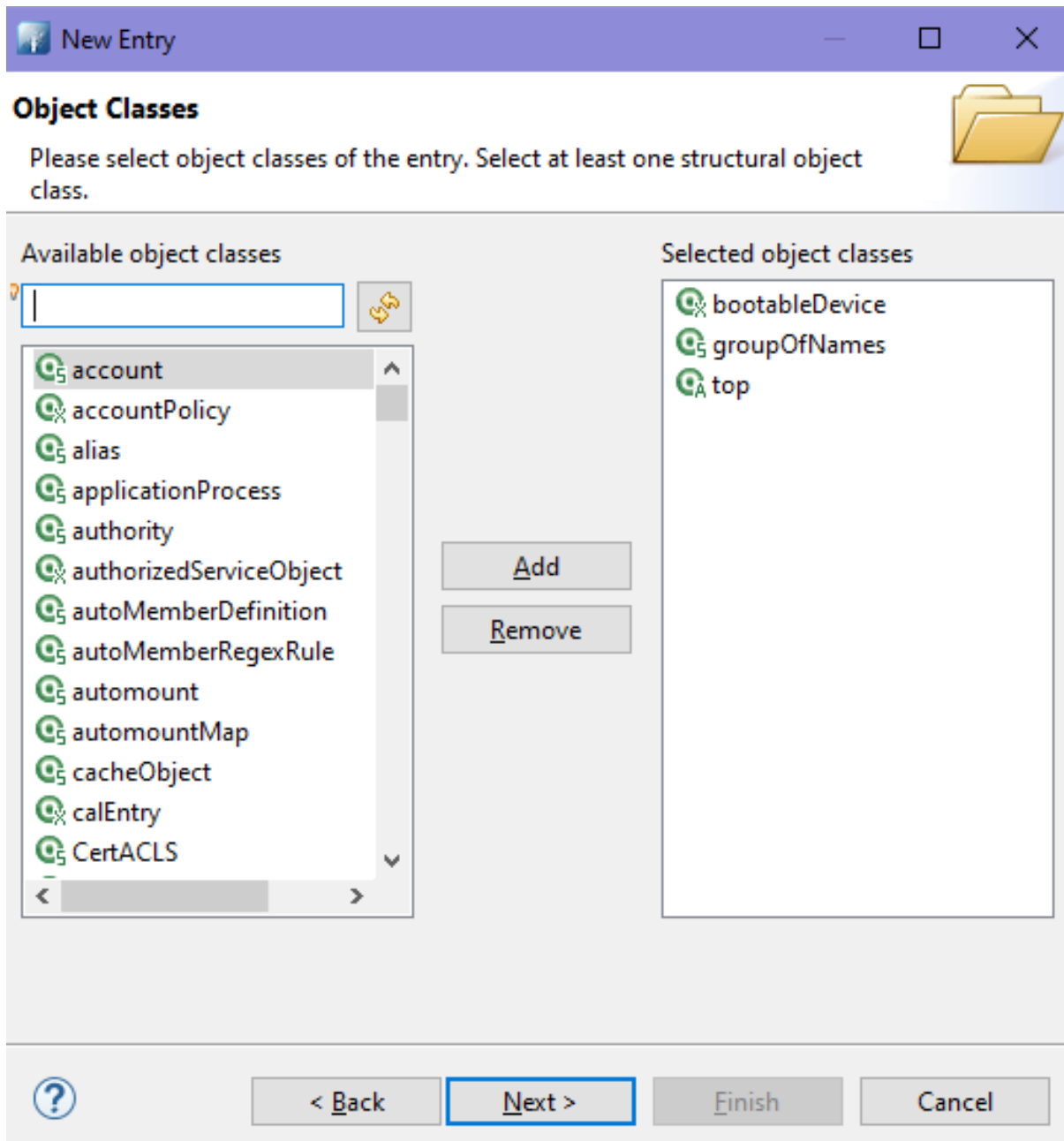
#### Способ 1 - через Apache Directory Studio

Нужно перейти к записи, относительно которой будет создаваться новая, и кликнуть на неё правой кнопкой мыши.

1. Выбрать *New - New Entry...*
2. Выбрать *Create entry from scratch* и нажать кнопку *Next*:
3. Выбрать объектные классы, необходимые для создания новой записи, и нажать *Next*. О том, какие объектные классы выбирать, будет написано позже при рассмотрении каждой конкретной записи.
4. Ввести RDN новой записи:
5. Нажать *Finish*.







**New Entry**

**Distinguished Name**

Please select the parent of the new entry and enter the RDN.

Parent:

RDN:  =

DN Preview:

## Способ 2 - через командную строку с помощью команды `ldapadd` (с использованием `ldif`-файла)

Общий вид `ldif`-файла при использовании команды `ldapadd`:

```
dn: DN
attr1: attrValue1
...
attrN: attrValueN
objectClass: objClass1
...
objectClass: objClass2
```

- *DN* - DN новой записи
- *attr1*, ... , *attrN* - атрибуты новой записи
- *objClass1*, ... *objClassN* - объектные классы, используемые для создания новой записи

Тогда команда будет выглядеть следующим образом:

```
ldapadd parameters -f path_to_file/file.ldif
```

*parameters* - другие параметры команды `ldapadd` (хост для LDAP-сервера, порт, учётная запись и т. д.)

Пример `ldif`-файла:

```
dn: cn=Robots For Test,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↪dc=it,dc=ru
cn: Robots For Test
objectClass: groupOfNames
objectClass: top
objectClass: bootableDevice
```

## Способ 3 - через командную строку с помощью команды `ldapadd` (без использования `ldif`-файла)

Ввести команду:

```
ldapadd parameters
```

*parameters* - другие параметры команды `ldapadd` (хост для LDAP-сервера, порт, учётная запись и т. д.)

После этого ввести остальные данные:

```
dn: cn=Robots For Test,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↪dc=it,dc=ru
cn: Robots For Test
objectClass: groupOfNames
objectClass: top
objectClass: bootableDevice
```

### 9.6.2 Добавление атрибута для записи

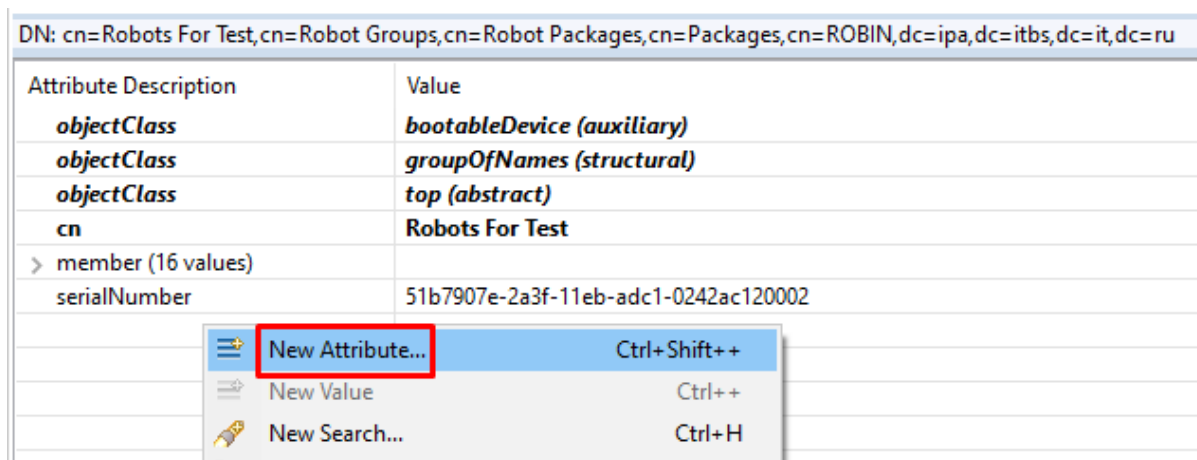
В примере рассматривается добавление записи о версии робота *Robin.Department.Test2-20201202120913Z* в группу *Robots For Test*.

Это происходит путём добавления атрибута `member` с DN записи *Robin.Department.Test2-20201202120913Z* в запись с группой **Robots For Test**.

## Способ 1 - через Apache Directory Studio

Нужно перейти к нужной записи и вызвать контекстное меню для записи (правой кнопкой мыши).

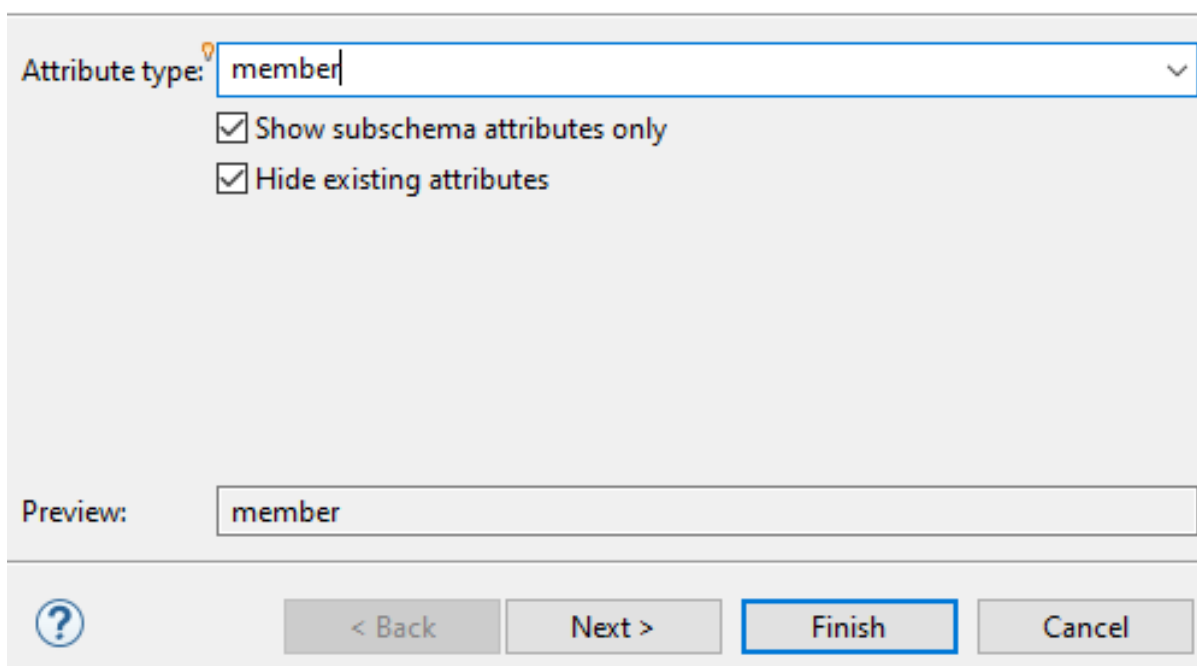
1. Выбрать *New Attribute...*



2. В открывшемся окне набрать «member» в поле ввода *Attribute type* и нажать *Finish*.

### Attribute Type

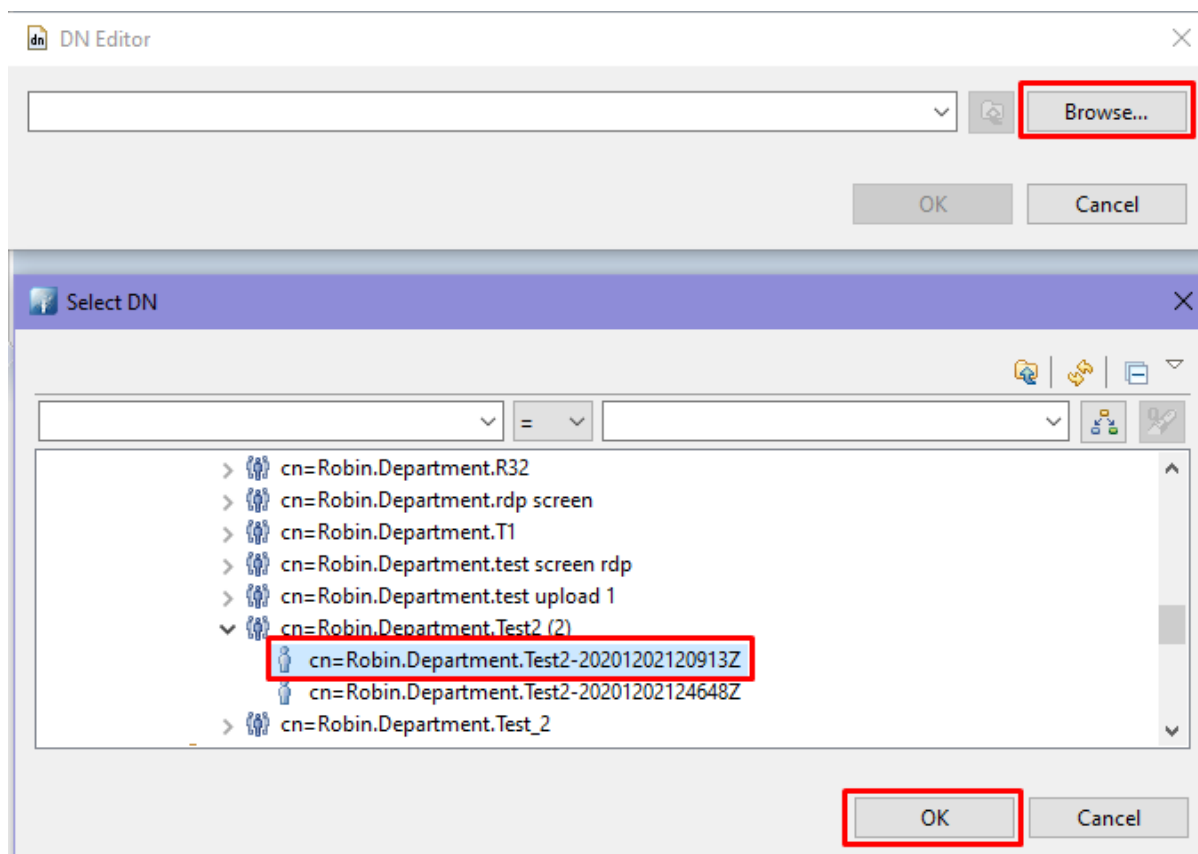
Please enter or select the attribute type.



3. Нажать *Browse...*, выбрать нужную запись, нажать *OK*.

## Способ 2 - через командную строку с помощью команды `ldapmodify` (с использованием `ldif`-файла)

Общий вид `ldif`-файла при использовании команды `ldapmodify` для добавления атрибута:



```
dn: DN
changetype: modify
add: attrAdd
attrAdd: attrAddValue
```

где:

- *DN* - DN изменяемой записи
- *attrAdd* - новый атрибут
- *attrAddValue* - значение нового атрибута

Тогда команда будет выглядеть следующим образом:

```
ldapmodify parameters -f path_to_file/file.ldif
```

где:

- *parameters* - другие параметры команды `ldapmodify` (хост для LDAP-сервера, порт, учётная запись и т. д.)

Пример `ldif`-файла:

```
dn: cn=Robots For Test,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↵dc=it,dc=ru
changetype: modify
add: member
member: cn=Robin.Department.Test2-20201202120913Z,cn=Robin.Department.Test2,cn=Robots,cn=Robot,
↵Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

### Способ 3 - через командную строку с помощью команды `ldapmodify` (без использования `ldif-файла`)

Ввести команду:

```
ldapmodify parameters
```

где:

- *parameters* - другие параметры команды `ldapmodify` (хост для LDAP-сервера, порт, учётная запись и т. д.)

После этого ввести остальные данные:

```
dn: cn=Robots For Test,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↪dc=it,dc=ru
changetype: modify
add: member
member: cn=Robin.Department.Test2-20201202120913Z,cn=Robin.Department.Test2,cn=Robots,cn=Robot_
↪Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

### 9.6.3 Пользователи и группы пользователей

Для того чтобы пользователь AD мог получить доступ к каталогу IPA, нужно создать переопределение ID этого пользователя в каталоге IPA.

После этого у пользователя будет возможность выполнять различные операции над записями в каталоге IPA, доступные аутентифицированным пользователям.

Информация о переопределениях:

[https://www.freeipa.org/page/V4/Migrating\\_existing\\_environments\\_to\\_Trust](https://www.freeipa.org/page/V4/Migrating_existing_environments_to_Trust)

Пользователи AD, которые будут работать с системой ROBIN должны входить в группу Robin в AD. На одном из доменов леса AD создается локальная группа, в которую добавляются все необходимые пользователи леса AD.

Только для пользователей группы Robin создается переопределение в FreeIPA.

Чтобы добавить переопределение нужно знать полное доменное имя пользователя — `ad_user@ad_domain`

Получить информацию о пользователях группы Robin может администратор AD через оснастку «Пользователи и компьютеры»

Или под любым аутентифицированным пользователем AD ldap-запросом

Ввести команду:

```
ldapsearch -D "CN=user,CN=Users,DC=example,DC=com" -w secret -h ad_host.example.com -p 389 -b
↪ "CN=Robin,CN=Users,DC=example,DC=com" -s sub "(&(objectClass=group)(cn=Robin))"
```

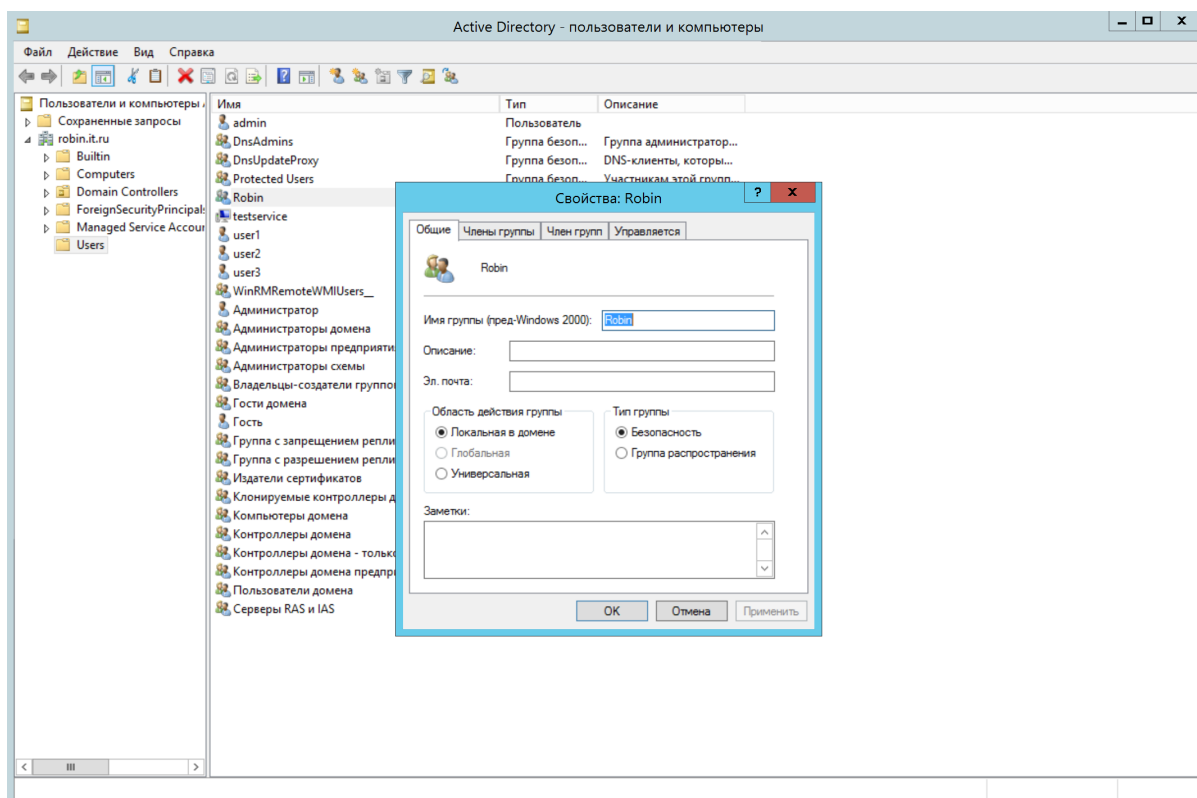
В ответе выведутся DN-ы всех пользователей, входящих в группу Robin, из которых можно составить полное доменное имя пользователя.

Создание переопределения ID пользователя из AD выполняется под администратором FreeIPA.

### Создание переопределения ID пользователя

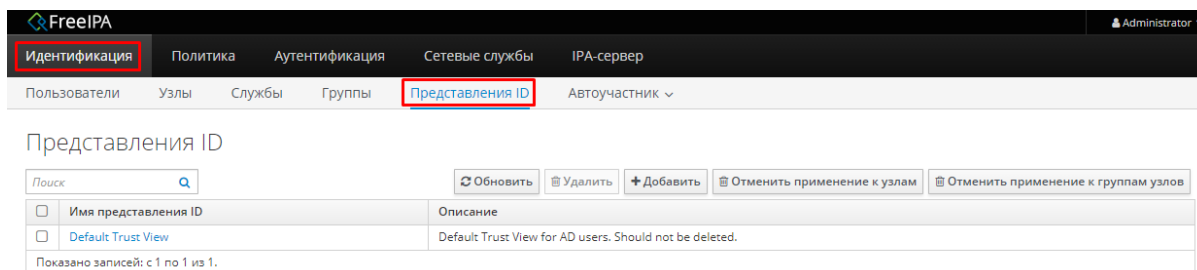
**Важно:** переопределения создаются на основном сервере FreeIPA

Создать переопределение можно через веб-интерфейс *FreeIPA* или инструмент командной строки.

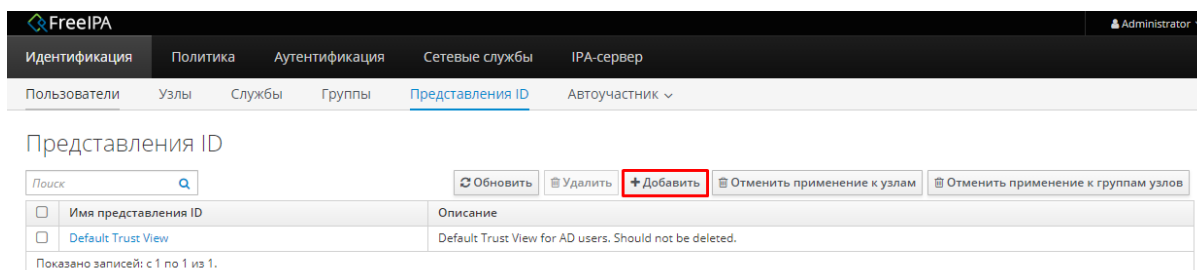


### Способ 1 - через веб-интерфейс FreeIPA

1. Перейти во вкладку *Идентификация - Представления ID*.



2. Если необходимо, то создать новое представление ID. Для этого нажать на кнопку *Добавить*.



3. Заполнить все поля и нажать кнопку *Добавить*.
4. Перейти к нужному представлению. Нажать кнопку *Добавить* для переопределения пользователя.
5. Заполнить все необходимые поля и нажать кнопку *Добавить*.



Добавить представление ID

✕

Имя \*

представления ID

Test View

Описание

Description

\* Обязательное поле

Добавить

Добавить и добавить ещё

Добавить и изменить

Отменить

FreeIPA

Administrator

Идентификация

Политика

Аутентификация

Сетевые службы

IPA-сервер

Пользователи

Узлы

Службы

Группы

Представления ID

Автоучастник

Представления ID

» Test View

Переопределения ID пользователя: Test View

Test View переопределяет:

Test View применяется к:

Пользователи

Группы пользователей

Узлы

Параметры

Поиск

Q

Обновить

Удалить

Добавить

☐

Пользователь, которого следует переопределить

Имя учётной записи пользователя

UID


Домашний каталог

Описание

Отсутствуют записи.

## Добавить переопределение ID пользователя



Пользователь, \*  
которого  
следует  
переопределить  


Имя учётной  
записи  
пользователя

GECOS

UID

ID группы

Сертификат

Открытый ключ  
SSH

Оболочка входа

Домашний  
каталог

Описание

\* Обязательное поле

**Добавить**

Добавить и добавить ещё

Добавить и изменить

Отменить

## Способ 2 - через командную строку с помощью команды `ipa idoverrideuser-add`

Для использования инструмента командной строки необходимо авторизоваться под административной учетной записью. Например:

```
kinit admin
```

Команда для добавления переопределения ID пользователя:

```
ipa idoverrideuser-add view user
```

где:

- *view* - представление, в которое будет добавлено переопределение пользователя.
- *user* - пользователь, для которого нужно добавить переопределение.

Например:

```
ipa idoverrideuser-add 'Default Trust View' user@robin.itbs.it.ru
```

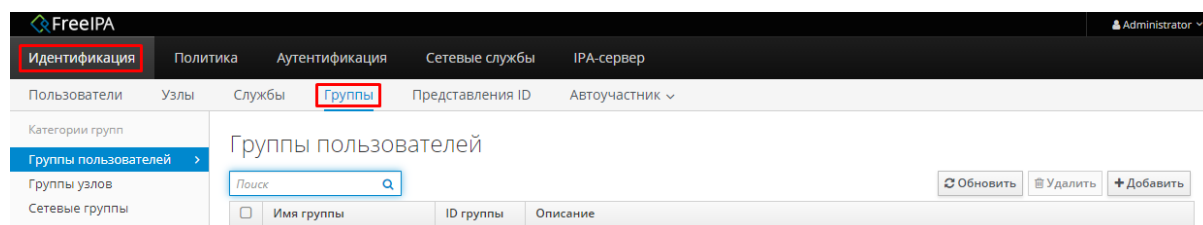
Переопределения ID пользователей можно добавлять в группы IPA для организации группового доступа к ресурсам IPA пользователям из AD.

## Создание группы пользователей

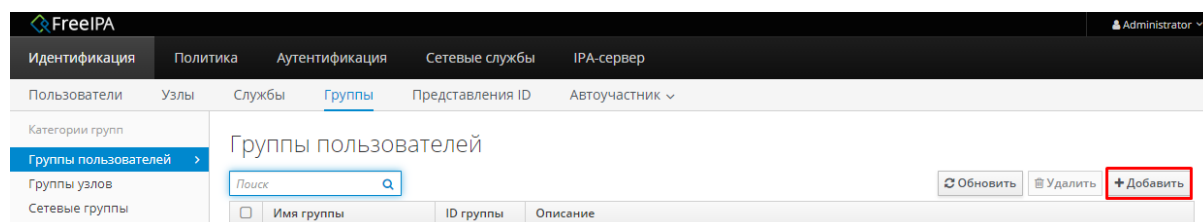
Группы создаются через веб-интерфейс *FreeIPA* или инструмент командной строки.

### Способ 1 - через веб-интерфейс *FreeIPA*

1. Перейти во вкладку *Идентификация - Группы*.



2. Нажать кнопку *Добавить*.



3. Заполнить необходимые поля и нажать кнопку *Добавить*.

### Способ 2 - через командную строку с помощью команды `ipa group-add`

Команда для добавления группы пользователей:

```
ipa group-add --desc=description groupName --nonposix
```

Добавить группу пользователей
✕

Имя группы \*

TestGroup

Описание

Description

Тип группы

☒ Не POSIX
☐ Внешний
☐ POSIX

ID группы

\* Обязательное поле

Добавить

Добавить и добавить ещё

Добавить и изменить

Отменить

где:

- *groupName* - название группы
- *description* - описание группы

Например:

```
ipa group-add --desc='Accounting robots' my_group --nonposix
```

### Добавление префикса артефактов в группу пользователей

Помимо имеющихся атрибутов в группу пользователей необходимо добавить атрибут, который будет использоваться в качестве префикса для создаваемых артефактов - пакетов с исходным кодом робота и пакетов с роботом.

В качестве префикса можно, например, указывать сокращенное наименование подразделения, в которое входят пользователи.

Правила именования префикса: допустимы буквы латинского алфавита в верхнем и нижнем регистрах, точка, тире и нижнее подчеркивание.

Группы пользователей расположены в поддереве

```
cn=groups,cn=accounts
```

Сначала необходимо добавить объектный класс к записи с группой путём добавления атрибута:

Атрибут	Значение
objectClass	nsCustomView

Затем добавить атрибут с префиксом артефактов:

Атрибут	Значение
nsDisplayName	Префикс, в названии которого допустимо использовать буквы латинского алфавита в верхнем и нижнем регистрах, точку, тире и нижнее подчёркивание

Пример добавления объектного класса *nsCustomView* в запись с группой *editors*:

```
dn: cn=editors,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: objectClass
objectClass: nsCustomView
```

Пример добавления атрибута с префиксом в запись с группой *editors*:

```
dn: cn=editors,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: nsDisplayName
nsDisplayName: Robin.Department
```

### Добавление переопределения ID пользователя в группу

Добавление переопределения в группу *FreeIPA* осуществляется добавлением атрибута *member* с DN переопределения в качестве значения. То, как это можно сделать, было описано ранее.

Группы пользователей расположены в поддереве

```
cn=groups,cn=accounts
```

Переопределения расположены в поддереве

```
viewRDN,cn=views,cn=accounts
```

Пример добавления переопределения пользователя *:SID:S-1-5-21-3493872076-3631449775-1555872641-29818* в группу *robin\_developers*:

```
dn: cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: member
member: ipaanchoruuid=:SID:S-1-5-21-3493872076-3631449775-1555872641-29818,cn=Default Trust_
View,cn=views,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru
```

### 9.6.4 Группы записей о пакетах с исходным кодом робота

Запись, являющаяся корневой для групп записей о пакетах с исходным кодом робота

```
cn=Source Groups,cn=Source Packages,cn=Packages,cn=ROBIN
```

При создании группы используются следующие объектные классы:

- groupOfNames
- bootableDevice

И атрибуты:

Атрибут	Многозначность	Описание
cn	Нет	Уникальное наименование группы
serialNumber	Нет	Идентификатор группы, UUID
description	Да, может быть на нескольких языках	Описание группы

В группу добавляются записи о пакетах с исходным кодом робота, лежащие в поддереве:

```
cn=Sources,cn=Source Packages,cn=Packages,cn=ROBIN
```

В этом случае в группу добавятся сразу все записи обо всех версиях пакета с исходным кодом робота.

Можно добавить в группу запись об отдельной версии пакета, лежащую в поддереве:

```
sourceRDN,cn=Sources,cn=Source Packages,cn=Packages,cn=ROBIN
```

Пример создания группы *Sources For Developers*:

```
dn: cn=Sources For Developers,cn=Source Groups,cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,
↪dc=itbs,dc=it,dc=ru
serialNumber: 840b60de-5a30-11eb-ae93-0242ac130002
description: Description
cn: Sources For Developers
objectClass: bootableDevice
objectClass: top
objectClass: groupOfNames
```

Пример добавления всех версий *Robin.Department.Test* в группу *Sources For Developers*:

```
dn: cn=Sources For Developers,cn=Source Groups,cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,
↪dc=itbs,dc=it,dc=ru
changetype: modify
add: member
member: cn=Robin.Department.Test,cn=Sources,cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,
↪dc=itbs,dc=it,dc=ru
```

Пример добавления отдельной версии *Robin.Department.Test-20201216082610Z* в группу *Sources For Developers*:

```
dn: cn=Sources For Developers,cn=Source Groups,cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,
↪dc=itbs,dc=it,dc=ru
changetype: modify
add: member
member: cn=Robin.Department.Test-20201216082610Z,cn=Robin.Department.Test,cn=Sources,cn=Source,
↪Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

### 9.6.5 Группы записей о пакетах с роботом

Запись, являющаяся корневой для групп записей о пакетах с роботом:

```
cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN
```

При создании группы используются следующие объектные классы:

- groupOfNames
- bootableDevice

И атрибуты:

Атрибут	Многозначность	Описание
cn	Нет	Уникальное наименование группы
serialNumber	Нет	Идентификатор группы, UUID
description	Да, может быть на нескольких языках	Описание группы

В группу добавляются записи о пакетах с роботом, лежащие в поддереве:

```
cn=Robots,cn=Robot Packages,cn=Packages,cn=ROBIN
```

В этом случае в группу добавятся сразу все записи обо всех версиях пакета с роботом.

Можно добавить в группу запись об отдельной версии пакета, лежащую в поддереве:

```
robotRDN,cn=Robots,cn=Robot Packages,cn=Packages,cn=ROBIN
```

Пример создания группы *Robots For Users*:

```
dn: cn=Robots For Users,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↪dc=it,dc=ru
serialNumber: 840b60de-5a30-11eb-ae93-0242ac130002
description: Description
cn: Robots For Users
objectClass: bootableDevice
objectClass: top
objectClass: groupOfNames
```

Пример добавления всех версий *Robin.Department.Test* в группу *Robots For Users*:

```
dn: cn=Robots For Users,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↪dc=it,dc=ru
changetype: modify
add: member
member: cn=Robin.Department.Test,cn=Robots,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,
↪dc=itbs,dc=it,dc=ru
```

Пример добавления отдельной версии *Robin.Department.Test-20201216082610Z* в группу *Robots For Users*:

```
dn: cn=Robots For Users,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,
↪dc=it,dc=ru
changetype: modify
add: member
member: cn=Robin.Department.Test-20201216082610Z,cn=Robin.Department.Test,cn=Robots,cn=Robot_
↪Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

### 9.6.6 Группы записей о хостах

Запись, являющаяся корневой для групп записей о хостах:

```
cn=Robot Host Groups,cn=Hosts,cn=ROBIN
```

При создании группы используются следующие объектные классы:

- groupOfNames

И атрибуты:

Атрибут	Многозначность	Описание
cn	Нет	Уникальное наименование группы
description	Да, может быть на нескольких языках	Описание группы

В группу добавляются записи о хостах, лежащие в поддереве:

```
cn=Hosts,cn=Hosts,cn=ROBIN
```

Пример создания группы *Test Host Group*:

```
dn: cn=Test Host Group,cn=Robot Host Groups,cn=Hosts,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
cn: Test Host Group
objectClass: groupOfNames
objectClass: top
```

Пример добавления записи о хосте *Test Host* в группу *Test Host Groups*:

```
dn: cn=Test Host Group,cn=Robot Host Groups,cn=Hosts,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: member
member: cn=Test Host,cn=Hosts,cn=Hosts,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

### 9.6.7 Запись о хранилище

Запись, являющаяся корневой для записей о хранилищах:

```
cn=Storages,cn=ROBIN
```

При создании записи о хранилище используются следующие объектные классы:

- nsConfig

И атрибуты:

Атрибут	Многочисленность	Описание
cn	Нет	Уникальное наименование хранилища
nsDefaultAcceptLanguage	Нет	Тип хранилища, указывающий на то, какие пакеты лежат в этом хранилище: с исходным кодом роботов (Source) или с роботами (Robot)
nsServerAddress	Нет	URL хранилища (например, URL сервиса RMS: <a href="https://rms-test.rpa-robin.ru/rpm/source">https://rms-test.rpa-robin.ru/rpm/source</a> или <a href="https://rms-test.rpa-robin.ru/rpm/robot">https://rms-test.rpa-robin.ru/rpm/robot</a> )
nsServerSecurity	Нет	Идентификатор хранилища, UUID - создаётся новый для каждого хранилища
description	Да, может быть на нескольких языках	Описание хранилища

На данный момент можно создать одну запись с хранилищем исходного кода роботов и одну запись с хранилищем роботов.

**Замечание:** Идентификатор хранилища и его DN нужно продублировать в `rpm.properties` (файл свойств Robin Package Manager, находится в каталоге Wildfly: `$wildfly_home/modules/rms2/main/properties`): `nsServerSecurity` - в свойстве `storage.source.id` (для записи о хранилище исходного кода роботов) и `storage.robot.id` (для записи о хранилище роботов); `dn` - в свойстве `storage.source.dn` (для записи о хранилище исходного кода роботов) и `storage.robot.dn` (для записи о хранилище роботов). Это будет исправлено в следующих обновлениях.

Пример создания хранилища *Storage*:

```
dn: cn=Storage,cn=Storages,cn=ROBIN,dc=ipa,dc=it,dc=ru
cn: Storage
```

(continues on next page)



(продолжение с предыдущей страницы)

```
nsServerSecurity: a5d9f4d0-a3e8-11eb-bcbc-0242ac130002
nsServerAddress: https://rms-test.rpa-robin.ru/rpm/source
description: Source storage
nsDefaultAcceptLanguage: Source
objectClass: nsConfig
objectClass: top
```

## 9.7 Управление полномочиями

### 9.7.1 ACI

Информация, определяющая правила разграничения доступа, представляется в директориях LDAP с помощью специальных атрибутов ACI (Access Control Information), которые могут быть добавлены к любому объекту директории. Атрибут ACI является множественным, то есть может принимать несколько значений, каждое из которых определяет самостоятельную инструкцию разрешающего или запрещающего характера.

Документация:

[https://www.freeipa.org/page/FreeIPAv2:Access\\_Control](https://www.freeipa.org/page/FreeIPAv2:Access_Control)

Назначение полномочий таких, как регулирование прав пользователей на запуск приложений, выполнение роботов и т. д. выполняется с помощью добавления атрибута ACI к различным записям в дереве LDAP.

Общий вид ACI:

```
aci: (target)(version 3.0;acl "name";permission bindRules;)
```

где:

- *target* - запись (набор записей), атрибуты (набор атрибутов), к которым регулируется доступ
- *name* - наименование ACI
- *permission* - права, которые выдаются (или, наоборот, запрещаются) по отношению к *target*, например. право на чтение
- *bindRules* - определяет пользователей, на которых распространяется ACI

### 9.7.2 Добавление ACI к записи

На примере рассмотрено добавление ACI

```
(targetattr = "*")(version 3.0; acl "Access to Studio"; allow (read,compare,search)(groupdn =
↪ "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru");)
```

к записи с DN

```
cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

#### Способ 1 - через Apache Directory Studio

Нужно перейти к нужной записи и вызвать контекстное меню для записи (правой кнопкой мыши)

1. Выбрать *New Attribute*
2. В открывшемся окне набрать «aci» в поле ввода *Attribute type* и нажать *Finish*

cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru

DN: cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru

Attribute Description	Value
<b>objectClass</b>	<b>nsApplication (structural)</b>
<b>objectClass</b>	<b>top (abstract)</b>
<b>cn</b>	<b>Studio</b>
<b>nsProductVersion</b>	<b>2.0.0</b>

**New Attribute...** Ctrl+Shift++

New Value Ctrl++

New Search... Ctrl+H

New Batch Operation...

## Attribute Type

Please enter or select the attribute type.

Attribute type:

☒ Show subschema attributes only

☒ Hide existing attributes

Preview:

cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru

DN: cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru

Attribute Description	Value
<b>objectClass</b>	<b>nsApplication (structural)</b>
<b>objectClass</b>	<b>top (abstract)</b>
<b>cn</b>	<b>Studio</b>
<b>nsProductVersion</b>	<b>2.0.0</b>
<b>aci</b>	(targetattr = *) (version 3.0; aci "Access to Studio"; allow (read,compare,search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru"))

3. Для атрибута *aci* добавить значение:

### Способ 2 - через командную строку с помощью команды `ldapmodify` (с использованием `ldif`-файла)

Допустим, есть следующий файл `example.ldif`

```
dn: cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Access to Studio"; allow (read,compare,
↪search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,
↪dc=ru");)
```

Тогда команда будет выглядеть следующим образом:

```
ldapmodify parameters -f path_to_file/example.ldif
```

*parameters* - другие параметры команды `ldapmodify` (хост для LDAP-сервера, порт, учётная запись и т. д.)

### Способ 3 - через командную строку с помощью команды `ldapmodify` (без использования `ldif`-файла)

Ввести команду:

```
ldapmodify parameters
```

*parameters* - другие параметры команды `ldapmodify` (хост для LDAP-сервера, порт, учётная запись и т. д.)

После этого ввести остальные данные:

```
dn: cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Access to Studio"; allow (read,compare,
↪search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,
↪dc=ru");)
```

## 9.7.3 Регулирование прав пользователей на запуск приложений

Право на запуск приложений выдаётся группе пользователей.

Таким образом, для того чтобы пользователь смог запустить какое-либо приложение, нужно:

1. Создать (если она ещё не создана) группу пользователей, которым будет доступно приложение на запуск
2. Добавить пользователя в эту группу
3. Выдать право группе пользователей на запуск приложения

Выдача права группе пользователей на запуск приложения выполняется с помощью добавления АСІ к записи с приложением в дереве LDAP.

Приложения, на запуск которых регулируются права - Studio, Robot Player, RMC.

DN в дереве LDAP для Studio:

```
cn=Studio,cn=Applications,cn=ROBIN
```

DN в дереве LDAP для Robot Player:

```
cn=Runner,cn=Applications,cn=ROBIN
```

DN в дереве LDAP для RMC:

```
cn=RMC,cn=Applications,cn=ROBIN
```

ACI имеет общий вид для всех приложений и выглядит следующим образом:

```
(targetattr = "*")(version 3.0; acl "Access to application"; allow (read,compare,
↪search)(groupdn = "ldap:///userGroupDN");)
```

где *userGroupDN* - DN группы пользователей, которой нужно выдать право на запуск приложения.

Пример добавления прав на запуск Studio пользователям группы *robin\_developers*:

```
dn: cn=Studio,cn=Applications,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Access to Studio"; allow (read,compare,
↪search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,
↪dc=ru");)
```

#### 9.7.4 Регулирование прав пользователей на модификацию исходного кода роботов

Доступ к исходному коду робота выполняется путём выдачи прав группе пользователей на группу записей с исходным кодом робота.

Таким образом, для того чтобы пользователь смог модифицировать какой-то исходный код робота, создать из него робота или новую версию исходного кода, нужно:

1. Создать (если она ещё не создана) группу пользователей, которым будет доступна модификация исходного кода роботов
2. Добавить пользователя в эту группу
3. Создать (если она ещё не создана) группу записей об исходном коде роботов, которых можно будет модифицировать определённым группам пользователей
4. Добавить запись об исходном коде робота в эту группу
5. Выдать право группе пользователей на модификацию исходного кода роботов

Выдача прав группе пользователей выполняется с помощью добавления ACI к записи, являющейся корневой для записей об исходном коде роботов и группы записей об исходном коде роботов:

```
cn=Source Packages,cn=Packages,cn=ROBIN
```

Общий вид ACI:

```
(targetattr!="serialNumber")(targetfilter="(|(&
↪(sourceGroupRDN)(objectClass=groupOfNames))(memberOf=sourceGroupDN))")(version 3.0; acl
↪"Allow to modify sources"; allow (read,search,compare)(groupdn="ldap:///userGroupDN");)
```

где:

- *sourceGroupRDN* - RDN группы записей об исходном коде роботов, на которую нужно выдать права

- *sourceGroupDN* - DN группы записей об исходном коде роботов, на которую нужно выдать права
- *userGroupDN* - DN группы пользователей, которой нужно выдать права

Пример добавления прав на модификацию исходного кода роботов из группы *Sources For Developers* пользователям группы *robin\_developers*:

```
dn: cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr!="serialNumber")(targetfilter="(|(&(cn=Sources For
↳ Developers)(objectClass=groupOfNames))(memberOf=cn=Sources For Developers,cn=Source Groups,
↳ cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru))")(version 3.0; acl
↳ "Allow to modify sources"; allow (read,search,compare)(groupdn="ldap:///cn=robin_developers,
↳ cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru");)
```

### 9.7.5 Регулирование прав пользователей на запуск роботов

Доступ к роботам выполняется путём выдачи прав группе пользователей на группу записей о пакетах с роботом.

Таким образом, для того чтобы пользователь смог запускать какого-то робота, нужно:

1. Создать (если она ещё не создана) группу пользователей, которым будет доступна группа роботов на запуск
2. Добавить пользователя в эту группу
3. Создать (если она ещё не создана) группу записей о пакете с роботами, которых можно будет запускать определённым группам пользователей
4. Добавить запись о пакете с роботом в эту группу
5. Выдать право группе пользователей на запуск роботов

Выдача прав группе пользователей выполняется с помощью добавления АСІ к записи, являющейся корневой для записей о пакетах с роботами и их группами:

```
cn=Robot Packages,cn=Packages,cn=ROBIN
```

Общий вид АСІ:

```
(targetattr!="serialNumber")(targetfilter="(|(&
↳ (robotGroupRDN)(objectClass=groupOfNames))(memberOf=robotGroupDN))")(version 3.0; acl "Allow
↳ to execute robots"; allow (read,search,compare)(groupdn="ldap:///userGroupDN");)
```

где:

- *robotGroupRDN* - RDN группы записей о пакетах роботов, на которую нужно выдать права
- *robotGroupDN* - DN группы записей о пакетах роботов, на которую нужно выдать права
- *userGroupDN* - DN группы пользователей, которой нужно выдать права

Пример добавления прав на запуск роботов из группы *Robots For Users* пользователям группы *robin\_users*:

```
dn: cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr!="serialNumber")(targetfilter="(|(&(cn=Robots For
↳ Users)(objectClass=groupOfNames))(memberOf=cn=Robots For Users,cn=Robot Groups,cn=Robot
↳ Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru))")(version 3.0; acl "Allow to
↳ execute robots"; allow (read,search,compare)(groupdn="ldap:///cn=robin_users,cn=groups,
↳ cn=accounts,dc=ipa,dc=itbs,dc=it,dc=ru");)
```

(continues on next page)

### 9.7.6 Регулирование прав пользователей на добавление записей о пакетах с исходным кодом робота в группы

Регулирование прав пользователей на добавление записей о пакетах с исходным кодом роботов в группы при публикации выполняется путём выдачи прав группе пользователей на группу записей с исходным кодом роботов.

Таким образом, для того чтобы пользователь смог при публикации пакета с исходным кодом робота добавить запись о нём в группу, нужно:

1. Создать (если она ещё не создана) группу пользователей, которым будет доступно добавление записи в группу
2. Добавить пользователя в эту группу
3. Создать (если она ещё не создана) группу записей о пакетах с исходным кодом робота, в которую пользователь сможет добавлять новые записи
4. Выдать право группе пользователей на добавление записей в группу

Выдача прав группе пользователей выполняется с помощью добавления ACI к записи с группой записей о пакетах с исходным кодом робота:

```
sourceGroupRDN,cn=Source Groups,cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,
↪dc=ru
```

где *sourceGroupRDN* - RDN группы записей об исходном коде робота

Общий вид ACI:

```
(targetattr = "*")(version 3.0; acl "Access to add to the group"; allow (read,compare,
↪search)(groupdn = "ldap:///userGroupDN");)
```

*userGroupDN* - DN группы пользователей, которой нужно выдать права

Пример выдачи прав на добавление в группу записей о пакетах с исходным кодом робота *Sources For Developers* группе пользователей *robin\_developers*:

```
dn: cn=Sources For Developers,cn=Source Groups,cn=Source Packages,cn=Packages,cn=ROBIN,dc=ipa,
↪dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Access to add to the group"; allow (read,compare,
↪search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,
↪dc=ru");)
```

### 9.7.7 Регулирование прав пользователей на добавление записей о пакетах с роботом в группы

Регулирование прав пользователей на добавление записей о пакетах с роботом в группы при публикации выполняется путём выдачи прав группе пользователей на группу роботов.

Таким образом, для того чтобы пользователь смог при публикации робота добавить запись о нём в группу, нужно:

1. Создать (если она ещё не создана) группу пользователей, которым будет доступно добавление записи в группу
2. Добавить пользователя в эту группу

3. Создать (если она ещё не создана) группу записей о пакетах с роботом, в которую пользователь сможет добавлять новые записи
4. Выдать право группе пользователей на добавление записей в группу

Выдача прав группе пользователей выполняется с помощью добавления ACI к записи с группой записей о пакетах с роботом:

```
robotGroupRDN,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
```

*robotGroupRDN* - RDN группы роботов

Общий вид ACI:

```
(targetattr = "*")(version 3.0; acl "Access to add to the group"; allow (read,compare,↵search)(groupdn = "ldap:///userGroupDN");)
```

*userGroupDN* - DN группы пользователей, которой нужно выдать права

Пример выдачи прав на добавление в группу роботов *Robots For Users* группе пользователей *robin\_developers*:

```
dn: cn=Robots For Users,cn=Robot Groups,cn=Robot Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,↵dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Access to add to the group"; allow (read,compare,↵search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,↵dc=ru");)
```

### 9.7.8 Регулирование прав пользователей на публикацию исходного кода роботов и роботов в хранилище

Регулирование прав пользователей на публикацию исходного кода роботов и роботов в хранилище выполняется путём выдачи прав группе пользователей на хранилище.

Таким образом, для того чтобы пользователь смог при публикации исходного кода робота или робота указать какое-либо хранилище, нужно:

1. Создать (если она ещё не создана) группу пользователей, которым будет доступна публикация в хранилище
2. Добавить пользователя в эту группу
3. Создать (если она ещё не создана) запись с хранилищем исходного кода роботов или роботов
4. Выдать право группе пользователей на публикацию в это хранилище

Выдача прав группе пользователей выполняется с помощью добавления ACI к записи с хранилищем:

```
storageRDN,cn=Storages,cn=ROBIN
```

*storageRDN* - RDN хранилища исходного кода роботов или роботов

Общий вид ACI:

```
(targetattr = "*")(version 3.0; acl "Access to push to the storage"; allow (read,compare,↵search)(groupdn = "ldap:///userGroupDN");)
```

*userGroupDN* - DN группы пользователей, которой нужно выдать право

Пример выдачи прав на публикацию в хранилище *Nexus Robot Storage* группе пользователей *robin\_developers*:

```
dn: cn=Nexus Robot Storage,cn=Storages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Access to push to the storage"; allow (read,compare,
↪search)(groupdn = "ldap:///cn=robin_developers,cn=groups,cn=accounts,dc=ipa,dc=itbs,dc=it,
↪dc=ru");)
```

### 9.7.9 Регулирование прав на запуск робота на хосте

Регулирование прав на запуск робота на хосте выполняется путём выдачи прав группе роботов выполняться на группе хостов.

Таким образом, для того чтобы робот мог запускаться на нужном хосте, нужно:

1. Создать (если она ещё не создана) группу записей о пакетах с роботом, которым можно будет выполняться на каком-то хосте
2. Добавить запись о пакете с роботом в эту группу
3. Создать (если она ещё не создана) группу хостов, на которой можно будет выполняться группе роботов
4. Добавить запись о хосте в эту группу
5. Выдать право группе роботов на выполнение на группе хостов

Выдача прав группе роботов выполняется с помощью добавления АСІ к записи, являющейся корневой для всех записей с хостами:

```
cn=Hosts,cn=Hosts,cn=ROBIN
```

Общий вид АСІ:

```
(targetattr="*)(targetfilter="(memberOf=hostGroupDN"))(version 3.0; acl "Allow to execute on_
↪host"; allow (read,search,compare)(groupdn="ldap:///robotGroupDN");)
```

где:

- *hostGroupDN* - DN группы хостов
- *robotGroupDN* - DN группы записей о пакетах с роботом

Пример выдачи прав группе роботов *Robots For Execute* выполняться на группе хостов *Development Host Group*:

```
dn: cn=Hosts,cn=Hosts,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru
changetype: modify
add: aci
aci: (targetattr="*)(targetfilter="(memberOf=cn=Development Host Group,cn=Robot Host Groups,
↪cn=Hosts,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru"))(version 3.0; acl "Allow to execute on host";
↪allow (read,search,compare)(groupdn="ldap:///cn=Robots For Execute,cn=Robot Groups,cn=Robot_
↪Packages,cn=Packages,cn=ROBIN,dc=ipa,dc=itbs,dc=it,dc=ru");)
```



---

## Серверные компоненты ROBIN Platform

---

### 10.1 Аннотация

Документ описывает архитектуру серверной части Платформы ROBIN (далее - Изделие).

### 10.2 Общие сведения

#### 10.2.1 Назначение изделия

Платформа ROBIN представляет собой комплекс программных средств, предназначенных для решения задач в области роботизации бизнес-процессов, а именно:

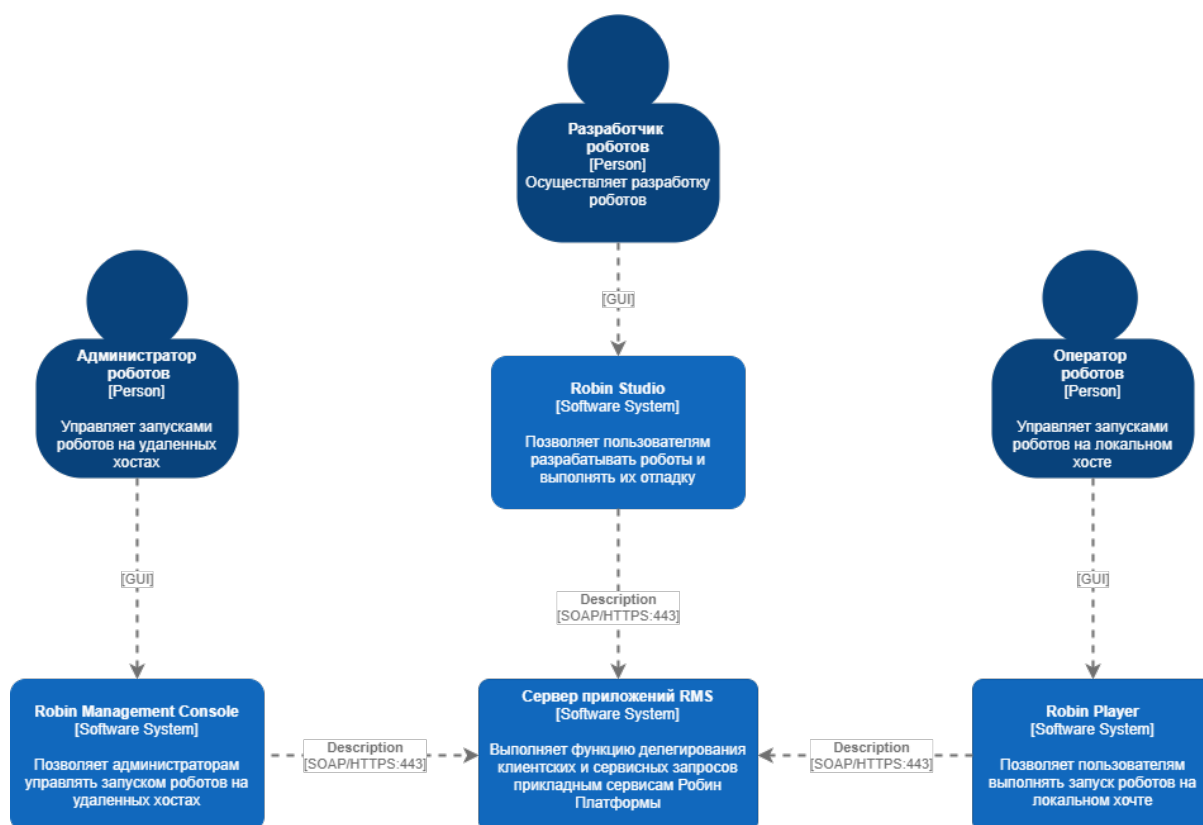
- осуществлять разработку новых программных роботов и действий,
- выполнять сценарии готовых программных роботов для решения задач Предприятия Заказчика.

Серверная часть Платформы ROBIN представляет собой набор программных серверных компонент, располагающихся в зависимости от исполнения на одном или нескольких физических серверах как с образованием кластеров так и без образования кластеров.

Серверные компоненты Платформы ROBIN реализуют следующий функционал:

- хранение информации о пользователях и правах доступа пользователей: к клиентским приложениям, к артефактам роботов и действий;
- хранение артефактов роботов, включая исходники роботов, действий и документации по действиям;
- хранение логов с оперативной информацией о запусках и остановках роботов;
- хранение логов серверов приложений;
- аутентификация клиентских приложений,
- аутентификация и авторизация пользователей в процессе их работы с приложениями;
- публикацию новых роботов и действий - сохранение их в хранилище роботов;

- транспорт сообщений (логов) с оперативной информации о состоянии роботов и агентов в хранилище логов;
- транспорт сообщений (команд управления) между консолью RMS и прикладными службами систем исполнения роботов;
- управление запусками роботов по расписанию или по требованию оператора;
- обеспечение безопасной передачи данных через очереди программного брокера сообщений.



## 10.2.2 Исполнение изделия

Изделие имеет следующие варианты исполнений:

- demo - все серверные компоненты размещаются на одном сервере,
- бескластерное исполнение - все серверные компоненты разворачиваются на отдельных серверах без образования кластеров,
- кластерное исполнение - в этой конфигурации каждый серверный компонент может быть развернут в кластере для обеспечения отказоустойчивости и возможности дальнейшего масштабирования системы.

## 10.2.3 Требования к программному обеспечению

### Требования к общесистемному программному обеспечению

Для обеспечения работы серверных компонент Платформы ROBIN на серверах должно быть установлено следующее общесистемное программное обеспечение (ОПО) - всё ПО с открытым кодом:

- операционные системы (ОС) - одна из:
  1. CentOS Linux release 7.6.1810 (Core) и выше,

2. Red Hat Enterprise Linux 7,
3. ОС Astra Linux Common Edition (релиз «Орёл»),
  - Java 8 JDK,
  - WildFly (JBoss Application Server) 20 и выше – сервер приложений Java EE,
  - PostgreSQL 10 x86\_64 и выше - система управления базами данных,
  - FreeIPA 4.8 и выше - интегрированное решение по идентификации и аутентификации пользователей, задания политик доступа к объектам доступа,
  - Sonatype Nexus 3.30 и выше – сервис менеджера пакетов, предназначенный для хранения артефактов роботов и действий,
  - Rabbit MQ 3.8 и выше – программный брокер сообщений на основе стандарта AMQP, обеспечивает асинхронную передачу сообщений между прикладными службами Платформы ROBIN,
  - HAProxy 2.1.3 и выше - серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов,
  - ELK-сервисы (Elasticsearch, Logstash и Kibana) – стек сервисов, предназначенный для сбора и хранения логов от прикладных сервисов Платформы ROBIN и от серверов приложений, а также для выполнения клиентских запросов с целью построения отчетов.

### Состав специального программного обеспечения (СПО)

В состав специального программного обеспечения (СПО) серверной части Платформы ROBIN входят:

- Robin Management Server (RMS) – централизованная служба, выступающая в роли оркестратора. Предоставляет программные интерфейсы (API), посредством которых с ним взаимодействуют клиентские приложения. Работает под управлением сервера приложений WildFly (JBoss Application Server).
- Robin Package Manager – сервис, предназначенный для выполнения клиентских запросов к хранилищу роботов. Работает под управлением сервера приложений WildFly (JBoss Application Server).
- Robin RDP-Manager – сервис, работающий на отдельном сервере. Предназначен для открытия и удержания RDP-сессий. Открытие RDP-сессии на удаленном хосте необходимо для робота, который работает с приложением, имеющим графический интерфейс, эмулируя работу пользователя.
- База данных для серверных приложений RMS – система хранения мета-информации о роботах и расписаниях их запуска. Работает под управлением СУБД PostgreSQL 10+.
- Каталоги сервера аутентификации и авторизации пользователей. Работает под управлением FreeIPA.
- Хранилище пакетов – система хранения артефактов роботов и действий. Работает под управлением Sonatype Nexus.
- Хранилище логов предназначено для хранения логов. Работает под управлением ELK-сервисов (Elasticsearch, Logstash и Kibana).
- Система транспорта сообщений - работает под управлением RabbitMQ.

В состав Платформы ROBIN на стороне конечного пользователя входят следующие программные компоненты:

- Robin Management Console (RMC) – клиентское приложение, предоставляющее пользователю графический интерфейс для централизованного управления запусками роботов на удаленных хостах,

- Robin Studio – клиентское приложение, предоставляющее пользователю графический интерфейс для разработки программных роботов,
  - Robin Player – клиентское приложение, предоставляющее пользователю графический интерфейс для управления запусками программных роботов на исполнение сценария,
  - Robin Agent – прикладной сервис, расположенный на стороне клиента (Robin Studio или Robin Player):
1. принимает через очереди сообщений программного брокера клиентские запросы со стороны Robin Studio, Robin Player и Robin Management Console,
  2. транслирует клиентские запросы системе исполнения сценариев программных роботов Robin Executor,
  3. отправляет в RMS через очереди сообщений программного брокера информационные сообщения о событиях в системе исполнения сценариев программных роботов Robin Executor (старт/останов робота), а также сообщения об ошибках.

## 10.2.4 Требования к вычислительной технике

### Аппаратная конфигурация в исполнении demo

Для целей оценки функциональности все серверные компоненты Платформы ROBIN могут быть размещены на одном сервере со следующими минимальными требованиями к аппаратному обеспечению:

Характеристика	Минимальные требования
Количество потоков процессора (vCPU - поток ядра)	8 vCPU
Количество оперативной памяти (RAM)	16 Gb
Емкость дисковой подсистемы (HDD)	250 Gb
Производительность дисковой подсистемы	125 IOPS

### Аппаратная конфигурация в бескластерном исполнении

В этой конфигурации все серверные компоненты Платформы ROBIN разворачиваются на отдельных серверах со следующими минимальными требованиями к аппаратному обеспечению:

### Аппаратная конфигурация в кластерном исполнении

В этой конфигурации каждый серверный компонент Платформы ROBIN развернут в кластере для обеспечения отказоустойчивости и возможности дальнейшего масштабирования системы:

Пример аппаратной конфигурации в кластерном исполнении:

Минимальная аппаратная конфигурация в исполнении cluster может быть такой:

	Сервер приложений	Число серверов	CPU	RAM, Gb	HDD, Gb
1	WildFly	1	4	8	50
2	PostgreSQL	1	4	4	50
3	FreelPA	1	4	6	50
4	Sonatype Nexus	1	4	6	100
5	Rabbit MQ	1	4	4	50
6	Elasticsearch	1	4	4	150
7	Logstash	1	2	2	50
8	Kibana	1	4	4	50
9	Robin RDP-Manager	1	2	4	30
	<b>Всего:</b>	<b>9</b>	<b>32</b>	<b>42</b>	<b>580</b>

Характеристика	Минимальные требования
Количество потоков процессора (CPU pCore- физическое ядро) (vCPU - поток ядра. 1pCore = 2 vCPU)	4 vCPU
Количество оперативной памяти (RAM)	2-6 Gb
Емкость дисковой подсистемы (HDD)	50-200 Gb
Производительность дисковой подсистемы	125 IOPS

Ресурс	CPU	RAM, Gb	HDD, Gb
Elasticsearch 1	4	4	200
Elasticsearch 2	4	4	200
Elasticsearch 3	4	4	200
FreeIPA LDAP 1	4	6	20
FreeIPA LDAP 2	4	6	20
HAProxy backup	4	4	20
HAProxy master	4	4	20
Kibana 1	4	4	20
Kibana 2	4	4	20
Logstash 1	2	2	20
Logstash 2	2	2	20
Nexus 1	4	6	100
Nexus 2	4	6	100
Nexus 3	4	6	100
PostgreSQL master	4	4	50
PostgreSQL slave	4	4	50
RabbitMQ 1	4	4	50
RabbitMQ 2	4	4	50
RabbitMQ 3	4	4	50
RDP Manager Server 1	2	4	20
RDP Manager Server 2	2	4	20
WildFly AppNode 1	4	6	50
WildFly AppNode 2	4	6	50
WildFly AppNode 3	4	6	50
<b>Всего:</b>	<b>88</b>	<b>108</b>	<b>1500</b>

### 10.2.5 Требования к персоналу (системному администратору)

Системный администратор должен иметь как минимум высшее техническое образование.

В перечень задач, выполняемых системным администратором, должны входить:

1. Задача поддержания работоспособности технических средств, на которых устанавливаются инфраструктурные сервисы (сервера приложений) и серверные компоненты Платформы ROBIN.
2. Задача установки (инсталляции) и поддержания работоспособности операционной системы и серверов приложений.
3. Задача установки (инсталляции), настройки и поддержания работоспособности специального программного обеспечения Платформы ROBIN:
  - серверных приложений RMS (на WildFly):
  - RDP Manager - самостоятельное java серверное приложение,
  - базы данных для серверных приложений RMS (на PostgreSQL),
  - базы данных (каталоги) сервера аутентификации и авторизации пользователей (на FreeIPA),
  - хранилища роботов (на Nexus),
  - системы очередей (на RabbitMQ),
  - коллекторов логов (filebeat) и фильтров логов (ELK-сервис).

Права, порядок доступа, а также результаты работы системного администратора в отношении серверных компонент RMS определяет и контролирует администратор безопасности на протяжении всех этапов эксплуатации RMS - от ввода в эксплуатацию до вывода из эксплуатации - в строгом соответствии с требованиями руководящих документов Заказчика по информационной безопасности.

### 10.2.6 Специальные требования к среде исполнения изделия для осуществления безопасности функционирования

Для безопасного выполнения Платформой ROBIN своих функций к среде функционирования предъявляются следующие требования в части касающейся безопасности и отсутствия влияния сторонних средств:

1. Не должно происходить запуска ОС в однопользовательском режиме, так как в этом случае невозможно гарантировать неизменность конфигурации системы или выполнения задач по ее обслуживанию. Кроме того, должны быть выполнены условия по запрету выполнения команд для осуществления перехода в однопользовательский режим (например «telinit 1»).
2. Всю системную конфигурацию среды исполнения должен проводить только администратор безопасности, за которым документально закреплена ответственность в части политики информационной безопасности.
3. Доступ прикладных служб Платформы ROBIN к открытым портам сервера приложений RMS должен быть защищен от несанкционированного доступа межсетевым экраном, а все сетевые соединения защищены посредством использования протокола TLS или SSL.

### 10.2.7 Пользователи Платформы ROBIN

Конечные пользователи Платформы ROBIN представлены тремя группами:

- группа администраторов - пользователи этой группы используют клиентское приложение Robin Management Console (RMC) для управления удаленными запусками роботов,
- разработчики роботов - пользователи этой группы используют клиентское приложение Robin Studio,

- операторы роботов - пользователи этой группы используют клиентское приложение Robin Player для локального запуска роботов.

Доступ пользователей к клиентским приложениям регулируется через их вхождение в группы, а также правилами разграничения доступа (ПРД), созданными в FreeIPA. Правила разграничения доступа определяют кто (пользователь), откуда (хост), куда (хост) и к какому приложению (службе) имеет право обращаться. За создание учетных записей пользователей, групп и правил разграничения доступа отвечает администратор Сервера идентификации пользователей FreeIPA. Права, порядок доступа, а также результаты работы администратор Сервера идентификации пользователей FreeIPA определяет и контролирует администратор безопасности на протяжении всех этапов эксплуатации Изделия - от ввода в эксплуатацию до вывода из эксплуатации - в строгом соответствии с требованиями руководящих документов Заказчика по информационной безопасности.

## 10.3 Функциональная архитектура серверной части Платформы ROBIN

### 10.3.1 Функциональные возможности Robin Management Server

#### Назначение RMS

RMS предназначен для обработки запросов от пользователей клиентских приложений - RMC, Robin Studio, Robin Player.

#### Функциональные возможности сервисов RMS

Сам RMS состоит из двух J2EE-приложений, выполняемых в WildFly как сервисы.

Сервисы RMS предоставляют конечным пользователям обширные функциональные возможности:

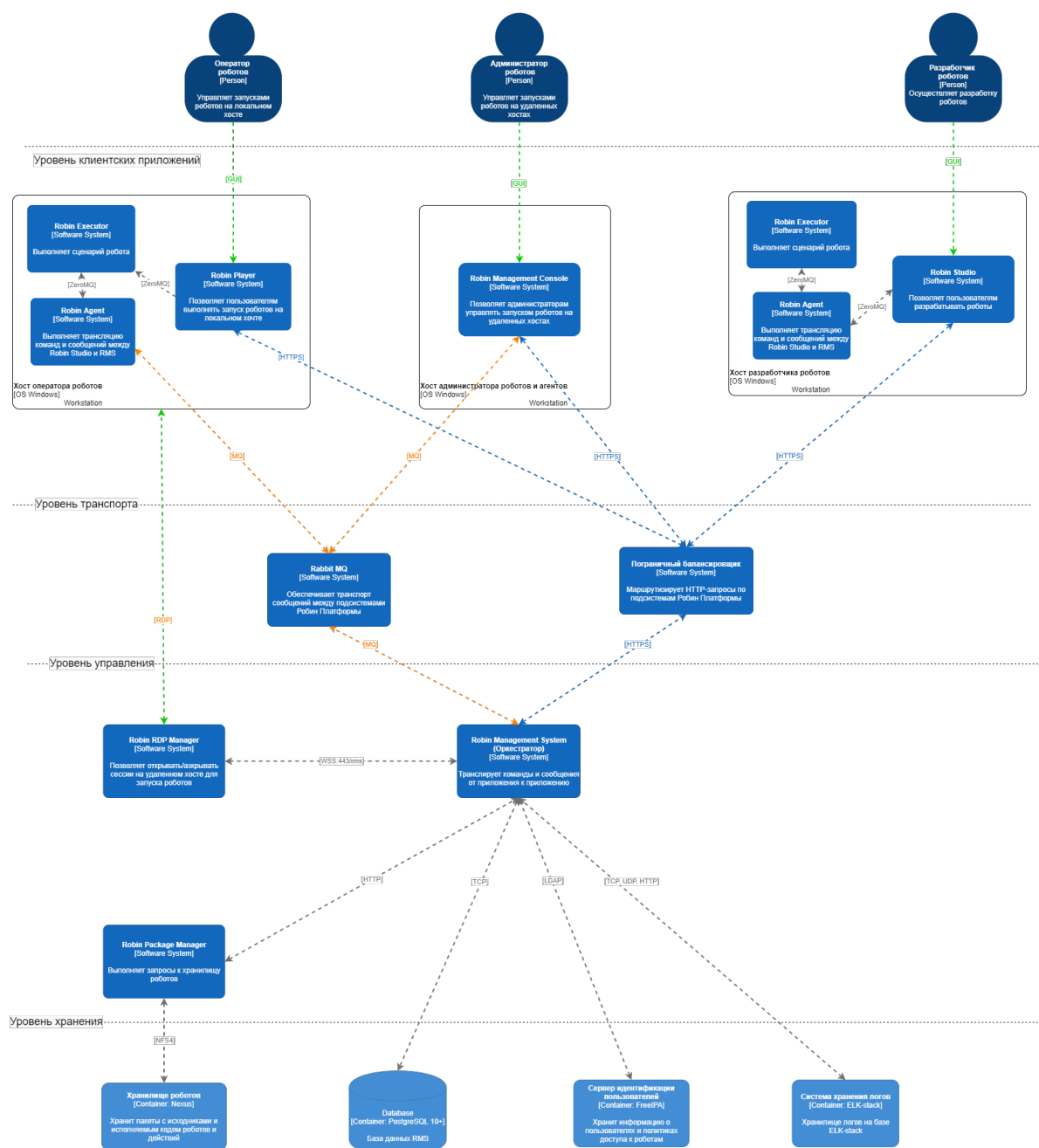
- выполняют аутентификацию клиентских приложений RMC, Robin Player и Robin Studio,
- позволяют пользователям клиентских приложений:
  1. управлять запусками и остановами роботов,
  2. просматривать журналы выполнения роботов,
  3. получать оперативную информацию о состоянии удаленных агентов, зарегистрированных в RMS (запущен/не запущен),
  4. получать оперативную информацию о текущем состоянии роботов (запущен/завершен),
  5. управлять запусками и остановами роботов на удаленных агентах в том числе составлять для них расписания запуска,
  6. просматривать историю запуска роботов на удаленных агентах,
  7. создавать/удалять учетную запись пользователя, предназначенную для открытия RDP-сессии на удаленном хосте,
  8. активировать/деактивировать учетную запись пользователя, предназначенную для открытия RDP-сессии на удаленном хосте.

#### Аутентификация и авторизация пользователей и клиентских приложений

В корпоративной и облачной версиях продукта RMS выполняет аутентификацию и авторизацию как пользователей так и клиентских приложений. Причем аутентификация пользователя возможна только после прохождения аутентификации клиентского приложения.

#### Аутентификация клиентских приложений





В корпоративной и облачной версиях продукта присутствуют приложения (RMC, Robin Studio, Robin Player, Robin Agent), взаимодействующие с серверной частью. Процедура аутентификации клиентских приложений реализована по технологии единого входа (Single sign-on, SSO), которая позволяет этим приложениям безопасно взаимодействовать с серверными компонентами (RMS, RPM, HTTP-проxy) Платформы ROBIN без необходимости повторной аутентификации, используя SSO-токен. Сервис RMS выполняет роль SSO-сервера. Во время установления сессии (handshake), подтверждается взаимная аутентичность клиентского приложения и сервера RMS.

После прохождения аутентификации клиентское приложение получает SSO-токен (JWT-токен), который клиентское приложение вместе с временной меткой указывает в заголовке всех своих запросов к серверным приложениям. Все серверные приложения (RMS, RPM, HTTP-проxy) при каждом запросе валидируют токен клиентского приложения с помощью публичного ключа сервиса RMS, проверяют время действия токена и временную метку.

В облачной версии клиентское приложение (RMC, Robin Studio, Robin Player) допускается к процедуре аутентификации только в случае валидного токена компании. Проверка токена компании выполняется до процедуры аутентификации приложения. Токен компании получает ответственное лицо компании в своем личном кабинете после регистрации на Портале МаркетПлейса Облачной Фабрики Роботизации. Администратор указывает токен компании во время установки каждого клиентского приложения. На клиентских компьютерах токен компании хранится в защищенной (зашифрованной) конфигурации приложения. Токен компании может быть помечен как не валидный администратором компании и вместо него может быть выпущен новый. После того как текущий токен будет помечен как не валидный, воспользоваться им будет уже нельзя и потребуются процедура его замены.

### Аутентификация пользователей

Если аутентификация приложения прошла успешно, то далее выполняется аутентификация пользователей.

Возможны следующие режимы аутентификации пользователя:

- пользователь клиентского приложения является пользователем корпоративного домена AD, рабочая станция, на которой выполняется запуск клиентского приложения, включена в тот же домен, а между сервером идентификации (FreeIPA) и контроллером домена AD установлены доверенные отношения;
- пользователь клиентского приложения является пользователем домена FreeIPA.

Если пользователь клиентского приложения является пользователем корпоративного домена и при этом между сервером идентификации (FreeIPA) и контроллером домена AD установлены доверенные отношения, то:

- Аутентификация пользователя инициируется средствами клиентского приложения в момент его запуска. При этом пользователю не надо вводить свои учетные данные (логин/пароль),
- Аутентификацию пользователя выполняет контроллер AD,
- Авторизацию пользователя выполняет сервер каталогов FreeIPA.

Если пользователь клиентского приложения не является пользователем AD (является только пользователем домена FreeIPA), то клиентское приложение должно запросить у пользователя логин и пароль и выполнить процедуру аутентификации пользователя.

### Авторизация пользователей

Возможны следующие режимы авторизации пользователя:

- Если пользователь клиентского приложения не является пользователем AD (является только пользователем домена FreeIPA), то аутентификация и авторизация пользователя выполняется сервером идентификации FreeIPA.
- Если пользователь клиентского приложения является пользователем корпоративного домена и между сервером идентификации (FreeIPA) и контроллером домена AD установлены доверенные отношения, то аутентификация пользователя выполняется контроллером домена AD, а авторизация пользователя выполняется сервером каталогов FreeIPA.

В этом случае вся информация о пользователе хранится на контроллере AD. Список AD групп, в которые входит пользователь, подтягиваются из AD, а его роли находятся в каталогах FreeIPA по их маппингу на группы AD.

### 10.3.2 Функциональные возможности Менеджера RDP-сессий

Менеджер RDP-сессий позволяет по запросу авторизованного пользователя RMC установить RDP-сессию на удаленном хосте для запуска роботов.

---

**Важно:** Для работы робота на хосте необходимо, чтобы предварительно был запущен Robin Agent или Robin Player. Если эти приложения не запущены, то сначала требуется установить RDP-сессию на удаленном хосте и запустить их. Если для этих приложений установлен режим автозагрузки, то приложения стартуют автоматически во время открытия пользовательской RDP-сессии.

---

Для запуска робота необходимо, чтобы был запущен Robin Player (процесс робота будет запущен приложением Robin Player из под учетной записи текущего пользователя RDP сессии). В качестве параметров Менеджер RDP-сессий получает от RMC параметры учетной записи пользователя, от имени которого будет установлена RDP сессия и будет выполнен робот.

### 10.3.3 Функциональные возможности серверного приложения Robin Package Manager (RPM)

Robin Package Manager предоставляет возможность аутентифицированным приложениям выполнять 2 вида запросов:

- запрос на скачивание пакетов (проектов роботов, готовых роботов);
- запрос на публикацию пакетов (проектов роботов, готовых роботов);

Оба типа запросов обрабатывает серверное приложение RPM (менеджер пакетов) оркестратора.

Взаимодействие пользователя с RPM происходит по протоколу HTTPS (порт 443 RMS) и требует предварительной аутентификации приложения.

#### Публикация пакетов «проекта робота» и «робота»

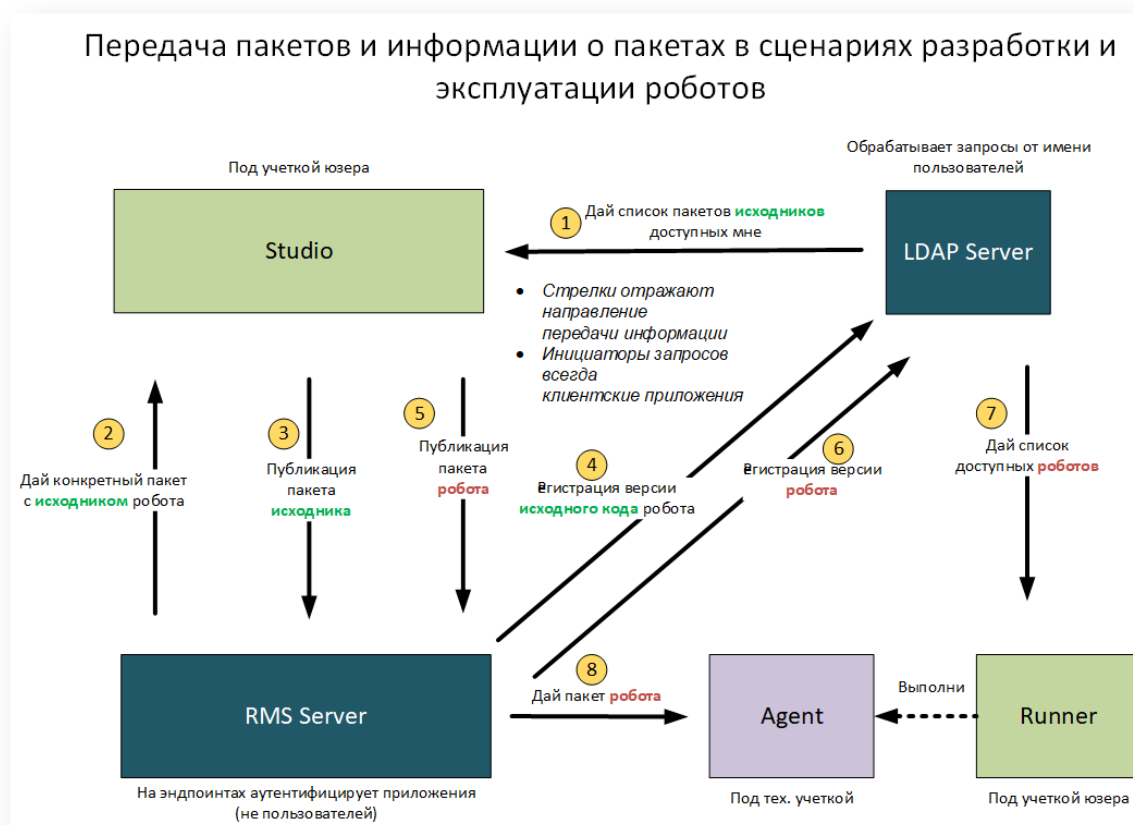
Приложение RPM, при получении от студии пакета выполняет следующие действия:

- получает из пакета информацию о контракте и версии;
- проверяет что в хранилище нет уже опубликованного пакета с таким контрактом и версией;
- генерирует уникальный идентификатор пакета, под которым информация о нем будет размещена в каталоге FreeIPA;
- создаёт новую запись о пакете в каталоге FreeIPA;
- добавляет пакет в группы, если они были указаны в запросе;
- размещает пакет на хранение на сервере Nexus.

### 10.3.4 Функциональные возможности брокера сообщений

В рамках Платформы ROBIN клиентские и серверные приложения могут обмениваться сообщениями. К таким приложениям относятся:

- Robin-агенты (может быть несколько),
- RMC - консоль управления роботами (может быть несколько),
- RDPM - менеджер RDP-сессий (может быть несколько),



- RMS - сервер управления (только один).

Поддержку транспорта сообщений и механизма очередей сообщений выполняет программный брокер сообщений RabbitMQ.

### Защита сообщений

Сообщения передаются через очередь в зашифрованном виде. Каждый экземпляр приложения получает собственный ключ алгоритма симметричного шифрования (AES-256) во время в ходе процедуры аутентификации приложения.

## 10.4 Информационная архитектура серверной части Платформы ROBIN

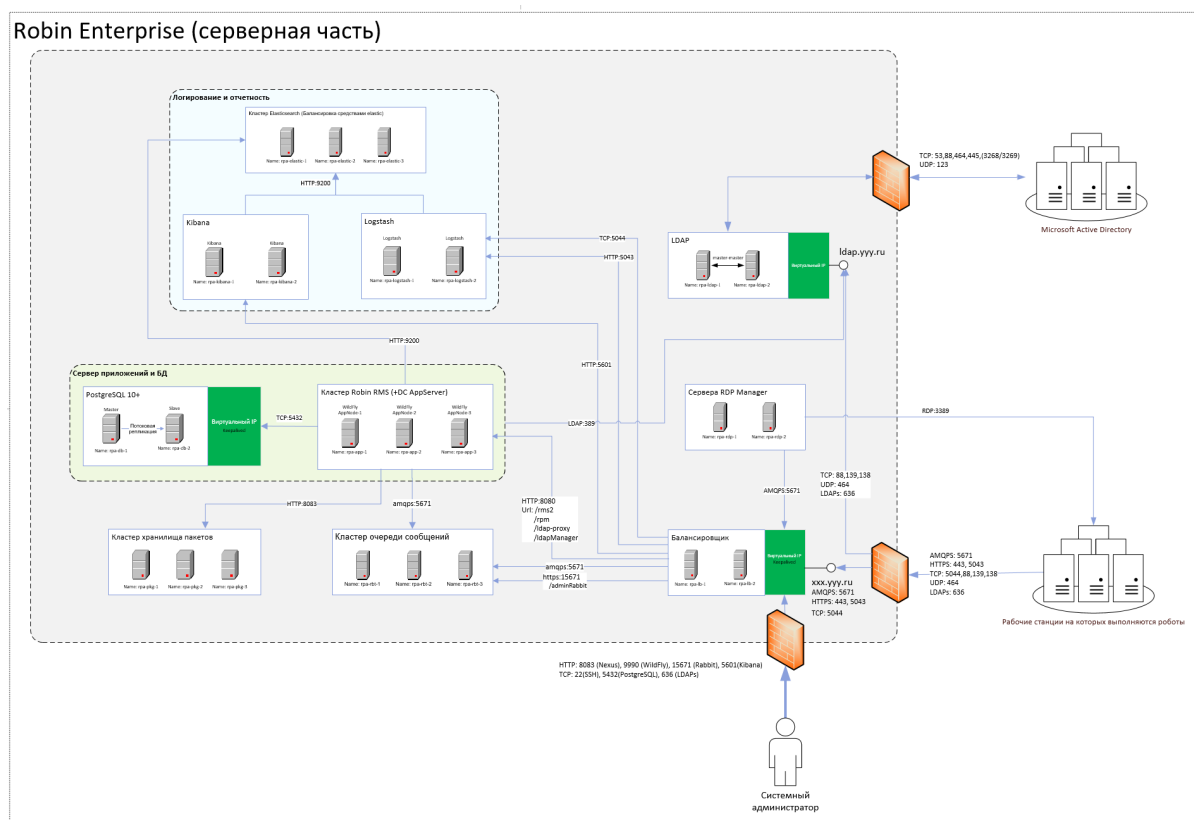
В состав Платформы ROBIN на стороне серверной части входят следующие информационные активы (данные):

- активы, хранимые в хранилище роботов: пакеты с исходниками роботов, зависимости (библиотеки), необходимые для выполнения исполняемого кода программных роботов;
- активы, хранимые на сервере каталогов FreeIPA: учетные записи пользователей (для варианта развертывания без AD), учетная информация о роботах, права доступа пользователей к роботам и приложениям, группы пользователей и роботов;
- активы, хранимые в базе данных RMS: структура пакетов роботов, действий и зависимостей; расписаний запуска роботов;

- конфигурационные активы: конфигурационные файлы с настройками всех прикладных и инфраструктурных сервисов Платформы ROBIN; схема конфигурации очередей сообщений, поддерживаемых RabbitMQ;
- активы, хранимые в системе хранения логов: логи от прикладных сервисов Платформы ROBIN.

## 10.5 Инфраструктурная схема серверной части с компонентами Платформы ROBIN 1.3 и 2.0

На схеме представлен пример физической инфраструктуры для размещения компонент серверной части Платформы 1.3 и 2.0. Сервера (и кластеры) на схеме объединены в функциональные группы:



### 10.5.1 Группа серверов «Сервер приложений и БД»

На данной группе серверов разворачиваются серверные компоненты RMS:

- Robin Management Server (RMS) – централизованная служба, выступающая в роли оркестратора,
- Robin Package Manager – сервис, предназначенный для выполнения клиентских и сервисных запросов к хранилищу роботов,
- база данных для серверных приложений RMS (на PostgreSQL):

OrchestratorDS - источник данных для хранения структуры пакетов роботов, действий и зависимостей. QuartzDS - источник данных для хранения расписаний запуска роботов.

## Кластер Robin RMS (+DC AppServer)

Кластер сервера приложений. Минимум три узла. На каждом узле установлен экземпляр сервера приложений и контролера домена серверов приложений. Один из узлов выделен под роль балансировщика нагрузки кластера. Именно на него поступает трафик с «граничного» балансировщика (функциональная группа «Балансировщик» на схеме), который направляет трафик (основываясь на части пути в запросе) на балансирующий узел кластера, и преобразует трафик из https в http.

## БД PostgreSQL 10+

База данных для серверных приложений RMS (на PostgreSQL):

- OrchestratorDS - источник данных для хранения структуры пакетов роботов, действий и зависимостей.
- QuartzDS - источник данных для хранения расписаний запуска роботов.

### 10.5.2 Группа серверов «Хранилище пакетов»

Все артефакты в системе (как создаваемые при помощи студии, так и привносимые в обновлениях вендором) представлены в виде пакетов:

- исходный код робота;
- робот, подготовленный к эксплуатации;
- действие, используемое в работе;
- библиотеки с бинарным кодом, которые необходимы действию - зависимости.

Все пакеты хранятся в специально предназначенном для этого хранилище - отказоустойчивое файловое хранилище пакетов.

Сервера хранилища пакетов «спрятаны» за сервером приложений. Клиентские приложения не имеют непосредственного доступа к хранилищу пакетов. Запросы поступают на сервер приложений, сервер проверяет право на скачивание (либо публикацию) пакета и в случае одобрения проксирует запрос на кластер хранилища.

Состав функционального блока «Хранилище пакетов»:

- Кластер Nexus Package Store - масштабируемое решение, позволяющее выполнять запросы на загрузку/выгрузку пакетов разных форматов, принятых для платформ .net, java, python.
- Балансировщик кластера хранилища пакетов распределяет запросы к Nexus Package Store по узлам кластера.
- Отказоустойчивое файловое хранилище пакетов. Как правило на предприятиях уже существуют отказоустойчивые файловые хранилища. Если таковое имеется и оно доступно по протоколу NFS, его можно использовать для хранения пакетов.

### 10.5.3 Группа серверов «Кластер очереди сообщений»

Основным недостатком серверного решения версии 1.0 была невозможность горизонтального масштабирования серверной части из-за использования синхронной передачи данных между агентами и сервером RMS по типу «запрос-ответ». В новой версии системы используется Rabbit MQ, который обеспечивает асинхронную передачу данных между прикладными службами Робин Платформы через масштабируемые очереди сообщений.

### 10.5.4 Группа серверов «LDAP»

С целью обеспечения ряда требований (возможность создания атрибутов объектов после ввода системы в эксплуатацию и обеспечения регулирования прав на уровне как групп, так и единичных записей), было принято решение использовать готовое решение корпоративного справочника. Для хранения записей о объектах и субъектах системы роботизации (и распределения полномочий) было спроектировано дерево объектов, хранимых в LDAP, и запросы доступа к данному дереву.

### 10.5.5 Группа серверов «Логирование и отчетность»

Для централизованного сбора и хранения логов (а так же построения отчетов в web интерфейсе) было выбрано существующее решение ELK (система из приложений Elasticsearch, Logstash, Kibana). На всех хостах системы, на которых работают клиентские приложения платформы ROBIN, устанавливается программное обеспечение агента ELK - Filebeat, в чью функции входит мониторинг и сбор лог-сообщений из лог-файлов и пересылке их в Elasticsearch или Logstash для индексирования.

Состав функциональной группы «Логирование и отчетность»:

- Сервер Logstash.
- Кластер Elasticsearch с балансировкой средствами elastic.
- Сервер Kibana.
- Агенты Filebeat на стороне клиентских приложений.

### 10.5.6 Группа серверов «Сервера RDP Manager»

Кластер этой группы серверов обеспечивает масштабируемое решение для Менеджера RDP-сессий.

### 10.5.7 Группа серверов «Балансировщик»

Функции балансировщика в Платформы ROBIN (в поставке с одним балансировщиком) включают:

- Маршрутизация сетевых соединений из внешней сети к компонентам в серверной подсети.
- Маршрутизация сетевых соединений между компонентам внутри серверной подсети.
- Балансировка нагрузки, когда целью сетевого соединения является кластер серверов.

### 10.5.8 Таблица с описанием информационных потоков между серверами

В таблице представлено описание потоков между компонентами Платформы ROBIN с указанием:

- источника данных;
- получателя данных;
- инициатора запроса на получение данных;
- состава данных;
- протокола и порта соединения;
- периодичности передачи данных (по команде пользователя, в соответствии со сценарием или по расписанию).

Но- мер свя- зи на схе- ме	Источ- ник дан- ных	Полу- чатель данных	Инициатор запроса на получение данных	Состав данных	Про- то- кол и порт сер- вера	Пери- одич- ность запроса
1	LDAP сервер (FreeIPA)	Robin Studio, Robin Robot, Robin Management Console (RMC)	Robin Studio, Robin Robot, Robin Management Console (RMC)	Список доступных пользо- вателю приложения робо- тов и проектов роботов	LDAP 389	По ко- манде пользо- вателя
2	Сервер прило- жений, Сервер отчетов, Сервер очереди сообщений	Robin Studio, Robin Robot, Robin Management Console (RMC), Админи- страторы приложений	Robin Studio, Robin Robot, Robin Management Console (RMC), Админи- страторы приложений	Пакеты содержащие робо- тов и проекты роботов. Информация о ходе вы- полнения роботов. HTML страницы с отчетами о вы- полнении роботов. HTML страницы административ- ных интерфейсов серверов.	HTTPS 443 AMQP 5672	По ко- манде пользо- вателя в соот- ветствии со сценари- ем
3	Сервер прило- жений	Через Балансиров- щик Robin Studio, Robin Management Console (RMC), Админи- страторы приложений	Robin Studio, Robin Management Console (RMC), Админи- страторы приложений	Пакеты содержащие робо- тов и проекты роботов. HTML страницы с отчета- ми о выполнении роботов. HTML страницы админи- стративных интерфейсов серверов.	HTTPS 8080	По ко- манде пользо- вателя в соот- ветствии со сценари- ем
4	База данных	Сервер при- ложений	Сервер при- ложений	Информация о составе и ресурсных потребностях роботов	TCP 5432	По ко- манде пользо- вателя в соот- ветствии со сценари- ем
5	Храни- лище пакетов	Сервер при- ложений	Сервер при- ложений	Пакеты содержащие робо- тов и проекты роботов	HTTP 8083	По ко- манде пользо- вателя в соот- ветствии со сценари- ем
6	Сервер прило- жений	LDAP сервер	Сервер при- ложений	Информация о пакетах робо- тов и пакетах проектов роботов	LDAP 389	По ко- манде пользо- вателя
7	Robin Robot	Через Балансиров- щик. Сервер очереди сообщений	Robin Robot	Информация о ходе вы- полнения роботов	AMQP 5672	В соот- ветствии со сценари- ем
8	Сервер прило- жений	Через Балансировщик.	Сервер при- ложений	Команды управления ра- ботой роботов	AMQP 5672	В соот- ветствии со сценари- ем. По расписа- нию. По
10.5.2.0	Инфраструктурная схема серверной части с компонентами Платформы ROBIN 1.8-070					По



## 10.5.9 Установка и настройка серверных компонент

Установка серверных компонент выполняется автоматически с помощью Ansible-скриптов (сценариев).

Ansible-скрипты:

1. выполняют установку и конфигурирование следующего серверного ПО:
  - PostgreSQL 10+ - система управления базами данных,
  - ELK-сервис (Elasticsearch, Logstash и Kibana) – сервис, предназначенный для сбора и хранения логов от прикладных сервисов Платформы Робин, а также для выполнения клиентских запросов с целью построения отчетов.
  - FreeIPA - интегрированное решение по идентификации и аутентификации пользователей, задания политик доступа,
  - Sonatype Nexus – сервер пакетного менеджера,
  - Rabbit MQ – брокер сообщений, обеспечивает передачу сообщений между прикладными службами Робин Платформы.
  - WildFly (JBoss Application Server) – сервер приложений Java EE,
2. загружают серверные компоненты ( j2ee-приложения) на сервер WildFly
3. создают структуру БД в PostgreSQL,
4. загружают структуру каталогов в FreeIPA.

Настройка конфигурационных параметров серверных компонент выполняется через конфигурационные файлы, которые являются неотъемлемой частью скриптов. Конфигурационные файлы задают:

- задают целевые хосты, на которых будут устанавливаться приложения. Хосты могут быть объединены в группы;
- параметры (пароли и ip) доступа к БД;
- параметры WildFly (версии, пароль, ip, пути к исполняемым файлам);
- параметры доступа к системе сообщений Rabbit;
- параметрами распределенного кэша infinispn;
- и другие параметры служб серверных компонент.

Запуск всех сервисов выполняется автоматически сразу же после завершения процедуры установки серверных компонент.

## 11.1 Конфигурирование Logstash после установки

**Примечание:** Предполагается что установка осуществлялась в соответствии с базовым руководством производителя.

Изменения вносятся в единственный файл `/etc/logstash/conf.d/base.conf`:

- Секция **input**:

```

1 input {
2   beats {
3     type=> "fromFileBeat"
4     port => 5044 # Любо любой желаемый порт
5     client_inactivity_timeout => 360
6   }
7 }
```

- Секция **filter**:

```

1 filter {
2   if [fields][log_type] == "robot" {
3     grok {
4       match => {
5         "message" => "%{TIMESTAMP_ISO8601:syslog_timestamp}%{SPACE}%{WORD:syslog_level}%{SPACE}
↪ %{HOSTNAME:syslog_hostname}%{SPACE}%{NOTSPACE:syslog_source}%{SPACE}%{NOTSPACE:syslog_pid}%
↪ %{SPACE}%{GREEDYDATA:json_part}"
6       }
7     }
8     mutate {
9       add_field => {
10        "[robot_log][message]" => "%{message}"
11        "[robot_log][creation_date_time]" => "%{syslog_timestamp}"
12        "[robot_log][level]" => "%{syslog_level}"
13        "[robot_log][hostname]" => "%{syslog_hostname}"
14        "[robot_log][source]" => "%{syslog_source}"
15        "[robot_log][pid]" => "%{syslog_pid}"

```

(continues on next page)

(продолжение с предыдущей страницы)

```

16     }
17   }
18   json {
19     source => "json_part"
20     target => "[robot_log][json]"
21   }
22   mutate {
23     remove_field => ["syslog_timestamp", "syslog_level", "syslog_hostname", "syslog_source",
↪ "syslog_pid", "message", "json_part"]
24   }
25 }
26 if [fields][log_type] == "application" {
27   grok{
28     match => {
29       "message" => "%{CISCOTIMESTAMP:timestamp}%{SPACE}%{HOSTNAME:hostname}%{SPACE}%
↪ %{NOTSPACE:debug_level}%{SPACE}%{GREEDYDATA:json_part}"
30     }
31   }
32   json{
33     source => "json_part"
34     target => "json_payload"
35   }
36   mutate {
37     remove_field => ["json_part", "message"]
38     gsub => ["debug_level", "<", ""]
39     gsub => ["debug_level", ">", ""]
40   }
41 }
42 }

```

- Секция **output**:

```

1 output {
2   if [fields][log_type] == "robot" {
3     elasticsearch {
4       hosts => ["<указываем IP:порт или fqdn:порт elastic>"]
5       index => "robot-logs-%{+YYYY.MM.dd}"
6     }
7   }
8   if [fields][log_type] == "application" {
9     elasticsearch {
10      hosts => ["<указываем IP:порт или fqdn:порт elastic>"]
11      index => "application-logs-%{+YYYY.MM.dd}"
12    }
13  }
14 }

```

**Примечание:** В зависимости от режима развертывания в секции `hosts` указывается либо `ip/fqdn:port` сервера `elastic` либо входная точка координатора кластера

## 11.2 Конфигурирование filebeat для Windows

**Примечание:** На примере версии **7.9.2**

Далее в описании предполагается путь установки `c:\filebeat`

- В зависимости от разрядности ОС копируем содержимое архива **filebeat-7.9.2-windows-x86.zip** или **filebeat-7.9.2-windows-x86\_64.zip** в директорию **c:\filebeat**
- Создаем или редактируем файл **c:\filebeat\filebeat.yml** следующего содержания:

```

1 filebeat.inputs:
2 # секция сбора логов выполнения роботов
3 - type: log
4   enabled: true
5   start_position: beginning
6   paths:
7     - C:\Users\*\AppData\Local\Robin\RobotLogs\**
8   fields:
9     {log_type: robot}
10 # секция сбора логов компонентов Robin Platform
11 - type: log
12   enabled: true
13   start_position: beginning
14   paths:
15     - C:\Users\*\AppData\Local\Robin\PlayerLogs\*.log
16     - C:\Users\*\AppData\Local\Robin\RobotLogs\*.log
17     # Устаревшее, не используется
18     # - C:\Users\*\AppData\Local\Temp\Robin Platform\2.0.0\Runner\Logs\*.log
19     - C:\Users\*\AppData\Local\Temp\Robin Platform\2.0.0\Studio\Logs\*.log
20     - C:\Users\*\AppData\Roaming\Robin Platform\2.0.0\Agent\Logs\*.log
21     - C:\Users\*\AppData\Roaming\Robin Platform\2.0.0\Executor\Logs\*.log
22     # Пользовательский каталог агента, запущенного как служба от системной учетной записи
23     - C:\Windows\System32\config\systemprofile\AppData\Roaming\Robin Platform\2.0.
24     ↪0\Agent\Logs\*.log
25   fields:
26     {log_type: application}
27 output.logstash:
28   hosts: ["<ip адрес/fqdn Logstash сервера : порт>"]
29 processors:
30   - add_host_metadata:
31     when not contains tags: forwarded
32     # netinfo.enabled: false # (enable/disable netinfo)

```

**Предупреждение:** В данной конфигурации подразумевается типовая установка без переназначения пути к пользовательским профилям. В ином случае в конфигурации следует заменить:

- **C:\Users** - на соответствующий путь к пользовательским профилям.
- **C:\Windows\System32\config\systemprofile** - на соответствующий путь к профилю системной учетной записи

- выполняем powershell-скрипт с правами администратора

```

cd c:/filebeat
c:/filebeat> ./install-service-filebeat.ps1

```

**Примечание:** В случае получения сообщения о запрете выполнения Powershell-скриптов можно временно разрешить запуск PowerShell скриптов выполнив:

### Set-ExecutionPolicy Bypass

- Альтернативный вариант установки службы через sc.exe:

```
sc.exe create filebeat binPath="c:/filebeat/filebeat.exe --environment=windows_  
↪service -c c:/filebeat/filebeat.yml --path.home c:/filebeat/ --path.data c:/  
↪filebeat/data --path.logs c:/filebeat/logs -E logging.files.redirect_stderr=true_  
↪start=auto"  
sc.exe start filebeat
```

---

**Примечание:** Требуется запуск с правами администратора

---

## 12.1 Ключевые термины и понятия ROBIN Platform

Глоссарий содержит перечень основных терминов и понятий, используемых при работе с ROBIN Platform.

Термин	Термин (англ.яз)	Действие
Debug	Debug	Журналирование моментов вызова «крупных» операций: старт/остановка потока, запрос пользователя и т.п.
Web Sockets	Web Sockets	Технология, позволяющая создавать интерактивное соединение для обмена сообщениями в онлайн-режиме. Соединение создается в сети между сервером и клиентом (браузером)
Win-элемент	Win-element	Ресурс, в котором записан уникальный идентификатор, определяющий конкретный элемент (кнопка/поле/т.п) в окне Windows.
Win-рекордер	Win-recorder	Инструмент, записывающий действия пользователя при работе с десктопными приложениями. Он автоматически создает необходимые win-элементы в ресурсах и добавляет соответствующие действия на флору.
Win-шпион	Win Spy	Инструмент, позволяющий определить уникальные свойства элемента десктопного приложения.
Web-элемент	Web-element	Ресурс, хранящий информацию об уникальном элементе на web-странице (текстовое поле/кнопка/блок текста).
Web-рекордер	Web-recorder	Инструмент для записи действий пользователя в браузере.
Web-шпион	Web Spy	Инструмент, который записывает XPath компонента страницы в переменную.
XPath	XPath	Путь до элемента, определяемый порядком следования элементов внутри друг друга
RDP	RDP	Remote Desktop Protocol – протокол для обеспечения удаленной работы с сервером
RMC	RMC	Robin Management Console – программное приложение, предназначенное для контроля за выполнением сценариев роботов
Блок-схема	Block diagram	Упорядоченный алгоритм или процесс, выраженный в форме блоков, соединенных между собой линиями. Каждая блок-схема имеет начало и конец.
Готовый робот	Finished robot	Робот, полностью собранный и исправно функционирующий. Имеет расширение.rpr (robin program robot), файлы с данным расширением являются исполняемыми.
Проект робота	Robot project	Схема Робота, собранная Пользователем в рабочем пространстве. Имеет расширение .rfs (robot file scheme). Файлы данного формата можно открыть в ROBIN Studio для просмотра и/или дальнейшего редактирования
Контекст	Context	Уникальный идентификатор сеанса, обозначающий 1 экземпляр запущенного приложения.
Переменная	Variable	Атрибут, который может изменять свое значение в процессе выполнения роботом отработки схемы. Первоначальное значение переменной может быть как пустым, так и иметь какое-либо значение (как правило, числовое).
Проект робота	Robot project	Схема Робота, собранная Пользователем в рабочем пространстве. Имеет расширение .rfs (robot file scheme). Файлы данного формата можно открыть в ROBIN Studio для просмотра и/или дальнейшего редактирования
Платформа Robin	Robin Platform	Комплекс программных средств, предназначенных для автоматизации рутинных операций, выполняемых сотрудниками различных организаций, и включающих в себя ПО «Robin Studio», «Robin Orchestrator», «Robin Player», «Robin Management Console».
ОФР	OFR	Облачный B2B сервис, предоставляющий онлайн-площадку для взаимодействия Компаний – контрагентов для разработки, тестирования и эксплуатации программных роботов
Родительская папка	Parent folder	Папка, содержащая внутри себя другие папки
Рабочее пространство	Work space	Область в ROBIN Studio, куда переносятся действия для выстраивания схемы робота.
12.1. Ключевые термины и понятия ROBIN Platform		377
Ресурс	Resource	Это информация, которую робот использует в ходе выполнения процесса.
Файл	File	Внешний ресурс, необходимый в работе некоторых действий, который можно

## 13.1 Release Notes 2.7.0

### Что нового

#### *ROBIN Studio*

- Добавлено уведомление при запуске студии с выключенным ROBIN Agent.
- Исправлена ошибка, возникавшая при наличии нескольких действий/ресурсов, название которых отличалось только регистром букв. Теперь использование таких ресурсов, действий не приводит к завершению работы студии.
- Изменено отображение значений Коллекций и Словарей в панели лога и в поле значения. Теперь указывается количество элементов коллекции/словаря, а не полный список значений
- Добавлена возможность переключения языка приложения между Английским и Русским. Сейчас установка английского языка влияет только на названия и описания действий во встроеной документации. Переключатель расположен в окне «Настройки программы» и изменение вступает в силу после перезапуска студии.

#### *ROBIN Player*

- Улучшена работа с параметрами запуска. Теперь окно выбора параметров отображается при любом запуске робота и позволяет указать уровень логирования для роботов без входных параметров.

#### *Группа действий «Браузеры»*

- При захвате элемента в браузере с помощью webSpy при зажатом Ctrl не производит реального клика по элементу на странице
- Добавлен драйвер для Google Chrome версии 97 для webSpy
- Для действий «Клик», «Отправить текст», «Ожидать состояния элемента», «Проверить наличие элемента», «Получить текст» добавлена возможность работы в режиме совместимости с IE 5 и выше
- Для действия «Открыть браузер» улучшена обработка ошибок, теперь при возникновении ошибки в действии, в логе пишется более точная ошибка, особенно если вызвана настройками браузера

#### *Группа действий «Файловая система»*



- Исправлена ошибка работы действия «Разархивировать» при указании длинного пути до архива.

#### Группа действий «1С»

- Добавлена новая группа действий, предназначенная для работы с приложением 1С через Odata, включающая:

действие «Подключение к 1С», позволяющее установить соединение с сервером 1С действие «Получение объекта» действие «Получение списка объектов» действие «Изменение полей объекта»

#### Группа действий «JSON»

- Добавлена новая группа действий, предназначенная для работы с файлами и текстом в формате \*.json, включающая:
1. действие «Открыть», позволяющее открыть существующий файл .json для дальнейшей работы
  2. действие «Очистить», позволяющее стереть информацию используемую в контексте
  3. действие «Получить коллекцию значений», позволяющее получить коллекцию всех имеющихся значений
  4. действие «Получить коллекцию ключей», позволяющее получить коллекцию всех имеющихся ключей
  5. действие «Получить количество элементов», позволяющее получить количество всех имеющихся пар Ключ:Значение
  6. действие «Преобразовать в словарь», позволяющее получить словарь наполненный парами Ключ:Значение используемыми в json
  7. действие «Преобразовать в строку», позволяющее получить содержимое json в виде строки
  8. действие «Преобразовать словарь в JSON», позволяющее использовать данные из словаря для формирования контекста json
  9. действие «Преобразовать строку в JSON», позволяющее использовать данные из строки для формирования контекста json
  10. действие «Содержит ключ», позволяющее проверить наличие указанного ключа в json
  11. действие «Создать JSON», позволяющее создать новый контекст json, не содержащий информации
  12. действие «Сохранить в файл», позволяющее сохранить значение из контекста json в указанный файл с расширением .json
  13. действие «Сравнить», позволяющее проверить, являются ли 2 объекта json одинаковыми
  14. действие «Удалить значение», позволяющее удалить значение у конкретного ключа
  15. действие «Установить значение», позволяющее установить значение ключа

#### Группа действий «Базы данных»

- Исправлена ошибка в действии «Пользовательское подключение». Теперь действие корректно работает при указании любого значения в параметре «Уровень изоляции»

#### Группа действий «Интеграции»

- Исправлена ошибка в действии «Отправить запрос» подгруппы «REST». Теперь действие корректно работает при использовании кириллицы в тексте запроса

#### Группа действий «Безопасность»

- Добавлено новое действие «Получить пароль из удаленного хранилища», которое позволяет взять сохраненный пароль, например, от ИС, с которой взаимодействует робот, из централизованного хранилища на стороне ROBIN Orchestrator

#### Группа действий «Сервисы Google»

- Изменено название группы, старое название - Google spreadsheets. Теперь группа включает в себя работу с сервисами Google:
- 1. новое действие «Подключиться к Google Docs», позволяющее установить подключение к проекту с настроенным API google docs
- 2. новое действие «Создать новый документ», позволяющее создавать новые пустые документы
- 3. новое действие «Подключиться к Google Drive», позволяющее установить подключение к проекту с настроенным API google drive
- 4. новое действие «Загрузить файл», позволяющее загрузить указанный файл на drive.google.com
- 5. новое действие «Получить список файлов», позволяющее получить список файлов, к которым имеет доступ пользователь в drive.google.com
- 6. новое действие «Скачать файл», позволяющее скачать файл по заданным id и дополнительным свойствам
- 7. Изменен процесс сохранения сессий работы с сервисами. При первом подключении и выборе пользователя, который будет использоваться для работы с действиями, файл с данными подключения сохраняется в папке Robin Platform2.0.0Runs%user% для каждого из сервисов используется собственная папка: GoogleDocsAuth, GoogleSpreadsheetsAuth, GoogleDriveAuth.

#### *Группа действий «Окна Java»*

- Добавлена новая версия JavaSpy позволяющее работать с приложениями запущенными с помощью JVM x32

#### *Группа действий «Базовые действия»*

- Исправлена проблема, приводившая к преждевременному завершению робота после выполнения действия TryCatch на языке java

#### *Известные проблемы:*

- При работе с WinSpy рамка выделяющая элемент мешает его захвату шпионом. Для корректной работы со шпионом рекомендуется выполнять многократные клики по элементу с зажатим Ctrl и одновременным перемещением курсора, чтоб он оставался в границах элемента. Повторять, пока элемент не будет захвачен. Возможен захват неправильных элементов, если после клика на элемент быстро переместить курсор в другую область.

Проблема будет решена в одном из ближайших релизов.

## 13.2 Release Notes 2.6.2

### **Что нового**

#### *ROBIN Studio*

- Добавлена утилита SapSpy позволяющая просматривать структуру и получать селекторы элементов в окнах SAPGUI, которые можно использовать в действиях группы «SAP».

Утилита пока запускается независимо от ROBIN Studio: /Robin Platform/2.0.0/Utils/SapSpy.exe

#### *Группа действий «SAP»*

Добавлена новая группа действий, включающая действия для взаимодействия с SAP GUI.

## 13.3 Release Notes 2.6.1

### **Что нового**

#### *ROBIN Studio*

- Добавлена утилита JavaSpy позволяющая просматривать структуру и получать селекторы элементов в окнах java-приложений, которые можно использовать в действиях группы «Java окна».

Для использования необходимо включить фичу «jabswitch -enable» в jre пакете java. Утилита пока запускается независимо от ROBIN Studio: /Robin Platform/2.0.0/Utils/JavaSpy.exe.

- Добавлена возможность использовать Веб-шпион с браузером Internet Explorer.
- При экспорте проекта робота, проект автоматически сохраняется, экспортируется актуальная версия.

#### *Группа действий «Java окна»*

Добавлена новая группа действий, включающая действия для взаимодействия с приложениями написанными на java. Для работы действий этой группы используется возможность «Java Access Bridge». Указать элемент приложения возможно с помощью JavaSpy.

#### *Группа действий «Браузеры»*

Исправлены возможные ошибки в действиях при работе в режиме совместимости с устаревшими версиями браузера. Изменения затронули действия:

- «Отправить клик»
- «Отправить текст»
- «Ждать состояния элемента»
- «Проверить состояние элемента»
- «Прокрутить до элемента»
- «Открыть браузер»

#### *Группа действий «Базовые действия»*

- Исправлено поведение в действии «Пока» в версиях java и .net. Теперь действие корректно работает при указании условия с помощью выражения.
- Исправлено поведение в действии «Цикл с постусловием» в версиях java и .net. Теперь действие корректно работает при указании условия с помощью выражения.

## 13.4 Release Notes 2.6.0

### **Что нового**

#### *ROBIN Studio*

- В действие «Поиск» верхней панели добавлена вкладка «Контексты». На ней отображаются все действия, создающие контексты и группирует все действия, которые работают в рамках этого контекста. Также доступна возможность поиска и перехода к выбранному действию.
- Теперь можно добавлять описания к шагам робота. У любого действия на схеме робота можно заполнить параметр «Описание» блока «Информация». Указанный текст будет отображаться как описание действия на схеме.
- Улучшена работа с большими сценариями, где в сценарии робота много шагов. Оптимизирована скорость загрузки действий и навигации. При открытии сохраненного робота группы могут быть свернуты.

#### *ROBIN Player*

- Добавлена возможность указания параметров при запуске роботов из ROBIN Player.
- Изменен механизм указания параметров для запусков по расписанию. Теперь отображается интерфейс заполнения параметров вместо указания xml файла. Для существующих роботов

использующих параметры может потребоваться заново указать параметры с помощью нового механизма.

- В настройках рассылки логов о успешном/неуспешном завершении робота, добавлена возможность переключиться на вариант отправки почты для почтового сервера без авторизации.

#### *Группа действий «Браузеры»*

- Улучшена работа действий «Отправить клик», «Отправить текст», «Проверка наличия элемента», «Ожидать состояния элемента» для предотвращения ложного срабатывания при проверке доступности элемента.
- В действии «Открыть браузер» добавлена поддержка работы с браузером Edge на движке chromium.
- Исправлена ошибка в действии «Открыть браузер» с браузерами Internet Explorer, Firefox, Yandex, в результате которой неверно определялась версия браузера, а также возникала ошибка при запуске указанных браузеров.

Обращаем внимание! При наличии проблем с работой браузера Internet Explorer необходимо проверить правильность настройки браузера по инструкции: Конфигурация Internet Explorer (IE)).  
 \* Для действия Открыть браузер добавлены драйверы для работы с браузерами Yandex, Firefox, Internet Explorer, Edge. \* Добавлено действие «Сохранить изображение», которое позволяет сохранить изображение со страницы в папку. Работает только с элементами <img>.

#### *Группа действий «Почта»*

- В действии «Настройки подключения» изменена обязательность параметров «Логин» и «Пароль» и добавлена возможность подключения к SMTP серверу без авторизации. Для этого необходимо оставить параметры «Логин» и «Пароль» не заполненными.

#### *Группа действий «Имитация устройств»*

- Добавлены действия «Нажать клавишу и удерживать» и «Отпустить клавишу», позволяющие указать клавишу, которая должна быть зажата. Робот может выполнять любые действия с зажатой клавишей и отпустить ее в нужный момент сценария.

Обращаем внимание! Рекомендуем использовать действиями парой и проверять, что после нажатия клавиши, робот ее отпускает. После выполнения действия «Нажать клавишу и удерживать» клавиша продолжает оставаться нажатой, пока не будет отпущена действием «Отпустить клавишу» или нажата соответствующая клавиша на клавиатуре.

#### *Группа действий «Машинное зрение»*

- Добавлено действие «Распознать капчу», позволяющее пользоваться сервисом RuCaptcha для распознавания капчи. Для использования сервиса необходимо подключение к интернету и API ключ сервиса, который можно получить в личном кабинете после регистрации. С условиями предоставления, стоимостью сервиса и его возможностями можно ознакомиться на официальном сайте: <https://rucaptcha.com/>.

#### *Группа действий «Базы данных»*

- Исправлена ошибка в действии «Стандартное подключение». Теперь дополнительные параметры подключения, указанные в параметре «Параметры» используются действием.

#### *Группа действий «Интеграции»*

- Исправлена ошибка в действии «Отправить запрос» подгруппы «REST». Теперь заголовки указанные в параметре «Заголовки запроса» используются действием.

#### *Группа действий «Базовые действия»*

- Исправлена ошибка в версии действия «Пока» с реализацией на java. Теперь при указании в параметре «Условие» переменной, действие проверяет значение переменной при каждой итерации

*Известные проблемы:*

- Для Firefox в действии «Открыть браузер» используется драйвер для 32 битной версии. Для корректной работы на системе x64 надо указать в конфиге свой драйвер по инструкции.
- Действие «Сохранить изображение» группы «Браузеры» может работать не со всеми изображениями, которые можно найти на страницах.
- Возможна проблема при публикации проектов роботов в версии Enterprise. Чтоб минимизировать вероятность проблем: 1) перед публикацией надо обязательно сохранить проект, 2) при изменении входных параметров робота, надо изменить контракт, сохранить, затем опубликовать проект робота.

## 13.5 Release Notes 2.5.2

### Что нового

#### Группа действий «Имитация устройств»

- Добавлены действия «Нажать клавишу и удерживать» и «Отпустить клавишу», позволяющие указать клавишу, которая должна быть зажата. Робот может выполнять любые действия с зажатой клавишей и отпустить ее нужный момент сценария.

Обращаем внимание! Рекомендуем использовать действиями парой и проверять, что после нажатия клавиши, робот ее отпускает. После выполнения действия «Нажать клавишу и удерживать» клавиша продолжает оставаться нажата, пока не будет отпущена действием «Отпустить клавишу» или нажата соответствующая клавиша на клавиатуре.

#### Группа действий «Машинное зрение»

- Добавлено действие «Распознать капчу», позволяющее пользоваться сервисом RuCaptcha для распознавания капчи. Для использования сервиса необходимо подключение к интернету и API ключ сервиса, который можно получить в личном кабинете после регистрации. С условиями предоставления, стоимостью сервиса и его возможностями можно ознакомиться на официальном сайте: <https://rucaptcha.com/>.

## 13.6 Release Notes 2.5.0

### Что нового

#### ROBIN Studio

- Реализована блокировка внесения любых изменений в схему робота и действия (изменение версии действий, добавление новых действий на схему, включать или выключать действия на схеме. . . .), пока выполняется отладка. Такие изменения не влияли на процесс отладки, но могли вызывать неверные ожидания после внесения изменений
- Исправлена ошибка с комментариями к действиям. Теперь у каждого действия свой комментарий и их содержание сохраняется
- Исправлена проблема с сохранением значений в параметрах Действий с типом Объект. В том числе проблема с действием «Отправить текст» группы «Браузеры»
- Исправлена ошибка при копировании действий из одной схемы в другую, приводившая к невозможности открыть схему после сохранения

#### ROBIN Player

- Добавлена история запуска роботов. На отдельной вкладке отображается таблица со списком запусков. Для каждого запуска представлены результат выполнения и логи запуска.

#### Группа действий «Браузеры»

- Исправлена проблема с возникновением ошибки ElementNotReachable. В частности, для действий Отправить текст и Отправить клик.

*Группа действий «Коллекции»*

- Исправлена ошибка, которая возникала при использовании действий «Получить значение по индексу», «Проверить на содержание значения»

*Группа действий «Файловая система»*

- Исправлен тип возвращаемого результата действия «Найти файл». Теперь элементы коллекции можно указывать в параметрах действий с типом «Путь к файлу»

*Группа действий «Excel»*

- Оптимизирована работа с большими таблицами. Ожидается ускорение работы, а также возможность работать с таблицами, которые ранее вызывали зависание или выполнялись неоправданно долго.

*Группа действий «Google sheets»*

- Добавлено действие «Взять данные из строки», позволяющее получить коллекцию значений всех ячеек в строке.
- Добавлено действие «Взять данные из столбца», позволяющее получить коллекцию значений всех ячеек в столбце.
- Добавлено действие «Установить значения строки», позволяющее заполнить ячейки в строке значениями из коллекции.
- Добавлено действие «Установить значения столбца», позволяющее заполнить ячейки в столбце значениями из коллекции.
- Добавлено действие «Очистить область», позволяющее очистить значения ячеек в указанной области.
- Добавлено действие «Удалить строку», позволяющее удалить строку таблицы со смещением вверх остальных.
- Добавлено действие «Удалить столбец», позволяющее удалить строку таблицы со смещением вверх остальных.
- Добавлено действие «Сортировать», позволяющее выполнить сортировку области по значениям заданного столбца.
- Добавлено действие «Установить цвет диапазона», позволяющее изменить цвет заливки ячеек выбранного диапазона.
- Изменено действие «Установить фильтры»:

Убраны параметры «Сортировать по возрастанию», «Сортировать по убыванию», «Индекс колонки для сортировки», «Начальный индекс колонки», «Начальный индекс строки», «Конечный индекс колонки», «Конечный индекс строки» Добавлены параметры «Ячейка От», «Ячейка До», «Столбец», «Критерий» Изменен функционал. Больше не выполняет функцию сортировки. Теперь позволяет указать критерий фильтрации диапазона.

- Изменено действие «Установить стиль текста» (бывшее Изменить стиль ячейки):

Убраны параметры «Начальный индекс колонки», «Начальный индекс строки», «Конечный индекс колонки», «Конечный индекс строки», «Красный», «Зеленый», «Синий» Добавлены параметры «Ячейка От», «Ячейка До», «Стиль шрифта», «Цвет», «Код цвета», «ID документа» Изменен функционал. Теперь дополнительно позволяет указать шрифт и стиль шрифта, а также указать предопределенный цвет текста

- Изменены названия параметров:

«Идентификатор таблицы» в «ID документа» в большинстве действий «Результат» в «ID документа» в действиях «Создать новый документ»

*Группа действий «Outlook»*

- Исправлена ошибка в действии «Сохранить вложения», в результате которой название папок могло быть неполным

## 13.7 Release Notes 2.4.0

### Что нового

#### *ROBIN Studio*

- Добавлена возможность выбора уровня логирования в окне Параметров запуска в студии - изменяет количество логов которые студия пишет при отладке
- Добавлена возможность перехода к действию при использовании поиска в схеме, выбрав его в результатах поиска
- Улучшена работа вин-шпиона и повышена стабильность работы
- Добавлено подтверждение при импорте робота в студию, когда робот с таким названием уже существует
- Убрано ошибочное сообщение об успешном экспорте робота, когда операция была отменена
- Исправлено поведение, при котором параметры действий заполненные только пробелами, оказывались не заполненными после открытия робота
- Стабилизирована работа Try-catch. Исправлено поведение при котором возникновение ошибки приводило к остановке исполнения робота, вместо обработки блоком try-catch
- Актуализированы описания, подсказки, документация по всем действиям

#### *ROBIN Player*

- При создании расписания в плеере добавлена возможность указать файл с параметрами запуска

#### *Группа действий «Базовые действия»*

- Добавлено новое действие «Изменить координаты точки», позволяющее изменить значение переменной типа «Точка»
- В действии «Try catch» исправлено нарушение нумерации блоков Catch при их удалении

#### *Группа действий «Базы данных»*

- Улучшено отображение возможных ошибок при работе действий, теперь сообщение в логе студии содержит подробное описание ошибок
- Исправлены возможные ошибки при работе совместно с действиями, написанных на других платформах: .NET и Python.

#### *Группа действий «Браузеры»*

- В действие «Отправить текст» добавлена возможность работать с типом переменной «Пароль»
- В действии «Открыть браузер» добавлен параметр «Сохранить в», позволяющий указать каталог для сохранения скачиваемых файлов по умолчанию.

#### *Группа действий «Диалоговые окна»*

- Доработаны действия «Сообщение» и «Вопрос» с реализацией на Java. Теперь иконка сообщения отображается в панели задач.

#### *Группа действий «Коллекции»*

- Доработано действие «Получить индекс значения» подгруппы «Массивы». Теперь если пытаться получить индекс элемента, которого нет в коллекции, возникает соответствующая ошибка

#### *Группа действий «Машинное зрение»*

- Улучшена стабильность работы и качество распознавания картинки на экране в действиях «Ждать появления» и «Существует на экране»

*Группа действий «Распознавание текста»*

- Стабилизирована работа с ABBYY Flexi Capture.
- Добавлена возможность указать папку для сохранения файла с распознанным текстом, полученный от сервиса

*Группа действий «Таблицы»*

- В действии «Сохранить таблицу в csv» расширение файла теперь добавляется автоматически

*Группа действий «Outlook»*

- В результате выполнения действия «Сохранить вложения» робот теперь возвращает список путей для папок с вложениями

## 13.8 Release Notes 2.3.0

### Что нового

*Группа действий «Таблицы»*

- Добавлено действие «Посчитать количество строк», позволяющее сделать подсчет строк в выделенной таблице

*Группа действий «Базовые действия»*

- Добавлена возможность конвертировать строку в переменную типа «Пароль» в действие «Конвертировать данные»
- Изменен тип свойства «Условие» на строковую в действие «Ветвление»
- Добавлены иконки ко всем действиям группы

*Группа действий «Браузеры»*

- Добавлена возможность обработки ошибка типа «Любая» для всех действий группы

*Группа действий «Outlook»*

- Исправлено поведение фильтров в действии «Выбрать сообщения». Теперь фильтры, устанавливаемые в действии, работают правильно.

*Группа действий «Имитация устройств»*

- Исправлено поведение действия «Ввод текста» при котором, в некоторых случаях, при вводе текста неправильно переключался регистр.

*Общие компоненты студии:* \* В окне «Параметры запуска роботов» теперь можно указать только следующие типы: строка, число, пароль, дата/время, логический \* Для enterprise версии студии: Отключена возможность экспортировать и загружать схемы роботов с помощью файлового менеджера, публикация готовых роботов невозможна без публикации проекта робота, после публикации робот виден в RMS и Player с соответствующими правами.

### Улучшения:

- Стабилизирована работа действий на системе исполнения Python

## 13.9 Release Notes 2.2.0

### Что нового

*Группа действий «Word»*

- Добавлен результат в действие «Сохранить», который содержит путь до сохранённого файла;

*Группа действий «Excel»*



- Добавлен результат в действие «Сохранить», который содержит путь до сохранённого файла;
- Добавлен результат в действие «Создать файл», который содержит путь до созданного файла;

*Группа действий «FTP»*

- Стабилизирована работа группы

*Группа действий «Windows»*

- Добавлено свойство «Фокус» в действие «Фокус на элементе», позволяющее работать с элементами, которые при фокусе на них ведут себя так, будто по ним был произведен ЛКМ;

*RMC:*

- Добавлена возможность скачивания отчета о запуске роботов;
- Для облачной версии: отображаются все роботы, доступные компании;

*Общие компоненты студии:*

- Добавлена обработка переменной типа Пароль;
- В окно web-рекордера добавлены кнопки «Пауза», позволяющая приостановить запись, «Стоп», позволяющая остановить запись, и «Продолжить», позволяющая продолжить запись после нажатия паузы;
- Убрано стартовое окно при первом входе в студию, теперь всегда отображается список роботов;
- Для облачной студии: нет возможности экспортировать и загружать исходники роботов с помощью файлового менеджера, публикация готовых роботов невозможна без публикации проекта робота, после публикации робот виден на портале, gms и в плеере;
- Для standalone версии студии: убрана возможность публикации исходников и готовых роботов;
- Оптимизирован дистрибутив и уменьшен размер устанавливаемых компонентов;

*Известные проблемы:*

- Действия на системе исполнения python после установки студии отрабатывают только со второго раза;

## 13.10 Release Notes 2.1.0

### Что нового

*Группа действий «Коллекции» подгруппа «Массивы»:*

- Стабилизирована работа действий группы;
- Добавлено свойство «Порядок сортировки» в действие «Сортировать», позволяющее указать порядок сортировки;

*Группа действий «Файловая система» подгруппа «Файлы»:*

- Добавлено свойство «Результат» у действий «Копировать файл», «Переместить файл», «Переименовать файл», позволяющее записать в результат действий путь до файла;

*Группа действий «Файловая система» подгруппа «Папки»:* \* Добавлено свойство «Результат» у действий «Копировать папку», «Переместить папку», «Переименовать папку», позволяющее записать в результат действий путь до папки;

*Группа действий «Браузеры 1.3»:*

- Добавлено свойство «Количество повторов нажатия» в действие «Отправить горячие клавиши», позволяющее указать количество нажатий

*Общие компоненты студии:*

- Добавлена возможность запуска студии под режимом разработчика действий;
- В режиме разработчика действий добавлена кнопка «Обновить кэш», позволяющая моментально обновить кэш в студии;

**Улучшения:**

- Стабилизировано открытие проектов старых версий в студии;
- Решена проблема сохранения результатов сценария при переключении сценариев на системе исполнений java;

## 13.11 Release Notes 2.0.12

**Что нового**

*Группа действий «Чатботы»*

- Стабилизирована работа действий группы;
- Добавлено свойство «Логин подключения» в действие «Настройки подключения», позволяющее указать логин для подключения к системе управления очередью сообщений;
- Добавлено свойство «Пароль подключения» в действие «Настройки подключения», позволяющее указать пароль для подключения к системе управления очередью сообщений;
- Добавлено свойство «Учетные данные» в действие «Настройки подключения», позволяющее указать учетные данные для подключения к системе управления очередью сообщений;

*Группа действий «Интеграции», подгруппа «MQ»*

- Стабилизирована работа действий группы;
- Добавлено действие «Создать подключение к MQ», позволяющее устанавливать соединение к брокеру MQ и создавать контекст с подпиской на выбранный топик;
- Добавлено действие «Разорвать подключение к MQ», позволяющее завершить сессию обмена сообщениями с сервером (брокером MQ);
- Удалено свойство «Топик» из действия «Опубликовать в топик MQ»;
- Удалено свойство «Топик» из действия «Получить сообщение»;

**Улучшения:**

- Ускорен запуск роботов;
- Ускорено выполнение действий роботами;
- Стабилизирована работа студии в режиме «Для всех», запуск роботов, работа под несколькими пользователями одновременно;
- Стабилизирована работа группы действий «Коллекции»;
- Стабилизирована работа группы действий «Браузеры», исправлены проблемы при работе с браузером Firefox;

## 13.12 Release Notes 2.0.11

**Что нового**

*Группа действий «Excel»*

- Переименовано свойство «Критерий» на «Критерий 1» в действии «Выбрать по автофильтру»

- Добавлено свойство «И / или» в действие «Выбрать по авто-фильтру», позволяющее контролировать логику применения критериев
- Добавлено свойство «Критерий 2» в действие «Выбрать по авто-фильтру», позволяющее указать критерий фильтрации по второму условию

*Общие компоненты студии:*

- В панель логирования добавлен поиск по содержимому лога
- В панель логирования добавлена кнопка «Копировать лог», позволяющая копировать лог в буфер обмена
- При установке студии встроено дополнительное окно, где запрашиваются конфигурации серверов. Следует заполнять только в случае использования совместно с ROBIN Orchestrator 2.0.
- Встроен механизм, позволяющий добавлять кастомные действия в студию
- Разработана дополнительная утилита ActionEditor, позволяющая создавать действия
- Разработана дополнительная утилита ActionLoader, позволяющая добавлять и удалять действия из студии, а также отдельно удалять реализации действий

**Улучшения:**

- Стабилизирована работы группы действий «Excel»: решена проблема фильтрации в больших документах
- Стабилизирована работа студии: решена проблема передачи схем
- Исправлена проблема с диалоговым окном при закрытии схемы, где были изменения

## 13.13 Release Notes 2.0.10

**Что нового**

*Группа действий «FTP»*

- Добавлено свойство «Хост» в действие «Подключить», позволяющее ввести адрес хоста
- Удалено свойство «Порт» в действии «Подключить»
- Переименовано свойство «Имя пользователя» в действии «Подключить» на «Логин»
- Удалено свойство «Учетные данные» в действии «Подключить»
- Переименовано действие «Получить детали содержимого папки» на «Получить содержимое папки»
- Действие «Получить содержимое папки» в старой версии удалено

*Группа действий «Интеграции», подгруппа «REST»*

- Удалено действие «GET»
- Удалено действие «POST»
- Добавлено действие «Отправить запрос», позволяющее отправлять HTTP-запрос для обращения к RESTful веб-сервису

*Группа действий «Базы данных», подгруппа «Общее»*

- Добавлено действие «Пользовательское подключение», позволяющее создать контекст подключения к базе данных, тип которой не входит в перечень стандартных

*Группа действий «Браузеры»*

- Добавлена возможность в свойство «Доступность» в действия «Ждать состояние элемента» и «Проверка наличия элемента», позволяющая проверить не только параметр disabled, но и факт что элемент не перекрыт другими элементами и с ним можно взаимодействовать.

*Общие компоненты студии:*

- При захвате элемента, вин-шпион не строит дерево элементов автоматически, а загружает только захваченный элемент и отображается кнопка «Построить дерево»
- В вин-шпион добавлена кнопка «Построить дерево», при клике для приложения строится дерево элементов, после чего элементы можно выбирать из дерева, как раньше (после построения дерева кнопка скрывается)
- При работе с вин-шпионом и веб-шпионом из панели параметров - сразу после захвата элемента создается ресурс
- Исправлено сохранение файла логов роботов в папку студии
- Исправлена передача переменной типа «Таблица» между действиями написанными на разных языках.

**Улучшения:**

- Убрано зеленое уведомление при открытии студии
- Убрано голубое уведомление при открытии схемы
- В выпадающем списке в панели логов пустая строка заменена на «Все»
- В панели логов убраны вкладки (перед установкой нужно удалить файл ApplicationConfiguration.xml в папке студии)
- Общие косметические улучшения интерфейса студии
- Решена проблема «Element not reachable» при работе с группой действий «Браузеры»
- Решена проблема с паролями при работе с группой действий «Браузеры 1.3»
- Решена проблема, когда студия не могла обнаружить агента при попытке запуска робота
- Решены проблемы платформы после установки .NET 5
- Решена проблема с вылетом студии после захвата вин-элемента
- Стабилизирована работа студии в режиме «Для всех»
- Стабилизирована работа студии с драйверами для браузеров
- Стабилизирована работа совместного использования действий из групп Word, Excel и Outlook в одном роботе

## 13.14 Release Notes 2.0.9

**Что нового**

*Группа действий «Интеграции», подгруппа «SOAP»*

- В действии «SOAP запрос» изменено свойство «Заголовки», теперь оно необязательно для заполнения
- Добавлено свойство «XML», позволяющее отправлять запрос в виде XML- файла
- Скрыта предыдущая версия действия «SOAP запрос»

*Группа действий «Словари»*

- Добавлено действие «Получить коллекцию значений», позволяющее получить коллекцию всех значений словаря

- Добавлено действие «Получить коллекцию ключей», позволяющее получить коллекцию ключей словаря

#### *Общие компоненты студии*

- В верхнее меню студии добавлен функционал «Поиск», позволяющий находить действия на схеме по наименованию
- В верхнее меню студии добавлен функционал «Массовые действия», позволяющий включать все выключенные действия, снимать все точки останова, сворачивать/разворачивать все блоки try-catch, группы действий, циклы, условия, ветвления и выражения (функционал для трех последних кнопок находится в разработке)
- Удалено предложение о запуске студии в конце установки

#### **Улучшения:**

- Стабилизирована работа студии в режиме «Для всех»
- Агент сворачивается, но не отображается в трее. Для того, чтоб выключить его нужно запустить утилиту Kill.Robin.Agent (находится в папке агента)
- Стабилизирована работа веб-рекордера, решена проблема с проставлением версий действий при записи
- Ускорен запуск и работа роботов

#### *Известные проблемы:*

После перезагрузки компьютера агент не стартует самостоятельно, необходимо включить его вручную

## 13.15 Release Notes 2.0.8

### **Что нового**

#### *Группа действий «Браузеры»*

- Добавлено действие «Получить таблицу» в подгруппу действий «Работа с элементами», позволяющее получить таблицу в переменную
- В действии «Открыть браузер» для свойства «Значение для Тайм-аут» изменена единица измерения времени на миллисекунды
- В действиях «Открыть браузер» и «Перейти по URL» для свойства «URL» добавлено автоматическое определение протокола

#### *Добавлена группа действий «Базы данных»*

- Действие «Стандартное подключение» позволяет создать контекст подключения к стандартной базе данных
- Действие «Закрывает подключение» позволяет закрыть подключение к базе данных
- Действие «Добавляет запись» позволяет добавить запись (строку) в указанную базу данных
- Действие «Обновляет запись» позволяет обновить существующие записи в базе данных
- Действие «Удаляет записи» позволяет удалить записи в базе данных
- Действие «Извлекает записи» позволяет извлечь запись из указанной таблицы в базе данных
- Действие «Выполняет запрос» позволяет выполнить запрос на получение таблиц из базы данных
- Действие «Откатить транзакцию» позволяет откатить текущую транзакцию
- Действие «Закрыть транзакцию» позволяет закрыть текущую транзакцию

*Группа действий «Таблицы»*

- Действие «Сохранить таблицу в CSV» реализовано в системе исполнения java

*Общие компоненты студии:*

- Добавлена возможность делать скриншоты из панели ресурсов и сохранять их
- Добавлена отправка писем о завершении и падении роботов из плеера и оркестратора
- Добавлены недостающие конвертеры в системе исполнения Python
- Добавлен вертикальный скролл для панели параметров веб-элемента
- Добавлена подсветка элементов при работе WebSpy

**Улучшения:**

- Стабилизирована работа группы действий «Файловая система» подгруппа «Архивы»
- Стабилизирована работа действия «Фокус на элементе» группы действий «Windows»
- Улучшена работа ROBIN Agent и Player
- Улучшена поддержка работы роботов, которые были созданы на предыдущих версиях студии
- Стабилизирована работа Try-Catch, добавлена передача исключений между системами исполнения
- Исправлены ошибки в действиях по работе с Браузером
- Стабилизирована работа действий, написанных на python
- Исправлены ошибки, приводящие к нештатным завершениям работы ROBIN Studio
- Добавлена подсветка веб-элементов при записи сценариев.

*Известные проблемы:*

- Нестабильная работа студии в режиме установки «для всех»

## 13.16 Release Notes 2.0.7

**Что нового***Группа действий «Машинное зрение»*

- Добавлено исключение «Файл не найден» во все действия группы
- Добавлено исключение «Параметр не найден» во все действия группы

*Группа действий «Windows»*

- Добавлено исключение «Окно не найдено» во все действия, которые работают с окнами
- Добавлено исключение «Win-элемент не найден» во все действия, которые работают с элементами
- Добавлено исключение «Время истекло» во все действия, которые работают с тайм-аут

*Группа действий «Безопасность»*

- Добавлена иконка для группы и действия

*Группа действий «Имитация устройств»*

- Добавлено свойство «Задержка» в действие «Ввод текста», позволяющее установить время между вводом символов из свойства «Текст»

*Группа действий «Файловая система»*

- Добавлено исключение «Файл не найден» во все действия подгруппы «Файлы»

- Добавлено исключение «Файл недоступен» во все действия подгруппы «Файлы»
- Добавлено исключение «Папка не найдена» во все действия подгруппы «Папки»
- Добавлено исключение «Папка недоступна» во все действия подгруппы «Папки»

*Общие компоненты студии:*

- Добавлена возможность использования параметров робота при запуске из студии

#### **Улучшения:**

- Стабилизирована работа группы действий «Файловая система» подгруппа «Папки»
- Стабилизирована работа группы действий «XML»
- Исправлены ошибки работы действия «Ввод текста» группы действий «Имитация устройств»
- Исправлена работа действия «Извлечь по шаблону» группы действий «Текст»
- Исправлена работа действия «Получить свойство элемента» группы действий » Windows»
- Исправлена работа действия «Сохранить» группы действий «Excel» связанная с указанием пути сохранения файл
- Исправлена работа исключений группы действий «Браузеры»

#### **Общие компоненты студии:**

- Добавлено предупреждение об устаревших действиях
- Добавлена поддержка устаревших действий в роботах
- Стабилизирована работа конвертеров типов данных
- Добавлено окно ввода входных параметров робота в студии
- Стабилизирована работа студии на win server 2008
- Исправлена работа действий с веб-элементами на java
- Изменена иконка View Finder, улучшен его интерфейс и функционал

## **13.17 Release Notes 2.0.6**

### **Что нового**

#### *Группа действий «Браузеры»*

- Добавлено исключение «Вкладка не найдена» во все действия, которые работают с вкладками
- Добавлено исключение «Веб-элемент не найден» во все действия, которые работают с элементами (в т.ч. фреймами)
- Добавлено исключение «Веб-элемент вне доступа» в действия «Оправить клик» и «Отправить текст»
- Добавлено исключение «Параметр не найден» во все действия группы
- Добавлено исключение Время истекло во все действия, которые работают с ожиданиями
- Добавлено свойство «Индекс» в действие «Переключиться на вкладку», позволяющее идентифицировать по индексу
- Добавлено свойство «Индекс» в действие «Закрыть вкладку», позволяющее идентифицировать по индексу

#### *Группа действий «Базовые действия»*

- Добавлено действие «Конвертировать данные», позволяющее конвертировать данные одного типа в другой

- Добавлено исключение «Неконвертируемый формат» в действие «Конвертировать данные»
- Параметр Sequence ID в действии «Группа действий» изменен на необязательный

#### *Группа действий «XML»*

- Добавлено исключение «Файл не найден» в действие «Открытие файла»
- Добавлено исключение «Параметр не найден» во все действия
- Добавлено исключение «Папка не найдена» в действие «Сохранение»

#### *Группа действий «Текст»*

- Добавлена подгруппа действий «Классификация текста»

Группа действий Preferentum переименована в «Классификация текста» и перенесена в подгруппа «Текст»

#### *Группа действий «Файловая система»*

- Изменен тип данных в результате действия «Получить содержимое папки» на коллекцию, содержащую пути до всех вложенных элементов

Добавлена группа действий «Безопасность»: \* Добавлено действие «Получить пароль из хранилища», позволяющее сохранять логин и пароль из указанной записи в диспетчере учетных данных Windows

#### **Улучшения:**

Группа «Текст» в действие «Разделить» добавлена возможность делить по переносу строк

#### *Общие компоненты студии:*

- Улучшен интерфейс компонента WinSpy
- Исправлено отображение документации по кнопке «Подробнее»
- Улучшен интерфейс окна «Спецификация робота»
- Доработано версионирование действий робота
- Добавлено подробное логирование при публикации роботов
- При установлении признака «Входной параметр робота» переменная отображается в спецификации робота во вкладке «Параметры»
- При установлении признака «Результат работы робота» переменная отображается в спецификации робота во вкладке «Результаты»

## 13.18 Release Notes 2.0.5

### **Что нового**

#### *Группа действий «Таблицы»:*

- Добавлено исключение «Ячейка не найдена» в действие «Получить данные из ячейки», позволяющее поймать эту ошибку в действии TryCatch.
- Добавлено исключение «Ячейка не найдена» в действие «Установить значение значение ячейки», позволяющее поймать эту ошибку в действии TryCatch.

#### *Группа действий «Word»:*

- Добавлено исключение «Файл не найден» в действие «Открыть», позволяющее поймать эту ошибку в действии TryCatch.
- Добавлено исключение «Файл не доступен» в действие «Открыть», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.



- Добавлено исключение «Папка не найдена» в действие «Сохранить как», позволяющее поймать эту ошибку в действии TryCatch.
- Добавлено исключение «Файл не доступен» в действие «Сохранить как», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.
- Во все действия группы добавлено исключение «Параметр не найден», позволяющее поймать действием TryCatch ошибку не созданного контекста для документа.

*Группа действий «Excel»:*

- Добавлено исключение «Файл не найден» в действие «Открыть», позволяющее поймать эту ошибку в действии TryCatch.
- Добавлено исключение «Файл не доступен» в действие «Открыть», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.
- Добавлено исключение «Папка не найдена» в действие «Сохранить», позволяющее поймать эту ошибку в действии TryCatch.
- Добавлено исключение «Файл не доступен» в действие «Сохранить», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.
- Добавлено исключение «Параметр не найден» в действие «Сохранить», позволяющее поймать действием TryCatch ошибку не созданного контекста для книги.
- Добавлено исключение «Параметр не найден» в действие «Закрыть», позволяющее поймать действием TryCatch ошибку не созданного контекста для книги.

*Группа действий «PDF»:*

- Добавлено исключение «Файл не найден» в действие «Открыть», позволяющее поймать эту ошибку в действии TryCatch.
- Добавлено исключение «Файл не доступен» в действие «Открыть», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.

*Группа действий «Файловая система»: Подгруппа «Файлы»:*

- Во все действия подгруппы добавлено исключение «Файл не найден», позволяющее поймать эту ошибку действием TryCatch.
- Во все действия подгруппы добавлено исключение «Файл недоступен», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.

*Подгруппа «Папки»:*

- Во все действия подгруппы добавлено исключение «Папка не найдена» в все действия подгруппы, позволяющее поймать эту ошибку действием TryCatch.
- Во все действия подгруппы добавлено исключение «Папка недоступна», позволяющее поймать действием TryCatch ошибку обращения к папке, файлы которой заняты другим процессом.

*Подгруппа «Архивы»:*

- Во все действия подгруппы добавлено исключение «Файл не найден», позволяющее поймать эту ошибку действием TryCatch.
- Во все действия подгруппы добавлено исключение «Файл недоступен», позволяющее поймать действием TryCatch ошибку обращения к файлу, занятому другим процессом.
- Во все действия подгруппы добавлено исключение «Папка не найдена» в все действия подгруппы, позволяющее поймать эту ошибку действием TryCatch.
- Во все действия подгруппы добавлено исключение «Папка недоступна», позволяющее поймать действием TryCatch ошибку обращения к папке, файлы которой заняты другим процессом.

**Улучшения**

- Добавлена возможность ловить ошибки любого и неизвестного типов в действии TryCatch группы Базовые.

- Добавлена возможность делить текст по переносу строк в действии «Разделить» группы «Текст» с помощью ввода символов «`gn`» в параметр «Разделитель».
- Тип данных «Папка» переименован в «Путь к папке».
- Тип данных «Файл» переименован в «Путь к файлу».
- Тип данных «Прямоугольник» переименован в «Область».
- Заменена иконка группы Коллекции.
- Заменена иконка группы Операторы.
- Заменены шаблонные однотипные иконки на уникальные для тех действий, которые таковых не имели (кроме группы Браузеры, Файловая система и Таблицы).
- Заменены шаблонные однотипные иконки на уникальные для тех типов данных, которые таковых не имели.
- Исправлена опечатка в значении «координаты» из выпадающего списка для параметра «Тип» действия «Сделать скриншот» группы «Базовые».
- Исправлена опечатка в названии параметра «С новой строки» действия «Ввести текст» группы «Word».
- Исправлена опечатка в названии параметра «Словарь» действия «Найти значение» группы «Коллекции».
- Исправлена опечатка в описании подсказки к действию «Удалить столбец» группы «Таблицы».

#### *Известные ошибки:*

- Не работает разделение по переносу строк в действии «Разделить» с помощью ввода символов «`gn`», если заполнять параметр «Текст» вручную (не из файла).
- При маленьком разрешении в инструменте TryCatch не отображаются кнопки «Сохранить» и «Отмена».
- Проставляется не последняя версия действий при добавлении в рабочую область. Нужно выбирать ручную.

## 13.19 Release Notes 2.0.4

### **Что нового**

#### *RMC:*

- Добавлено логирование в режиме реального времени
- Добавлена история запусков

#### *Группа действий «Машинное зрение»:*

- Добавлено действие «Считать штрихкод с изображения», позволяющее найти и считать штрихкоды (линейных и двумерных - QR) с изображений
- Добавлено действие «Считать штрихкод из ПДФ», позволяющее найти и считать штрихкоды (линейных и двумерных - QR) с указанной страницы PDF документа

#### *Группа действий «Браузеры», подгруппа «Работа с окнами, вкладками, фреймами браузера»*

- Добавлено свойство «Расширенные настройки (Свойства)» в действие «Открыть браузер», позволяющее установить список свойств при открытии браузера
- Добавлено свойство «Значение для Тайм-аут (Время явного ожидания)» в действие «Открыть браузер», позволяющее задать время ожидания

- Добавлено свойство «URL» в действие «Открыть браузер», позволяющее задать адрес страницы, на которую при открытии браузера должен быть выполнен переход
- Удалены свойства «Тип страницы», «Профиль», «Подключить расширения» в действии «Открыть браузер»
- Добавлено свойство «В новой вкладке» в действие «Перейти по URL», позволяющее открыть нужный url в новой вкладке

Перенесено действие Сгенерировать случайное число из группы действий «Текст» в группу действий «Базовые действия»

### Улучшения

Обеспечен автоматический поиск и использование веб-драйвера подходящего для версии браузера доступного пользователю. Исправлена ошибка активирования нового листа в действиях «Взять значение ячейки», «Получить данные из столбца», «Получить данные из строки» группы действий Excel Исправлена ошибка следования действия «Активировать лист» перед действием «Установить значение строки» группы действий Excel Исправлена ошибка отправки писем без вложений группы действий «Почта» Исправлена работа действия «Завершить процесс» Исправлены ошибки группы действий «Конвертация дат»

*Общие компоненты студии:*

- Стабилизирована работа системы исполнения python и java
- Улучшена работа инструмента выражений
- Появилась возможность подавать переменную типа «Объект» в качестве входного параметра во все действия
- Отключена работа Win-шпиона и Win-рекордера с панелью задач, проводником и рабочим столом
- Реализована работа горячих клавиш в студии
- Стабилизирована работа публикации роботов

*Известные ошибки:*

- Действие «Переключиться на вкладку» не работает
- Win-шпион не реагирует на кнопки свернуть/развернуть/закрыть, в Notepad++ не всегда реагирует на иконки меню (плавающая проблема на разных машинах)

## 13.20 Release Notes 2.0.3

### Что нового

*Группа действий «Конвертация дат»:*

- Добавлено действие «Получить элемент даты», позволяющее получить из выбранной даты необходимый элемент
- Добавлено свойство «Тип результата» в действие «Вычесть дату», позволяющее выбрать тип результата
- Удалено свойство «Формат» в действии «Преобразовать строку в дату»

Переделаны подгруппы «Свойства элемента» и «Простые действия с элементами» в подгруппу «Элементы и свойства» группы «Браузеры»

*Группа действий «Браузеры», подгруппа «Элементы и Свойства»*

- Добавлено свойство «Убрать лишние пробелы» в действие «Получить текст», позволяющее удалять лишние пробелы и пробельные символы

- Добавлено свойство «Все текстовое содержимое» в действие «Получить текст», позволяющее получить все текстовое содержимое
- Добавлено действие «Получить элементы», позволяющее получить коллекции всех элементов, удовлетворяющих заданным условиям, в переменную
- Добавлено свойство «Выбор атрибута» в действие «Получить атрибут элемента», позволяющее выбрать тип атрибута, значение которого нужно узнать
- Добавлено свойство «Ввод атрибута» в действие «Получить атрибут элемента», позволяющее указать ручную атрибут
- Добавлено свойство «Выбор атрибута» в действие «Установить атрибут элемента», позволяющее выбрать тип атрибута, значение которого нужно узнать
- Добавлено свойство «Ввод атрибута» в действие «Установить атрибут элемента», позволяющее указать ручную атрибут
- Добавлено свойство «Имя свойства» в действие «Получить стиль элемента», позволяющее указать имя свойства стиля, значение которого нужно узнать
- Добавлено свойство «Имя свойства» в действие «Установить стиль элемента», позволяющее указать имя свойства стиля, значение которого нужно установить
- Добавлено свойство «Значение свойства» в действие «Установить стиль элемента», позволяющее указать значение свойства стиля
- Добавлено свойство «Внутренние значения» в действие «Получить значение списка», позволяющее выбрать в результирующую коллекцию внутренние значения списка
- Добавлено свойство «Убрать лишние пробелы» в действие «Получить значение списка», позволяющее установить необходимость удаления лишних пробелов
- Добавлено действие «Установить значение списка», позволяющее установить для прокручиваемого или выпадающего списка (элемент Select) выбранные в списке опции
- Переименовано действие «Получить координаты элемента» в «Получить размер и координаты элемента»
- Добавлено свойство «Видимость» в действие «Проверка наличия элемента», позволяющее установить признак необходимости проверки видимости
- Добавлено свойство «Доступность» в действие «Проверка наличия элемента», позволяющее установить признак необходимости проверки доступности
- Добавлено свойство «Результат» в действие «Проверка наличия элемента», позволяющее сохранить результат действия
- Добавлено действие «Получить таблицу» позволяющее получить таблицу в переменную
- Удалены действия: «Получить тип элемента», «Проверить выбран ли элемент», «Пометить элемент выбранным», «Получить значение чек-бокса», «Получить значение радио-кнопки», «Очистить элемент», «Получить значение», «Установить значение»

*Группа действий «Браузеры», подгруппа «Мышь и клавиатура»*

- Переименовано действие «Расширенный клик» на «Отправить клик»
- Добавлено свойство «Горячие клавиши» в действие «Отправить клик», позволяющее установить список горячих клавиш
- Добавлено свойство «Смещение по оси X» в действие «Отправить клик», позволяющее установить смещение по оси X относительно левого верхнего угла элемента
- Добавлено свойство «Смещение по оси Y» в действие «Отправить клик», позволяющее установить смещение по оси Y относительно левого верхнего угла элемента
- Добавлено свойство «Быстрый ввод» в действие «Отправить текст», позволяющее отметить необходимость возможность быстрого ввода для больших объёмов текста

- Удалено свойство «Горячие клавиши» в действие «Отправить текст»
- Добавлено действие «Отправить горячие клавиши», позволяющее эмулировать нажатие горячих клавиш на странице на странице браузера
- Переименовано действие «Переместиться до элемента» на «Прокрутить до элемента»
- Добавлено свойство «Смещение по горизонтали» в действие «Прокрутить до элемента», позволяющее указать количество пикселей, на которое нужно прокрутить вправо в горизонтальном направлении
- Добавлено действие «Навести указатель мыши», позволяющее эмулировать наведение указателя мыши на выбранный элемент
- Удалены свойства: «Цепочка действий», «Цепочка действий» в результате в действие «Прокрутить до элемента»
- Удалены действия: «Клик», «Ввести текст», «Отправить нажатие клавиш», «Перетащить»

Добавлено действие «Получить элементы» (GetElementList) в группу Браузеры

*Группа действий «Браузеры», подгруппа «Ожидания»*

- Добавлено действие «Ждать значения атрибута», позволяющее проверить состояние выбора элемента по завершению тайм-аута в зависимости от переданного параметра
- Добавлено действие «Ждать значения стиля», позволяющее проверить состояние выбора элемента по завершению тайм-аута в зависимости от переданного параметра
- Добавлено свойство «Использовать все текстовое содержимое» в действие «Ждать текст элемента», позволяющее выбрать весь текст
- Добавлено свойство «Тайм-ауте» в действие «Ждать текст элемента», позволяющее установить время ожидания
- Добавлено действие «Ждать заголовок страницы», позволяющее проверить существование текста в элементе по завершению тайм-аута
- Добавлено действие «Ждать состояния элемента», позволяющее проверить существования элемента на странице по завершению тайм-аута
- Добавлено действие «Ждать исчезновения элемента», позволяющее проверить факта того, что элемент отсутствует в DOM по завершению тайм-аута
- Удалены действия: «Ожидание видимости элемента», «Ожидание кликабельности возможности нажать на элемента», «Ожидание невидимости элемента», «Ожидание невидимости элемента с текстом», «Ожидание видимости группы элементов», «Ожидание существования группы элементов»

*Группа «Текст»*

- Добавлено свойство «Строка составного формата» в действие «Составное форматирование», позволяющее указать строку, в которой места для подстановки текста обозначены как {x}, где x - порядковый номер объекта в коллекции
- Добавлено свойство «Объекты для подстановки» в действие «Составное форматирование», позволяющее указать коллекцию объектов, текстовое представление которых заменит в указанной строке обозначения вида {x} в соответствии с их порядковыми номерами
- Добавлено свойство «Пробелы считать пустотой» в действие «Проверить на null и пустоту», позволяющее указать признак: если текст содержит только пробельные символы, считать его пустым

## Улучшения

Перенесено действие «Сгенерировать случайное число» из группы «Текст» в группу «Базовые действия» Стабилизирована работа подгруппы «Диалоговые окна» группы «Браузеры»

*Общие компоненты студии:*

- Реализована возможность передачи типа «Объект» на вход во все действия
- Стабилизирована работа Web-recorder
- Стабилизирован запуск роботов в плеере из ROBIN Orchestrator

## 13.21 Release Notes 2.0.2

### Что нового

#### *Группа действий «Файловая система», подгруппа «Файлы»*

- Добавлено свойство «Расширение файла» в действие «Создать файл», позволяющее указать расширение создаваемого файла
- Добавлено свойство «Результат» в действие «Записать в файл», позволяющее сохранять в нем путь до созданного файла
- Добавлено свойство «Новое расширение» в действие «Переименовать файл», позволяющее указать расширение файла
- Переименовано действие «Прочитать файл» в «Прочитать текстовый файл»
- Переименовано действие «Записать в файл» в «Записать в текстовый файл»

#### *Группа действий «Файловая система», подгруппа «Архивы»*

- Добавлено свойство «Путь до файла» в действие «Архивировать», позволяющее указывать путь
- Добавлено свойство «Формат архива» в действие «Архивировать», позволяющее из выпадающего списка выбрать нужный формат
- Добавлено свойство «При дублировании» в действие «Архивировать», позволяющее выбрать действие в случае, если в папке уже есть архив с выбранным именем
- Добавлено свойство «Результат» в действие «Архивировать», позволяющее сохранять в нем путь до созданного архива
- Удалено свойство «Формат» в действие «Архивировать»
- Удалено свойство «Путь до gag.exe» в действие «Архивировать»
- Добавлено свойство «При дублировании» в действие «Разархивировать», позволяющее выбрать действие в случае, если в папке уже есть файл с выбранным именем
- Удалено свойство «Формат» в действие «Разархивировать»
- Удалено свойство «Путь до 7z.exe» в действие «Разархивировать»
- Удалено свойство «Путь до unpag.exe» в действие «Разархивировать»
- Удалено действие «Редактировать»

#### *Группа действий «Файловая система», подгруппа «Папки»*

- Добавлено свойство «Родительская папка» в действие «Создать папку», позволяющее указать путь до директории, где нужно создать папку
- Добавлено свойство «Результат» в действие «Создать папку», позволяющее сохранять путь до созданной папки
- Добавлено свойство «Папка» в действие «Удалить папку», позволяющее указать путь до папки, которую нужно удалить
- Добавлено действие «Очистить папку», позволяющее переместить в корзину содержимое указанной папки
- Переименовано действие «Проверить существование папки» на «Проверить существование»

- Добавлено свойство «Папка» в действие «Проверить существование», позволяющее указать путь до папки
- Добавлено свойство «Шаблон поиска» в действие «Найти папку», позволяющее указать маску поискового запроса для имени папки
- Добавлено свойство «Папка» в действие «Получить содержимое папки», позволяющее указать путь до папки
- Добавлено свойство «Папка» в действие «Копировать папку», позволяющее указать путь до папки
- Добавлено свойство «Папка» в действие «Переместить папку», позволяющее указать путь до папки
- Добавлено свойство «Папка» в действие «Переименовать папку», позволяющее указать путь до папки
- Переименовано действие «Получить папку» на «Получить родительскую папку»
- Добавлено свойство «Папка» в действие «Получить родительскую папку», позволяющее указать путь до папки

#### *Группа действий «Таблицы»*

- Добавлено действие «Добавить столбец», позволяющее добавить столбец в конец таблицы
- Добавлено действие «Добавить строку», позволяющее добавить строку в конец таблицы

#### *Группа «Конвертация дат»*

- Добавлено действие «Получить текущую дату», позволяющее получить из выбранной даты необходимый элемент

#### *Группа действий «Операторы», подгруппа «Логические»*

- Добавлено действие Not equal

#### *Группа действий «Коллекции»*

- Удалено действие «Очистить коллекцию»

#### *Группа действий «Браузеры»*

- В действии «Открыть браузер» удален параметр «Путь к драйверу». Теперь по умолчанию используется

Удалена группа действий «Функции»

### **Улучшения**

Добавлены описания по кнопке «Подробнее» для групп действий «Базовые действия», «Диалоговые окна», «Почта», «Excel», «Outlook», «Word», «PDF», «Файловая система», «Имитация устройств», «Таблицы», «Текст»

Стабилизирована работа групп действий PDF и Word

В действие «Сделать скриншот» в группе действий «Базовые действия» добавлена возможность выделения области экрана

#### *Общие компоненты студии*

- Реализовано получение информации о текущем пользователе (публикация роботов/источников)
- Реализован вывод типа параметра при наведении на подсказку
- Изменено наименование действий: добавлен порядковый номер, а также изменено наименование действия в параметрах на русский язык
- Изменено отображение значений логического типа на чекбокс

## 13.22 Release Notes 2.0.1

### Что нового

#### Группа действий «Почта»

- Добавлено новое свойство «URL сервера» в действие «Настройки подключения».
- Добавлено новые свойства «Название вложения», «Таблица результатов» в действие «Получить сообщение».

#### Группа действий «Текст»

- Перенесены из группы «Функция» в группу «Текст» следующие действия: «Проверить на null и пустоту», «Проверить окончание», «Проверить начало», «Удалить с начала и конца», «Заменить по шаблону», «Перевернуть».
- Добавлено новое свойство «Текст» в действие «Проверить на null и пустоту», используется для исходного текста для обработки.
- Добавлено новое свойство «Текст» в действие «Проверить окончание», используется для исходного текста для обработки.
- Добавлено новое свойство «Предполагаемое окончание» в действие «Проверить окончание», позволяющее вставлять текст, с которым будет сравниваться окончание исходного текста.
- Добавлено новое свойство «Текст» в действие «Проверить начало», используется для исходного текста для обработки.
- Добавлено новое свойство «Предполагаемое начало» в действие «Проверить начало», позволяющее вставлять текст, с которым будет сравниваться начало исходного текста.
- Добавлено новое свойство «Текст» в действие «Удалить с начала и конца», используется для исходного текста для обработки.
- Добавлено новое свойство «Текст» в действие «Заменить по шаблону», используется для исходного текста для обработки.
- Добавлено новое свойство «Шаблон» в действие «Заменить по шаблону», используется для обозначения регулярного выражения.
- Добавлено новое свойство «Заменитель» в действие «Заменить по шаблону», используется для нового значения текста, который нужно заменить.
- Добавлено новое свойство «Текст» в действие «Перевернуть», используется для исходного текста для обработки.

#### Группа действий «Функции» и «Текст»

- Удалены действия: » ToUpper», «ToLower» - вместо них используется действие «Установить регистр».
- Удалено действие: «Substring» - вместо него используется действие «Извлечь с позиции».
- Удалено действие: «Replace» - вместо него используется действие «Заменить».
- Удалено действие: «Length» - вместо него используется действие «Получить длину».
- Удалены действия: «FindIndex », «FindLastIndex» - вместо них используется действие «Найти вхождение».
- Удалено действие: «FindByPattern» - вместо него используется действие «Извлечь по шаблону».

#### Группа действий «Windows»

- Добавлено действие «Активировать окно», позволяющее активировать окно.
- Добавлено действие «Максимизировать окно», позволяющее развернуть окно на весь экран.



- Добавлено действие «Свернуть окно», позволяющее свернуть окно.
- Добавлено действие «Закрыть окно, если существует», позволяющее закрыть окно, если оно существует.
- Добавлено действие «Получить имя активного окна», позволяющее сохранить наименование активного окна.
- Добавлено действие «Получить текст элемента», позволяющее получить текст выбранного элемента.
- Добавлено действие «Получить свойство элемента», позволяющее получить значение выбранного свойства элемента.
- Добавлено действие «Получить координаты элемента», позволяющее определить координаты центра элемента.
- Добавлено действие «Дождаться исчезновения элемента», позволяющее ожидать исчезновение объекта по выбранным свойствам.
- Добавлено действие «Сменить язык приложения», позволяющее изменять язык для написания текста.
- Добавлено действие «Сменить язык приложения», позволяющее изменять язык для написания текста.

*Группа действий «Файловая система», подгруппа «Файлы»*

- Добавлено действие «Получить родительскую папку», позволяющее получить путь до родительской папки.
- Добавлено свойство «Данные» в действие «Записать в файл», позволяющее указать данные, которые необходимо записать.
- Добавлено свойство «С новой строки» в действие «Записать в файл», позволяющее добавить данные с новой строки.
- Удалено свойство «Одним элементом» в действии «Прочитать файл».
- Добавлено свойство «Файл» в действие «Проверить признак «Только для чтения»», позволяющее указать путь до файла, который нужно проверить.
- Добавлено свойство «Папка» в действие «Создать файл», позволяющее указать путь до папки в которой нужно создать файл.
- Добавлено свойство «Результат» в действие «Создать файл», позволяющее записывать путь до созданного файла.

## Улучшения

*Группа действий «Диалоговые окна»*

- Исправлена работа действия «Форма ввода данных»
- Исправлена работа действий группы на версиях действий python, java (для всех действий, кроме «Форма ввода данных» - оно представлено только на net)

*Группа действий «Базовые действия»*

- Исправлена работа Цикл «Для каждого» для Java-версии действия

*Общие компоненты студии*

- Исправлено логирование
- Восстановлена публикация роботов и исходников в хранилище
- Исправлена работа действий при переключении версий действий
- Оптимизирована работа студии во время исполнения робота
- Исправлены ошибки при работе с рабочей областью

- Исправлены ошибки связанные с вылетом студии

## 14.1 Общие положения

1. Услуги, оказываемые в рамках технической поддержки, включают в себя:

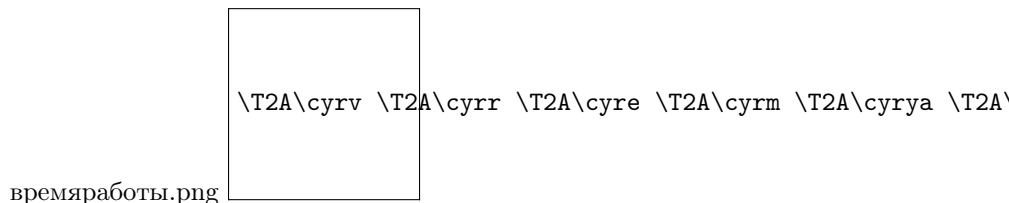
- квалификация и анализ поступивших обращений;
- консультации по вопросам функционирования ПО в соответствии с технической и пользовательской документацией в случае, если требуемая информация не содержится в технической документации к ПО;
- предоставление очередных обновлений ПО, содержащих исправления причин инцидентов, квалифицированных как «ошибка», а также новую/улучшенную функциональность;
- для критических случаев – предоставление внеочередных обновлений ПО содержащих исправления прочих инцидентов, квалифицированных как «ошибка»;
- информирование Заказчика о планах поставки очередных обновлений ПО в течение срока действия настоящего Договора.

2. Прием обращений происходит по следующим каналам:

По электронной почте - email: [support@rpa-robin.ru](mailto:support@rpa-robin.ru)

По телефону +7 (495) 320-61-12

3. Прием и обработка обращений происходит в соответствии с режимом работы поддержки.

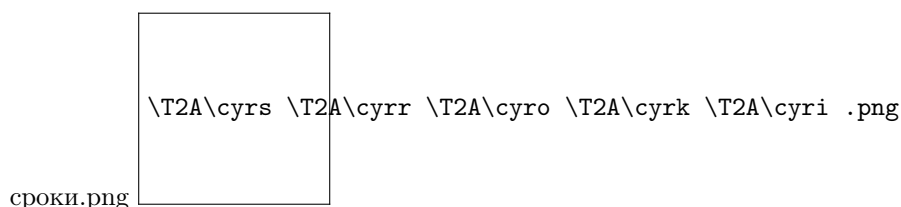


Каждому поступившему обращению присваивается приоритет и может быть установлен крайний срок, не позднее которого обращение должно быть решено.

4. Приоритет запроса присваивается в зависимости от степени воздействия:

- Критическая. Возникновение критической ситуации, влияющей на основной бизнес- процесс. Произошла полная остановка основного (имеющего критическое значение для выполнения необходимых задач) бизнес-процесса, и работа не может быть продолжена надлежащим образом.
- Серьезная. Существенное негативное влияние на основной бизнес-процесс. Произошла утрата в существенной степени возможности пользоваться услугами или существенное ухудшение работы служб, необходимых для основного бизнес-процесса.
- Незначительная. Небольшое негативное влияние на бизнес-процесс. Структуры Заказчика продолжают в существенной степени функционировать с небольшими помехами или без помех в работе служб.
- Общий вопрос. Запрос информации по работе с компонентами Платформы.

## 14.2 Сроки реагирования и время предоставления ответа в зависимости от приоритета запросов



При невозможности решения обращения в указанный срок, может быть предложен временный механизм возобновления работоспособности Системы (work-around) или предоставлен альтернативный срок решения.

## 14.3 Порядок обработки и выполнения запросов

1.1. Запрос считается выполненным в любом из следующих случаев:

1.1.1. Пользователь подтверждает, что запрос выполнен;

1.1.2. Служба технической поддержки выполняет необходимые мероприятия для исправления Ошибки или предоставляет разумное и взаимоприемлемое техническое решение проблемы;

1.1.3. Служба технической поддержки не получает уточняющих сведений или иной информации от Пользователя в течение 5 (пяти) рабочих дней подряд с момента обращения к нему за пояснением запроса.

1.2. Выполнение Запроса может быть отложено или невозможно по следующим причинам:

1.2.1. Ошибка или проблема с ПО не может быть воспроизведена на аппаратном и программном обеспечении Вендора, при этом Пользователь не предоставил удаленный доступ к системе для исследования проблемы;

1.2.2. Пользователь не может предоставить достаточно информации для решения проблемы;

1.2.3. Ошибки или проблемы с ПО являются следствием:

- использования ПО с аппаратным или программным обеспечением, не предназначенным для использования с операционными системами, одобренными в Документации на ПО;
- использования ПО не в соответствии с Документацией на ПО и/или лицензионным / сублицензионным договором или партнерским соглашением.

1.3. Время первоначальной реакции на Запрос составляет до 5 (пяти) рабочих дней

1.4. Время выполнения Запроса составляет до 20 (двадцати) рабочих дней.

## 14.4 Порядок предоставления консультаций

1.1. Консультации предоставляются только Пользователям, прошедшим обучение Вендора (подтверждается сертификатом Вендора о прохождении обучения).

1.2. Консультации не предоставляются по:

- общим вопросам программирования;
- установке и настройке стороннего ПО;
- вопросам разработки роботов под конкретные операции.

1.3. Консультации предоставляются в виде ответов на Запрос, содержащий вопрос, по которому необходима консультация.

1.4. Время первоначальной реакции на Запрос Консультации и время его выполнения аналогично времени, указанному в пп. 1.3 и 1.4 порядка обработки и выполнения запросов.

## 14.5 Порядок обновления

1.1. При выходе новой версии (нового релиза) ПО, Вендор рассылает по электронной почте уведомления об этом следующим компаниям:

1.1.1. Всем Дистрибуторам;

1.1.2. Клиентам, имеющим прямые лицензионные договора с Вендором.

1.2. Уведомления Клиентам, имеющим сублицензионные договора с Дистрибутором, о выходе новой версии (нового релиза) ПО, рассылаются Дистрибутором.

1.3. Клиенты и/или Дистрибуторы, решившие выполнить Обновление, направляют Вендору Запрос, содержащий данное решение.

1.4. В ответ на указанный в п. 1.3 порядка обновления Запрос, Вендор направляет ссылку на дистрибутив новой версии (нового релиза) ПО.