

Утвержден
64509942.00214-01 96 01-ЛУ

Программное средство
«Система централизованного управления
доступом пользователей» (СЦУДП)
«Руководство администратора»
64509942.00214-01 96 01

Листов 412

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

Аннотация

Настоящий документ представляет собой руководство по развертыванию и администрированию программного средства (далее – ПС). Документ предназначен для интеграторов и администраторов ПС и содержит описание процедур установки, настройки и сопровождения комплекса.

Оглавление

Перечень терминов и сокращений	9
1 Сведения о ПС	9
1.1 Назначение и основные возможности ПС	10
1.2 Архитектура ПС	10
1.3 Объектная модель ПС	12
1.4 Алгоритм работы ПС	16
1.5 Комплект поставки	17
1.5.1 Состав и содержание дистрибутива	17
1.5.2 Состав эксплуатационной документации	18
1.6 Условия эксплуатации	18
1.6.1 Требования к квалификации интегратора / администратора	18
1.6.2 Требования к аппаратному обеспечению	19
1.6.3 Требования к программному обеспечению	19
2 Установка ПС	21
2.1 Развертывание веб-интерфейсов ПС	21
2.1.1 Рекомендации по установке программного средства	21
2.1.2 Подготовительные операции: установка необходимого ПО и веб-сервера, настройка соединения с БД	23
2.1.3 Развертывание приложений idm и admin-gui на веб-сервере	29
2.2 Установка ПО, необходимого для взаимодействия ИС с ПС	39

2.2.1	Установка коннекторов на базе сервера коннекторов .NET	40
2.2.2	Установка коннекторов на базе сервера коннекторов JAVA / внутреннего сервера коннекторов	46
3	Описание административного веб-интерфейса ПС	52
3.1	Вход в веб-интерфейс администратора	52
3.2	Описание основных элементов интерфейса	52
3.3	Панель управления ПС: получение сведений о текущем пользователе и состоянии системы	54
3.4	Пользователи: управление пользователями, задание организационной структуры предприятия	55
3.4.1	Управление учетными записями пользователей	55
3.4.2	Задание организационной структуры предприятия	63
3.5	Роли: задание ролевой модели и управление ролями пользователей	66
3.5.1	Создание ролей	67
3.5.2	Управление списком ролей: просмотр, редактирование, удаление ролей	69
3.6	Информационные системы: управление ИС, взаимодействующими с ПС	72
3.6.1	Создание ИС	72
3.6.2	Управление списком ИС: просмотр, поиск, редактирование и удаление ИС	77
3.7	Серверные задачи: управление задачами, выполняемыми на сервере ПС	82
3.7.1	Управление списком задач	83

3.7.2	Редактирование параметров задачи.....	86
3.8	Конфигурирование: настройка работы ПС	87
3.8.1	Загрузка (импорт) объектов в систему	88
3.8.2	Управление объектами репозитория	89
3.8.3	Просмотр и изменение параметров работы ПС.....	95
3.9	Изменение паролей УЗ администратора	102
4	Администрирование ПС	103
4.1	Загрузка кадровых данных	103
4.1.1	Настройка подключения к стандартным источникам данных	103
4.1.2	Настройка объектов: подразделений, должностей, пользователей	111
4.1.3	Задание шаблонов объектов	142
4.2	Управление УЗ пользователей в целевых системах	165
4.2.1	Настройка соответствия полей в ПС полям в подключенной ИС (маппинг атрибутов).....	165
4.2.2	Настройка генерации имен пользователей	172
4.2.3	Настройка запрета повторного использования имен УЗ пользователей.....	174
4.2.4	Настройка генерации паролей.....	175
4.2.5	Настройка отображения карточки пользователя	184
4.2.6	Настройка временной блокировки УЗ пользователя после неуспешных попыток входа	192
4.2.7	Ручная разблокировка УЗ пользователя, заблокированного после неуспешных попыток входа.....	193

4.2.8	Настройка блокировки неактивных УЗ пользователей	194
4.2.9	Задание автоматической блокировки временных УЗ пользователей по окончании периода их действия	195
4.3	Настройка прав доступа (ролей).....	195
4.3.1	Управление доступом к объектам и операциям в ПС	195
4.3.2	Настройка маршрута согласования заявок на предоставление доступа.....	226
4.3.3	Назначение контекстных ролей	249
4.3.4	Назначение прав доступа в контексте должностей (работа с объектом Трудоустройство).....	250
4.3.5	Настройка базовых ролей	255
4.3.6	Настройка наследования ролей	274
4.3.7	Управление полномочиями в подключенных системах	275
4.4	Настройка уведомлений	287
4.4.1	Общие сведения	287
4.4.2	Настройка подключения к почтовому серверу	288
4.4.3	Настройка уведомителей	291
4.5	Настройка взаимодействия ПС и Active Directory.....	316
4.5.1	Минимальные полномочия УЗ для настройки коннектора AD.....	316
4.5.2	Настройка мультидоменной структуры (кросс-ресурсных ассоциаций).....	318

4.6	Настройка взаимодействия ПС и Microsoft Exchange Server	
	330	
4.6.1	Минимальные полномочия УЗ для настройки коннектора Exchange Server	330
4.6.2	Подключение к нескольким Exchange-серверам.....	331
4.7	Настройка серверных задач.....	333
4.7.1	Общие сведения	333
4.7.2	Создание и настройка задач типа Custom.....	334
4.8	Настройка таймаута сессии пользователя	336
4.9	Настройка журналирования событий информационной безопасности	337
4.9.1	Общие сведения	337
4.9.2	Настройка журналирования событий в syslog или определенный файл.....	340
4.9.3	Защита от использования журнала событий безопасности в качестве скрытого канала передачи информации	
	345	
4.9.4	Использование журнала событий безопасности для контроля скрытых каналов передачи информации	346
5	Получение технической поддержки.....	347
Приложение А. Описание событий информационной безопасности .		347
	Вход и выход из системы:	347
	Операции с ресурсными объектами:.....	352
	Операции с пользователями:.....	359
	Операции с трудоустройствами:.....	364
	Операции с назначениями:	368

Операции с ролями:.....	375
Операции с должностями:.....	378
Операции с системными объектами:	382
Операции с серверными задачами:	385
Операции с заявками:.....	389
Факты доступа к объектам:	396
Интеграция с системами:	396
Операции с отчетами:.....	400

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

Таблица 1 – Перечень терминов и сокращений

№	Термин	Определение
1.	АРМ	Автоматизированное рабочее место
2.	БД	База данных
3.	ИС	Информационная система
4.	ОС	Операционная система
5.	ПО	Программное обеспечение
6.	ПК	Программный комплекс
7.	СУБД	Система управления базами данных
8.	УЗ	Учетная запись
9.	УС	Управляемая система
10.	ADSI	Active Directory Service Interface
11.	AGUI	Веб-интерфейс администратора ПС
12.	CLI	Command Line Interface – интерфейс командной строки
13.	CSV	comma separated values, табличные данные в виде значений, разделенных запятой
14.	GUI	Graphical User Interface – графический интерфейс пользователя
15.	HTTP(S)	HyperText Transfer Protocol Secure
16.	ICF	Identity Connector Framework
17.	IdM	Identity Management
18.	JDBC	Java Database Connectivity
19.	SMTP	Simple Mail Transfer Protocol
20.	SQL	Structured query language

1.1 Назначение и основные возможности ПС

ПС — это программный комплекс класса **Identity Management (IdM)**, автоматизирующий процессы контроля и управления правами доступа работников в управляемых системах предприятия. С помощью ПС можно из кадровых систем предприятия получить все имеющиеся сведения о работниках и далее управлять их учётными записями в различных УС предприятия.

ПС получает данные о пользователях из доверенных источников и на основе специальных правил и шаблонов преобразует их в учетные записи, профили и права доступа пользователей. Этому может предшествовать процесс согласования, подтверждения прав и другие активности.

1.2 Архитектура ПС

ПС построен с использованием клиент-серверной архитектуры и состоит из следующих логических компонентов:

- **Веб-интерфейс** – предоставляет доступ пользователей к пользовательскому и административному интерфейсам ПС. Включает в себя следующие программные компоненты:

- о тонкий клиент пользовательского интерфейса;
- о тонкий клиент административного интерфейса.

- **Сервер приложений** – обеспечивает работу пользовательских веб-интерфейсов ПС, взаимодействует с подключенными информационными системами и управляет выполнением фоновых задач. Включает в себя следующие программные компоненты:

- о графический интерфейс пользователя – программное решение, обеспечивающее работу в веб-интерфейсе пользователя;

- о графический интерфейс администратора – программное решение, обеспечивающее работу веб-интерфейса администратора;
- о приложение бизнес-логики и обработки фоновых заданий – программное решение, обеспечивающее целевую работу ОО.

- **Сервер баз данных** – обеспечивает централизованное хранение информации, необходимой для функционирования ОО:

- о база данных – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами ОО.

- **Сервер коннекторов** – обеспечивает связь ПС с системами-источниками данных и целевыми ИС с помощью коннекторов, специальных модулей, которые являются каналами передачи данных. Включает в себя следующие программные компоненты:

- о Сервер коннекторов Java (connector-framework.jar) – программное решение, позволяющее осуществить взаимодействие с системами-источниками данных и целевыми ИС с помощью коннекторов, написанных на языке Java;

- о Сервер коннекторов .NET (ConnectorServer.exe) – программное решение, позволяющее осуществить взаимодействие с системами-источниками данных и целевыми ИС с помощью коннекторов, написанных на языке C#;

- **Коннекторы** – набор независимых программных модулей – коннекторов, предоставляющих интерфейс определенного вида для обеспечения взаимодействия ПС с системами-источниками данных и целевыми ИС.

Схема архитектуры СЦУДП приведена на .

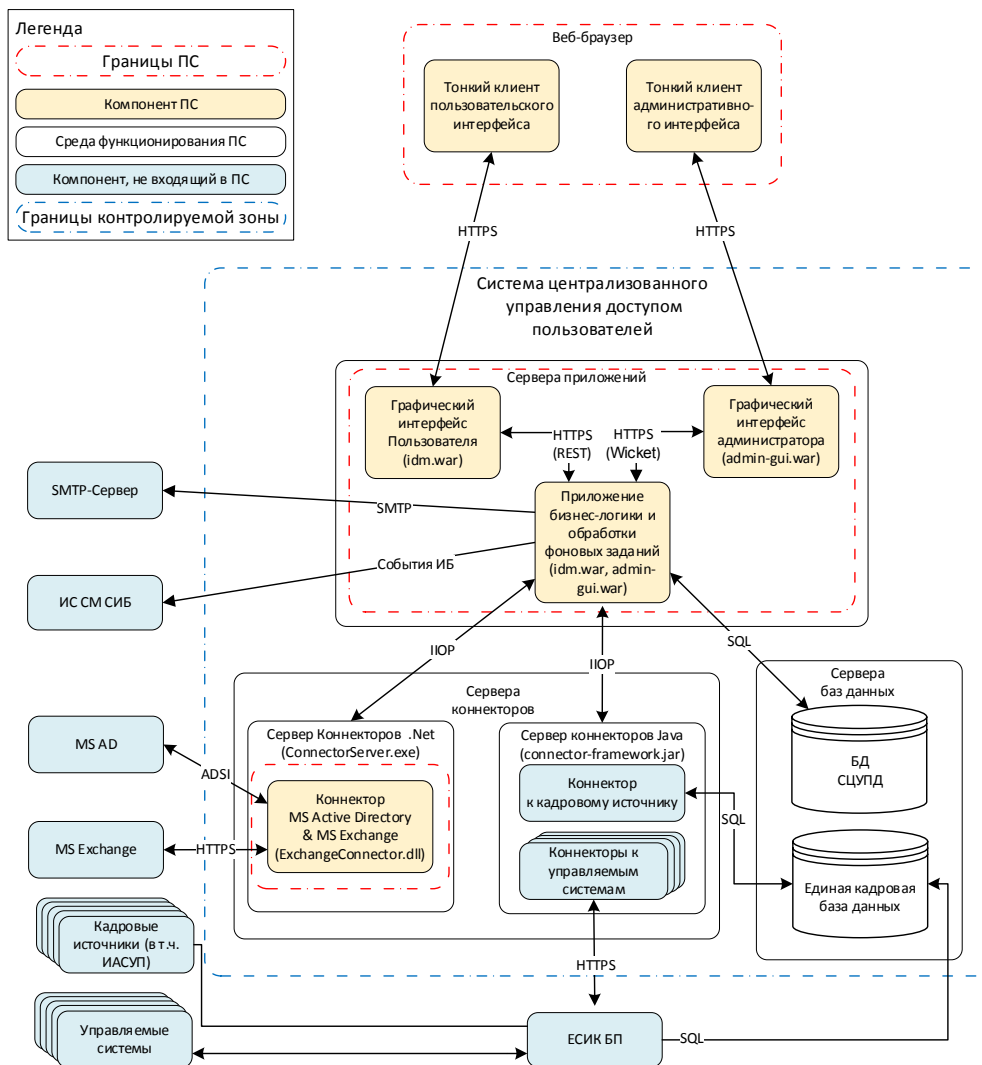


Рис. 1. Схема архитектуры СЦУДП

1.3 Объектная модель ПС

Все данные и метаданные, с которыми работает ПС, являются объектами разных типов. Часть объектов выполняют служебные функции (например, описание маршрута согласования), другие являются проекциями объектов, получаемых из внешних информационных систем, или представляют собой данные, с которыми работает ПС (пользователи, подразделения, роли и др.). Объекты хранятся в базе данных, их можно экспортировать и импортировать в виде XML-документов.

Основными объектами в ПС являются:

- **служебные объекты:**

- о конфигурация системы;
- о шаблон объекта;
- о ресурсный объект.
- **объекты, формирующие организационную структуру**

данных:

- о дивизион;
- о организация;
- о подразделение;
- о должность;
- о трудоустройство;
- о пользователь.
- **объекты, формирующие перечень бизнес-ролей УС:**
- о УС;
- о ИТ-ресурс;
- о бизнес-роль;
- о учетная запись;
- о роль УС.

- **объекты, определяющие работу с заявками:**

- о заявка;
- о элемент заявки;
- о маршрут;
- о шаг.

- **объекты, определяющие перечень операций и объектов, доступных пользователю в Системе:**

- о роль СЦУДП.

- **объекты, определяющие связи между объектами УС и объектами организационной структуры:**

- о назначение.

- **дополнительные объекты:**

- о SoD-конфликт.

Одна из важных функций Системы — автоматическая синхронизация атрибутов между внутренними объектами Системы и данными во внешних системах. Объектами, для которых возможна автоматическая синхронизация, являются пользователь, бизнес-роль, организация, подразделение, должность и трудоустройство.

ПС может получать данные из кадровых и управляемых систем в нескольких режимах: ручном, по заданному расписанию или в режиме реального времени. Последний режим позволяет незамедлительно получать изменения из информационных систем, обеспечивая тем самым непрерывность процесса управления и контроля прав доступа работников. При этом Система взаимодействует с системами-источниками данных и УС с помощью коннекторов. Настройки взаимодействия (параметры подключения, схемы данных, правила обработки этих данных, правила синхронизации) с конкретной системой задаются в Системе в XML-описании этой системы.

В соответствии с заданными настройками, Система преобразует полученные данные в специальную структуру Ресурсный объект, что позволяет работать с объектами различных УС в едином формате. В Ресурсном объекте хранятся атрибуты УС – общие унифицирующие и специальные, определяемые для каждой УС отдельно.

При передаче данных применяются специальные правила, которые настраиваются при описании УС. Эти правила задают соответствие между данными в Системе и данными в других системах. Если данные передаются из других систем в нужные атрибуты конкретного объекта Системы, то срабатывают правила переноса данных из Ресурсного объекта в атрибуты объектов (inbound mapping). При обратной передаче срабатывают правила outbound mapping.

Обработка данных объекта в самой Системе выполняется в соответствии с правилами, заданными в шаблоне объектов. Для различных видов объектов можно задать разные шаблоны, где описаны

алгоритмы и правила обработки атрибутов объекта, которые будут применяться ко всем объектам одного вида.

Общий процесс взаимодействия ПС и ИС включает этапы:

- 1) Загрузка данных из систем-источников и преобразование полученной информации в формат Ресурсных объектов.
- 2) Перенос данных из **Ресурсных объектов** в атрибуты объекта ПС в соответствии с правилами **inbound mapping**.
- 3) Обработка данных объекта ПС в соответствии с правилами, заданными в шаблоне объектов.
- 4) Установление связей между объектами ПС и объектами целевой системы. Например, установление связи конкретного объекта Пользователь с конкретным аккаунтом в целевой системе.
- 5) Перенос данных в соответствии с правилами **outbound mapping** и их преобразование в Ресурсные объекты.
- 6) Загрузка данных из ПС в целевую систему.
- 7) Реконсиляция данных.

Примечание

Реконсиляция — это установка соответствия полномочий пользователей в ПС и целевых системах.

8) Загрузка данных из целевых систем и преобразование полученной информации в формат Ресурсных объектов.

9) Перенос данных из Ресурсных объектов в атрибуты объекта ПС в соответствии с правилами **inbound mapping**.

Настройка взаимодействия ПС с ИС предполагает:

- настройку подключения к источникам данных;
- задание шаблонов объектов (в том числе настройку генерации имен пользователей и паролей для объекта **Пользователь**);
- определение и настройку правил **inbound mapping** и **outbound mapping**.

1.4 Алгоритм работы ПС

ПС обрабатывает запросы на изменение учетных данных, инициируя требуемые бизнес-процессы в соответствии с конфигурацией, в результате которых выполняются необходимые изменения в подключенных системах и рассылаются уведомления о произошедших событиях.

Запросы на изменение учетных данных поступают в систему:

- в виде заявок, отправляемых пользователем из веб-интерфейса.

Пользователь системы может напрямую запросить актуальные данные того или иного работника из карточки пользователя. Также пользователь может отправить заявку на добавление каких-либо ролей, которые будут предоставлены после успешного процесса согласования.

- из внешних систем, подключенных в качестве источника данных (например, из кадровых систем).

Из таких систем поступают данные о новых работниках, изменениях их кадровых данных, переводах в организационно-штатной структуре, отпусках, увольнениях.

В зависимости от характера запрашиваемых изменений, ПС инициирует бизнес-процессы согласно настройкам (ПС предоставляет инструменты для настройки бизнес-процессов). С помощью ПС можно настроить следующие бизнес-процессы:

- **прием работника на работу:** из кадровой системы (или по запросу пользователя из веб-интерфейса) поступает информация о новом работнике. С помощью ПС создается профиль этого работника, назначается набор базовых ролей для него, отправляются заявки на дополнительные роли, создаются учетные записи в соответствующих системах, предоставляются необходимые полномочия;

- **получение изменений в кадровых данных:** из кадровой системы (или по запросу пользователя из веб-интерфейса) поступает информация об изменениях кадровых данных работника. В ПС обновляются соответствующие атрибуты учетных записей работника;
- **перевод по должности или в другое подразделение:** из кадровой системы (или по запросу пользователя из веб-интерфейса) поступает информация о переводе работника по должности или в другое подразделение. ПС инициирует процесс отзыва лишних полномочий (выданных в рамках предыдущей должности) и предоставления новых полномочий (требуемых в рамках новой должности);
- **уход работника в декретный отпуск:** из кадровой системы поступает информация об уходе работника в декретный отпуск. ПС инициирует блокировку учетных записей работника в информационных системах. По окончании декретного отпуска (по сигналу из кадровой системы) ПС разблокирует учетные записи;
- **увольнение работника:** из кадровой системы (или по запросу пользователя из веб-интерфейса) поступает информация об увольнении работника. ПС инициирует отзыв всех ролей работника и блокировку его учетных записей.

В ходе выполнения бизнес-процессов обычно рассылаются уведомления заинтересованным лицам о шагах выполнения и произошедших событиях. В результате выполнения бизнес-процессов ПС инициирует изменения в подключенных системах (добавление, модификация, удаление, блокировка/разблокировка учетных записей/групп/ролей и других объектов системы).

1.5 Комплект поставки

1.5.1 Состав и содержание дистрибутива

В состав дистрибутива ПС входят следующие установочные файлы,

- файлы, собираемые из кода, разработанного предприятием-изготовителем:

- o idm.war;
- o admin-gui.war;
- o connid-connector-server-java-1.4.4.0-SNAPSHOT.zip;
- o connid-dotnet-1.5.1.0.msi;
- o connector-csvfile.jar;
- o Exchange.Connector-1.4.7.5.zip.

1.5.2 Состав эксплуатационной документации

В поставку ПС входят следующие эксплуатационные документы:

- Руководство по администрированию – содержит подробную информацию по установке, настройке и сопровождению ПС (настоящий документ);
- Руководство пользователя – содержит подробную информацию по использованию ПС.

1.6 Условия эксплуатации

1.6.1 Требования к квалификации интегратора / администратора

Квалификация интегратора / администратора ПС должна быть достаточной для выполнения задач по обслуживанию комплекса, обеспечивающих бесперебойное функционирование всех его компонентов.

К задачам интегратора / системного администратора ПС относятся:

- установка и настройка компонентов ПС;

- мониторинг процессов и своевременное реагирование на служебные уведомления ПС.

Интегратор / администратор ПС должен:

- обладать необходимыми знаниями и навыками администрирования используемых ОС и СУБД;
- ориентироваться в особенностях работы ПС;
- иметь навыки программирования;
- иметь базовые знания языка PL/SQL.

1.6.2 Требования к аппаратному обеспечению

1.6.2.1 Серверная часть

Для функционирования ПС требуется оборудование со следующими минимальными характеристиками:

- 4-ядерный процессор Intel Xeon с тактовой частотой 2.5 ГГц;
- объем оперативной памяти – 12 ГБ;
- объем жесткого диска – 50 ГБ.

1.6.2.2 Клиентская часть

АРМ пользователя / администратора ПС должно быть оборудовано компьютером, обладающим следующими минимальными характеристиками:

- 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 2 ГГц;
- объем оперативной памяти – 2 ГБ;
- объем жесткого диска – 20 ГБ.

Кроме того, в состав аппаратного обеспечения должен входить сетевой адаптер Ethernet.

1.6.3 Требования к программному обеспечению

1.6.3.1 Серверная часть

Для функционирования ПС на серверном оборудовании должно быть установлено следующее программное обеспечение:

- Операционная система (см. Таблица 2 –);
- СУБД – PostgreSQL версии 10;
- Сервер приложений – Apache Tomcat версии 7, 8, 9;
- Серверы коннекторов – Connector Server .NET (для .net-коннекторов) и Connector Server Java (для .java-коннекторов);
- JRE (Java 8).

Таблица 2 – Перечень совместимых ОС

Наименование операционной системы	Версии
Microsoft Windows	Microsoft Windows Server 2016;
Astra Linux	1.6 (релиз «Смоленск»)
ALT Linux	Альт 8 СП, Альт Сервер 9
CentOS	7

1.6.3.2 Клиентская часть

Для корректной работы клиентской части (веб-приложения) ПС требуется компьютер под управлением одной из совместимых ОС (см. Таблица 2 –).

В состав программного обеспечения компьютера должна входить программа-клиент, предоставляющая пользователю возможность навигации и просмотра web-ресурсов (браузер). Рекомендуемые браузеры:

- Internet Explorer (версии 8.0, 9.0, 11.0);
- Mozilla Firefox (версии 50, 52-77);
- Google Chrome (версии 64-75).

Кроме того, рекомендуется в настройках браузера разрешить выполнение **javascript** и сохранение файлов **cookies**.

1.1. Приемка поставленного средства

Подробные правила приемки изложены в технических условиях RU.64509942.00214-01 91 01 в разделе 2 Правила приемки. В общем случае производится проверка контрольных сумм дистрибутива ПС.

Полученные контрольные суммы должны соответствовать эталонным контрольным суммам ПС, приведенным в формуляре RU.64509942.00214-01 30 01, поставляемом вместе с дистрибутивом ПС.

2 УСТАНОВКА ПС

2.1 Развертывание веб-интерфейсов ПС

2.1.1 Рекомендации по установке программного средства

Существует несколько видов конфигураций развертывания ПС (см. Рис. 1), разработанных с учетом простоты развертывания и обслуживания, производительности, доступности и масштабируемости компонентов. Описание основных рекомендуемых стратегий развертывания приведено в таблице ниже (см. Таблица 3 –).

Таблица 3 – Стратегии развертывания ПС

Конфигурация	Назначение	Особенности
Простая: один сервер приложений, один сервер БД	• пилотные проекты; • непродуктивные тестовые среды	• простота развертывания; • экономичность; • невысокая доступность компонентов
Средняя: несколько серверов приложений, один сервер БД	продуктивные среды в небольших организациях	доступность серверов приложений

Продвинутая: несколько групп серверов приложений (Frontend, фоновые задачи), несколько серверов БД	продуктивные среды в крупных организациях	- гибкость; - масштабируемость (за счет добавления дополнительных экземпляров серверов приложений и СУБД); - высокая производительность; - высокая доступность всех компонентов (возможно настроить автоматическое переключение в случае отказа одного из компонентов системы)

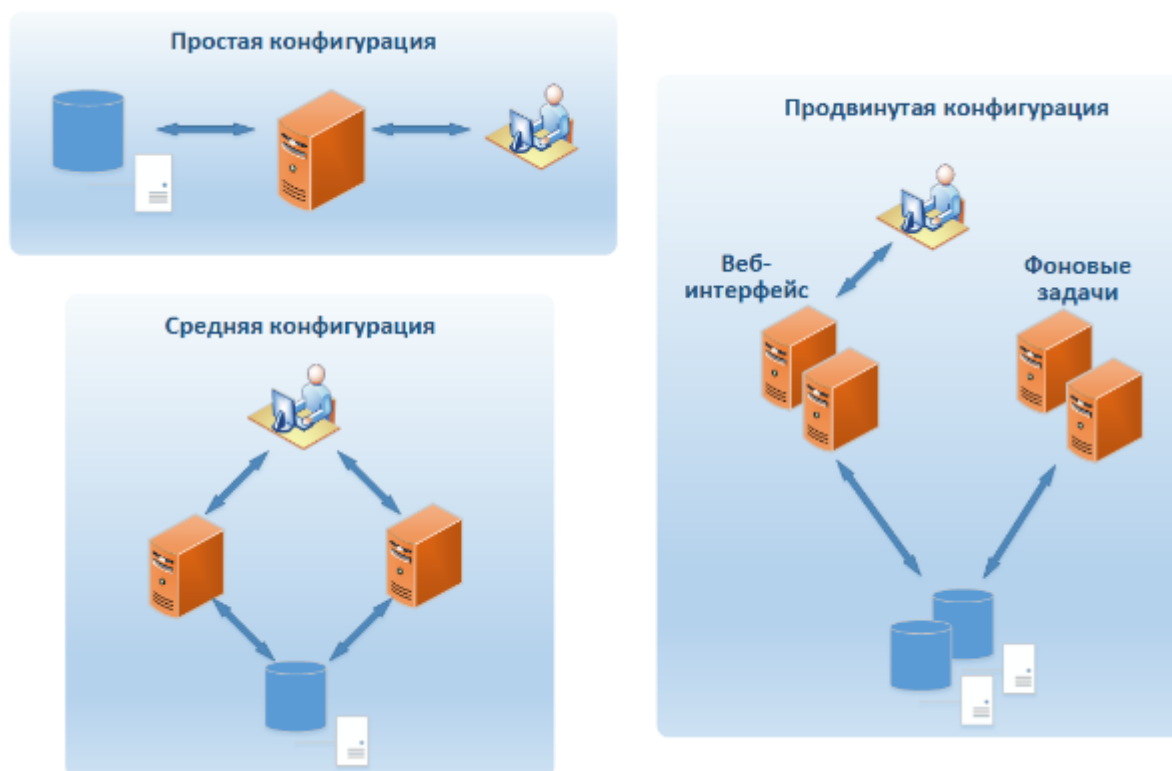


Рис. 1 – Конфигурации развертывания ПС

При ожидаемой высокой нагрузке рекомендуется распределить функциональные роли ПС по нескольким серверам приложений. Наиболее распространенным способом разделения кластера из двух серверов приложений является использование одного из них для

взаимодействия пользователей с ПС (основной сервер), а второго – для обработки фоновых/пакетных задач. Второй сервер также может служить как вспомогательный веб-интерфейс на случай перегрузки или отказа основного сервера.

2.1.2 Подготовительные операции: установка необходимого ПО и веб-сервера, настройка соединения с БД

2.1.2.1 Настройка hostname

За рекомендациями по настройке hostname следует обратиться к документации по администрированию соответствующей ОС.

2.1.2.2 Установка и настройка ПО Java (Linux)

Для корректного функционирования ПС необходимо на сервере приложений:

1) Установить ПО JRE. Пакеты для установки можно загрузить с официального сайта разработчика:
<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>.

Примечание

Для ОС **Astra Linux**, следует загружать архив **tar.gz** (например, **jre-8u111-linux-i586.tar.gz**).

2) Создать пользователя **tomcat** и переключиться на работу от его имени:

```
adduser tomcat  
su - tomcat
```

3) Записать в файл **~/.bashrc** системную переменную:
JRE_HOME = <путь к JRE>

Примечание

Прописать системную переменную можно на уровне системы в файле **/etc/environment** (операция выполняется от имени пользователя **root**). В этом случае прописывать системную переменную в файле **~/.bashrc** не требуется.

2.1.2.3 Установка и настройка ПО Java (Windows)

Для корректного функционирования ПС необходимо на сервере приложений:

1) Установить ПО JRE. Пакеты для установки можно загрузить с официального сайта разработчика: <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>.

2) Задать системную переменную **JRE_HOME** (открыть окно **Компьютер > Свойства > Дополнительные параметры системы**, выбрать вкладку **Дополнительно**, нажать кнопку **Переменные среды...** и в блоке **Системные переменные** нажать кнопку **Создать...**):
JRE_HOME = <путь к JRE>

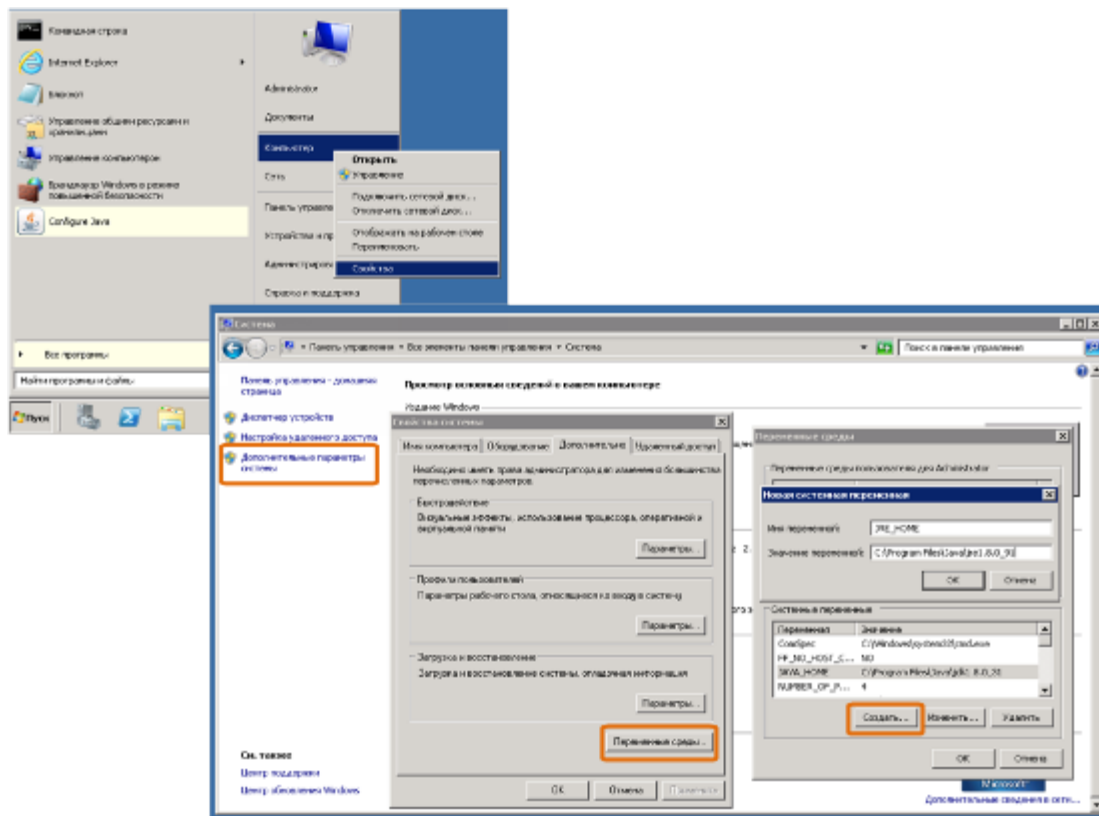


Рис. 2. Создание системной переменной

Примечание

Чтобы новые переменные среды вступили в силу, необходимо перезагрузить компьютер.

2.1.2.4 Установка СУБД и создание БД ПС (Windows)

Всю служебную информацию ПС хранит в базе данных (PostgreSQL).

Чтобы развернуть сервер БД в ОС семейства Windows, необходимо:

1) 1. Загрузить требуемую версию СУБД PostgreSQL с официального сайта разработчика <https://www.postgresql.org/download/> и установить ее, следуя инструкциям, выдаваемым в процессе установки.

2) Создать новую базу данных с именем **idm** и остальными параметрами по умолчанию.

3) Для создания БД необходимо выполнить запуск ПО **pgAdmin** (Пуск > Все программы > PostgreSQL[версия_PostgreSQL] > pgAdmin[версия pgAdmin]) и в открывшемся окне интерфейса **pgAdmin** выполнить следующие действия:

- подключиться к требуемому серверу БД;
- в контекстном меню элемента **Базы данных** выбрать пункт **Новая база данных...**;
- в открывшемся окне в поле **Имя** ввести имя БД и нажать кнопку **ОК**.

4) При необходимости добавить в файл <каталог, где установлена PostgreSQL>/.../data/pg_hba.conf следующую строку:

```
| host all all <IP>/32 md5
```

где <IP> – IP-адрес узла, с которого будет выполняться подключение к БД. Если предполагается доступ с различных IP-

адресов, необходимо добавить аналогичные строки для каждого IP-адреса.

5) При необходимости добавить в файл <каталог, где установлена PostgreSQL>/.../postgresql.conf следующую строку:

```
| listen_addresses = '*'
```

2.1.2.5 Установка СУБД и создание БД ПС (Linux)

Чтобы развернуть сервер БД в ОС семейства Linux, необходимо:

1) Загрузить требуемую версию СУБД PostgreSQL с официального сайта разработчика <https://www.postgresql.org/download/> и установить ее, следуя инструкциям, выдаваемым в процессе установки.

2) Перейти в соответствующий каталог, выполнить команду

```
| # POSTGRESQL_HOME/install.sh
```

и следовать инструкциям, выдаваемым в процессе установки.

Примечание

В процессе установки СУБД указываются:

- <host> – адрес узла, на котором функционирует БД,
- <login> и <password> – имя и пароль пользователя БД.

3) Создать новую базу данных с именем idm и остальными параметрами по умолчанию:

```
| $psql  
| create database idm;
```

Примечание

В конце строки команды создания БД знак «;» обязателен.

4) При необходимости добавить в файл <каталог, где установлена PostgreSQL>/.../data/pg_hba.conf следующую строку:

```
| host all all <IP>/32 md5
```

где <IP> – IP-адрес узла, с которого будет выполняться подключение к БД. Если предполагается доступ с различных IP-адресов, необходимо добавить аналогичные строки для каждого IP-адреса.

5) При необходимости добавить в файл <каталог, где установлена **PostgreSQL**>/.../**postgresql.conf** следующую строку:

```
| listen_addresses = '*'
```

2.1.2.6 Установка сервера Apache Tomcat

Для установки **Apache Tomcat** следует:

1) Загрузить zip-архив, содержащий актуальную версию **Apache Tomcat**, с официального сайта разработчика **<http://tomcat.apache.org>**.

2) Загрузить tar.gz-архив, содержащий актуальную версию **Apache Tomcat**, с официального сайта разработчика **<http://tomcat.apache.org>**.

3) Распаковать архив в любой каталог по усмотрению администратора. Далее в руководстве путь к этому каталогу будет обозначаться **TOMCAT_HOME**.

4) С помощью команды

```
| tar -xvf <название архива>
```

распаковать архив в любой каталог по усмотрению администратора. Далее в руководстве путь к этому каталогу будет обозначаться **TOMCAT_HOME**.

5) Отредактировать файл **TOMCAT_HOME\bin\catalina.bat**, удалив кавычки (при их наличии) в следующих строках:

```
| :noJuliConfig
| set "JAVA_OPTS=%JAVA_OPTS% %LOGGING_CONFIG%"
|
| :noJuliManager
| set "JAVA_OPTS=%JAVA_OPTS% %LOGGING_MANAGER%"
```

6) Настроить для пользователя **tomcat** права доступа к каталогу **TOMCAT_HOME**:

```
# chown tomcat:tomcat TOMCAT_HOME -R
```

2.1.2.7 Настройка соединения с базой данных

Для настройки соединения с БД **idm** следует:

1) Загрузить JDBC41 Postgresql Driver / JDBC42 Postgresql Driver со страницы <https://jdbc.postgresql.org/download.html> в каталог **TOMCAT_HOME\lib**.

2) Загрузить JDBC41 Postgresql Driver / JDBC42 Postgresql Driver со страницы <https://jdbc.postgresql.org/download.html> в каталог **TOMCAT_HOME\lib** и выполнить команду

```
# chown tomcat:tomcat <файл драйвера>
```

3) При необходимости (если используется внешняя БД) отредактировать файл **TOMCAT_HOME\conf\server.xml**, указав при объявлении ресурса уровень изоляции транзакций:

```
defaultTransactionIsolation="READ_COMMITTED"
```

Примечание

Под уровнем изоляции транзакций понимается степень обеспечиваемой внутренними механизмами СУБД (то есть не требующей специального программирования) защиты от всех или некоторых видов несогласованностей данных, возникающих при параллельном выполнении транзакций. Определены четыре уровня изоляции: **Read uncommitted**, **Read committed**, **Repeatable read**, **Serializable**. Первый из них является самым слабым, последний — самым сильным, каждый последующий включает в себя все предыдущие. В большинстве СУБД используется уровень **Read committed** (чтение зафиксированных данных). Более подробные сведения об уровнях изоляции транзакций можно получить в соответствующей литературе.

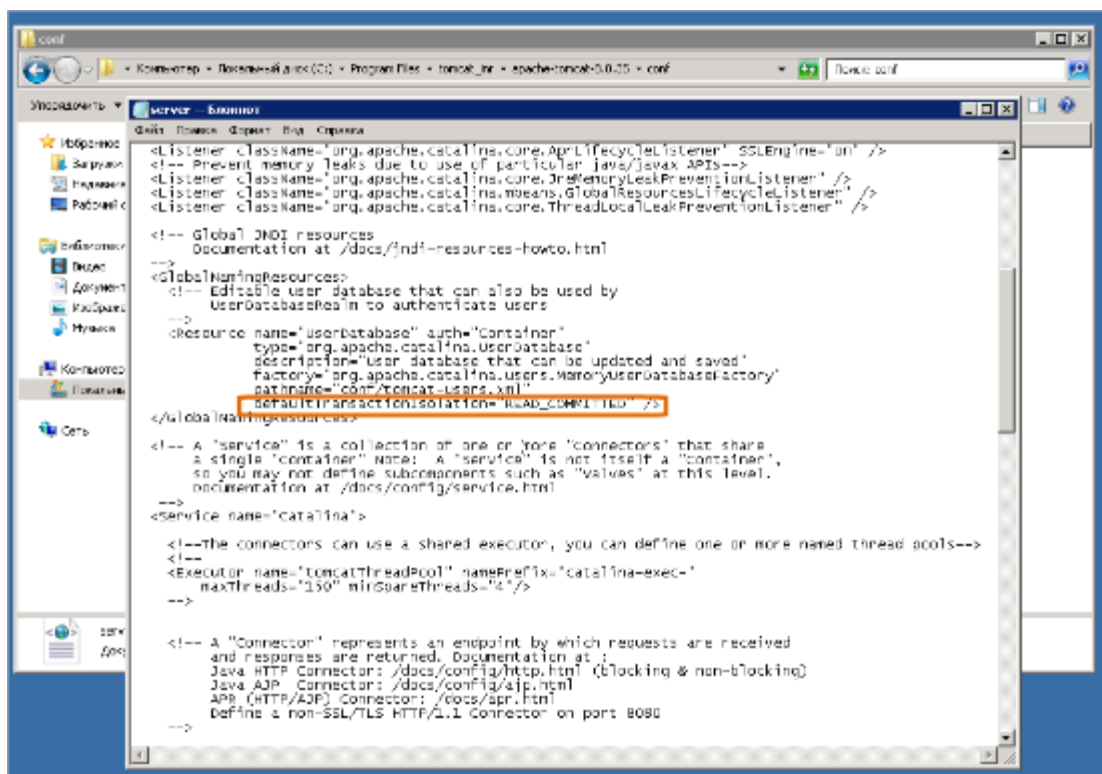


Рис. 2 – Задание уровня изоляции транзакций

4) Добавить в раздел **GlobalNamingResources** файла **ТОМСАТ_HOME\conf\server.xml** следующий блок:

```
<Resource name="jdbc/idm" auth="Container"
type="javax.sql.DataSource"
    username="<login>" password="<password>"
    url="jdbc:postgresql://<host>:5432/idm"
    driverClassName="org.postgresql.Driver"
    accessToUnderlyingConnectionAllowed="true"
    defaultTransactionIsolation="READ_COMMITTED"
    initialSize="5" maxWait="5000"
    maxActive="30" maxIdle="5"
    validationQuery="select 1"
    poolPreparedStatements="true"/>
```

где:

- <host> – адрес узла, на котором функционирует БД,
- <login> и <password> – имя и пароль пользователя, которые были указаны при установке СУБД.

5) Отредактировать блок **Context** в файле **TOMCAT_HOME\conf\context.xml** следующим образом:

```
<?xml version='1.0' encoding='utf-8'?>
<Context>
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <ResourceLink name="jdbc/idm"
                  global="jdbc/idm"
                  type="javax.sql.DataSource"/>
</Context>
```

2.1.3 Развертывание приложений idm и admin-gui на веб-сервере

Установка ПС предполагает развертывание на веб-сервере приложений, разработанных предприятием-изготовителем, idm.war и admin-gui.war. В результате пользователю будут доступны веб-интерфейсы idm (пользовательский интерфейс) и admin-gui (интерфейс администратора).

ВАЖНО! Приложения **idm** и **admin-gui** не могут быть развернуты на одном сервере Tomcat, поскольку это приводит к конфликтам, препятствующим нормальному функционированию ПС. Поэтому требуется установить два веб-сервера **Apache Tomcat**, которые будут работать на разных портах.

2.1.3.1 Развертывание приложения ПС

Для развёртывания приложения **idm** необходимо:

- 1) Распаковать idm.war в каталог TOMCAT_HOME/webapps/idm.
- 2) Проверить, что в каталоге отобразилось дерево каталогов.
- 3) Выполнить команду

```
# chown tomcat:tomcat idm.war
```

4) Добавить в начало файла
TOMCAT_HOME\bin\catalina.bat**TOMCAT_HOME/bin/catalina.sh**
следующую строку:

```
CATALINA_OPTS = "-Ddm.home=/home/tomcat/idm
-
Djavax.net.ssl.trustStore=/home/tomcat/idm/keystore
.jceks
-Djavax.net.ssl.trustStoreType=jceks -Xms128m
-Xmx1024m -XX:PermSize=64m -XX:MaxPermSize=256m
-Ddm.nodeId=admin-gui -
Dcom.sun.management.jmxremote
-
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.port=5012 -
Duser.country=RU -Duser.language=ru
-
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5002"
set JAVA_OPTS=%JAVA_OPTS% -Ddm.home="c:/idm"
-Djavax.net.ssl.trustStore=c:/idm/keystore.jceks -
Djavax.net.ssl.trustStoreType=jceks
-Xms128m -Xmx1024m -XX:PermSize=64m -
XX:MaxPermSize=256m -Ddm.nodeId=main -Xdebug
-
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5001
-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.port=5011
-Duser.country=RU -Duser.language=ru
```

ВАЖНО! Строка не должна содержать символов перевода каретки.

5) В файле **TOMCAT_HOME\bin\service.bat** задать значение параметра **-JvmOptions** команды **%EXECUTABLE%** **//IS//%SERVICE_NAME%** следующим образом:

```
--JvmOptions "-Dcatalina.home=%CATALINA_HOME%;-
Dcatalina.base=%CATALINA_BASE%;
-Djava.io.tmpdir=%CATALINA_BASE%\temp;
-
Djava.util.logging.manager=org.apache.juli.ClassLoa
derLogManager;
-
Djava.util.logging.config.file=%CATALINA_BASE%\conf
\logging.properties;
-Didm.home="c:/idm";-
Djavax.net.ssl.trustStore=c:/idm/keystore.jceks;
-Djavax.net.ssl.trustStoreType=jceks;-Xms128m;-
Xmx1024m;-XX:PermSize=64m;
-XX:MaxPermSize=256m;-Didm.nodeId=main;-
Dcom.sun.management.jmxremote;
-Dcom.sun.management.jmxremote.authenticate=false;
-Dcom.sun.management.jmxremote.ssl=false;-
Dcom.sun.management.jmxremote.port=5011;
-Duser.country=RU;-Duser.language=ru"
```

ВАЖНО! Строка со значением параметра – **JvmOptions** не должна содержать символов перевода каретки.

6) Установить службу **Apache Tomcat**, для чего в CLI выполнить команду:

```
cd <TOMCAT_HOME>\bin
service.bat install <service_name>
```

где **service_name** – имя устанавливаемой службы Tomcat для веб-приложения, например, **Tomcatln**.

7) Запустить установленную службу **Tomcat** с помощью ммс-оснастки **Services (Службы)**.

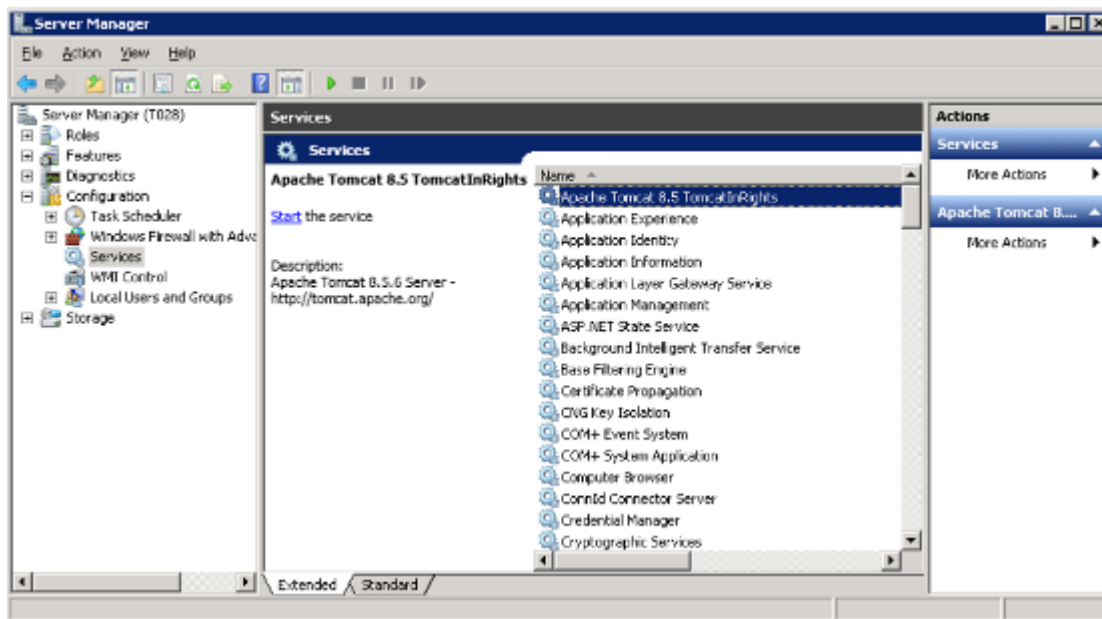


Рис. 3 – Запуск службы Tomcat

Примечание

Ход работы службы записывается в журнальный файл **TOMCAT_HOME/logs/commons-daemon.<текущая_дата>.log**.

Появление в журнальном файле строки вида:

```
[info] [1596] Service 'Tomcat' installed
```

означает, что служба установлена.

Ход операций запуска сервера и инициализации ПС записывается в журнальные файлы **TOMCAT_HOME/logs/idm.log** и **TOMCAT_HOME/logs/catalina.log**. Появление в журнальном файле **catalina.log** строки вида:

```
INFO [main] org.apache.catalina.startup.Catalina.start Server startup in ... ms
```

означает, что сервер запущен и ПС инициализирован.

При необходимости сервер может быть запущен как приложение, в этом случае для запуска сервера в CLI следует выполнить команды:

```
cd TOMCAT_HOME
```

```
bin\startup.bat
```

8) При первом запуске сервера создается каталог **C:/IDM_HOME**.

Создание каталога может занять некоторое время (около 1 минуты).

В этом каталоге хранятся следующие файлы и подкаталоги:

- **config.xml** – файл, в котором хранятся параметры базовой конфигурации ПС:
 - о параметры подключения к БД;
 - о путь к каталогу с дополнительными коннекторами, которые загружаются в локальный сервер коннекторов ПС;
 - о путь к хранилищу ключей, которыми шифруются пароли в ПС.
- **keystore.jceks** – файл с ключами для шифрования паролей в ПС. Расположение этого файла может задаваться в файле **config.xml** или при запуске Tomcat с помощью ключа **-Djavax.net.ssl.trustStore=**
 - **export** – каталог, в котором хранятся файлы, генерируемые ПС (например, отчёты);
 - **icf-connectors** – каталог для дополнительных коннекторов, которые должны работать на локальном сервере ПС. Расположение этого каталога может задаваться в файле **config.xml**;
 - **schema** – каталог для файлов, описывающих расширения объектной модели ПС;
 - **tmp** – каталог для временных файлов ПС;
 - **web** – каталог, в котором хранятся данные, используемые при кастомизации внешнего вида ПС.

9) Запустить **Apache Tomcat**, для чего выполнить команду **TOMCAT_HOME/bin/startup.sh**.

Примечание

Ход операций запуска сервера и инициализации ПС записывается в журнальные файлы **TOMCAT_HOME/logs/idm.log** и **TOMCAT_HOME/logs/catalina.log**. Появление в журнальном файле **catalina.log** строки вида

INFO [main] org.apache.catalina.startup.Catalina.start Server startup in ... ms

означает, что сервер запущен и ПС инициализирован.

При первом запуске сервера создается каталог **/home/tomcat/idm_home**. Создание каталога может занять некоторое время (около 1 минуты).

В этом каталоге хранятся следующие файлы и подкаталоги:

- **config.xml** – файл, в котором хранятся параметры базовой конфигурации ПС:
 - о параметры подключения к БД;
 - о путь к каталогу с дополнительными коннекторами, которые загружаются в локальный сервер коннекторов ПС;
 - о путь к хранилищу ключей, которыми шифруются пароли в ПС.
- **keystore.jceks** – файл с ключами для шифрования паролей в ПС. Расположение этого файла может задаваться в файле **config.xml** или при запуске Tomcat с помощью ключа **-Djavax.net.ssl.trustStore=**
 - **export** – каталог, в котором хранятся файлы, генерируемые ПС (например, отчёты);
 - **icf-connectors** – каталог для дополнительных коннекторов, которые должны работать на локальном сервере ПС. Расположение этого каталога может задаваться в файле **config.xml**;
 - **schema** – каталог для файлов, описывающих расширения объектной модели ПС;
 - **tmp** – каталог для временных файлов ПС;
 - **web** – каталог, в котором хранятся данные, используемые при кастомизации внешнего вида ПС.

10) После первого запуска отредактировать файл **IDM_HOME/config.xml** так, чтобы тег **<repository>** содержал следующий код:

```
<repository>  
    <repositoryServiceFactoryClass>
```

```
ru.solarsecurity.inrights.repo.sql.SqlRepositoryFactory
</repositoryServiceFactoryClass>
<embedded>
    false
</embedded>
<hibernateDialect>

ru.solarsecurity.inrights.repo.sql.util.InRightsPostgreSQLDialect
</hibernateDialect>
<hibernateHbm2ddl>
    validate
</hibernateHbm2ddl>
<lockForUpdateViaHibernate>
    true
</lockForUpdateViaHibernate>
<dataSource>
    java:comp/env/jdbc/idm
</dataSource>
</repository>
```

11)Перезапустить сервер **Apache Tomcat** (перезапустить службу **Tomcat**).

Примечание

Если сервер был запущен как приложение, для его перезапуска следует:

- a. В диспетчере задач Windows остановить процесс и закрыть окно диспетчера;
- b. В CLI выполнить команды:

```
cd TOMCAT_HOME
bin\startup.bat
```

12. Перезапустить **Apache Tomcat**, для чего выполнить команды:


```
TOMCAT_HOME/bin/shutdown.sh  
TOMCAT_HOME/bin/startup.sh
```

После успешно выполненных операций пользовательский веб-интерфейс ПС будет доступен по адресу **http://<host>:ZZZZ/idm**, где **<host>** – адрес сервера, на котором был установлен ПС. Имя пользователя и пароль по умолчанию – **administrator/5ecr3t**. Описание пользовательского веб-интерфейса приведено в документе Руководство пользователя.

2.1.3.2 Развертывание приложения admin-gui.war

Развертывание приложения **admin-gui.war** проходит аналогично развертыванию **idm.war** за исключением следующих моментов:

- требуется установить дополнительный веб-сервер **Apache Tomcat**, который будет работать на порту, отличном от того, на котором работает веб-сервер с **idm.war**;
- дополнительный веб-сервер должен быть настроен на каталог **idm_HOME**, созданный при развертывании **idm.war**;
- операции записываются в журнальный файл **TOMCAT_HOME/logs/idm.log**.

Для развёртывания приложения **admin-gui** необходимо:

- 1) Создать копию каталога **TOMCAT_HOME**. Далее путь к этому каталогу будет обозначаться **TOMCAT_HOME_ADMGUI**.
- 2) Удалить каталог **TOMCAT_HOME_ADMGUI/webapps/inrights**.
- 3) 3. Распаковать **admin-gui.war** в каталог **TOMCAT_HOME_ADMGUI/webapps/admin-gui**.
- 4) Проверить, что в каталоге отобразилось дерево каталогов.
- 5) Все библиотеки из файла **admin-giu.zip** распаковать в каталог **TOMCAT_HOME_ADMGUI/webapps/admin-gui/WEB-INF/lib**. В результате для распакованных файлов должен получиться следующий

путь **TOMCAT_HOME_ADMGUI/webapps/admin-gui/WEB-INF/lib/%название файла%.**

6) Выполнить команду

```
# chown tomcat:tomcat admin-gui.war
```

7) В файле

TOMCAT_HOME_ADMGUI/bin/catalina.sh**TOMCAT_HOME_ADMGUI\bin\catalina.bat:**

- значение ключа **-Dinrights.nodeld** заменить с **main** на **admin-gui**

- значение ключа **-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=** заменить с 5001 на 5002

- значение ключа **-Dcom.sun.management.jmxremote.port** заменить с 5011 на 5012

8) В файле **TOMCAT_HOME_ADMGUI/conf/server.xml** все значения портов (значения параметров **port** и **redirect_port**), кроме порта БД, увеличить на единицу.

9) В файле **TOMCAT_HOME_ADMGUI/bin/catalina.sh****TOMCAT_HOME_ADMGUI\bin\service.bat** задать значение параметра **-JvmOptions** команды **%EXECUTABLE%" //IS//%SERVICE_NAME%** следующим образом:

```
--JvmOptions "-Dcatalina.home=%CATALINA_HOME%;-Dcatalina.base=%CATALINA_BASE%;-Djava.io.tmpdir=%CATALINA_BASE%\temp;-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;-Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties;
```

```
-Dinrights.home="c:/inrights";-  
Djavax.net.ssl.trustStore=c:/inrights/keystore.jceks;  
-Djavax.net.ssl.trustStoreType=jceks;-Xms128m;-Xmx1024m;-XX:PermSize=64m;  
-XX:MaxPermSize=256m;-Dinrights.nodeId=admin-gui;-Dcom.sun.management.jmxremote;  
-Dcom.sun.management.jmxremote.authenticate=false;  
-Dcom.sun.management.jmxremote.ssl=false;-Dcom.sun.management.jmxremote.port=5012;  
-Duser.country=RU;-Duser.language=ru"
```

ВАЖНО! Строка со значением параметра – **JvmOptions** не должна содержать символов перевода каретки.

10) Установить службу **Apache Tomcat**, для чего в CLI выполнить команду:

```
cd <TOMCAT_HOME_ADMGUI>\bin  
service.bat install <service_name>
```

где **service_name** – имя устанавливаемой службы **Tomcat** для веб-приложения ПС, например, **TomcatAdminGUI**.

11) Запустить установленную службу **TomcatAdminGUI** с помощью мтс-оснастки **Services (Службы)**.

Примечание

При необходимости сервер **Tomcat** может быть запущен как приложение, в этом случае для запуска сервера в CLI следует выполнить команды:

```
cd TOMCAT_HOME_ADMGUI  
bin\startup.bat
```

12) Запустить **Apache Tomcat**, для чего выполнить команду
TOMCAT_HOME_ADMGUI/bin/startup.sh

После выполненных операций веб-интерфейс **admin-gui** будет доступен по адресу **http://<host>:8081/admin-gui**, где **<host>** – адрес сервера, на который был установлен ПС. Имя пользователя и пароль по умолчанию – **administrator/5ecr3t**.

2.2 Установка ПО, необходимого для взаимодействия ИС с ПС

ПС взаимодействует с системами-источниками данных и целевыми (управляемыми) системами с помощью java- и .net-коннекторов.

Коннекторы работают на сервере коннекторов, который может быть как внешним (устанавливается отдельно, используется для .net-коннекторов), так и встроенным (входит в состав ПС, используется для java-коннекторов). Например, для СУБД MySQL, Oracle, PostgreSQL используется сервер для java-коннекторов (как внешний, так и встроенный); для Active Directory или Exchange используется сервер для net-коннекторов.

Примечание

За подробной информацией о коннекторе, в том числе сведений о типе коннектора (java или .net), необходимо обратиться к документации на соответствующий коннектор.

Таблица 4 – Программное обеспечение для взаимодействия ПС и ИС

№	Название	Описание, назначение	Ссылка для загрузки
1.	СЕРВЕРЫ КОННЕКТОРОВ		
1.1	Connector Server	Сервер для коннекторов .NET	
1.2	connector-framework	Сервер для коннекторов JAVA (используется в случае необходимости установить отдельный сервер для java-коннекторов)	

2	.NET-КОННЕКТОРЫ		
2.1	ExchangeConnector	Коннектор к MS Active Directory и MS Exchange	
3	JAVA-КОННЕКТОРЫ		
3.1	Connector-csvfile	Коннектор к файлу формата CSV	
3.2	Lotus Notes	Коннектор к Lotus Notes	
3.3	Database Table	Коннектор к таблице БД	
3.4	LDAP	Коннектор к LDAP	
3.5	Scripted SQL	Коннектор к SQL-скрипту	
4	JDBC-ДРАЙВЕРЫ		
4.1	PostgreSQL	Драйвер для БД PostgreSQL	https://jdbc.postgresql.org/download.html
4.2	MySQL	Драйвер для БД MySQL	http://dev.mysql.com/downloads/connector/j/
4.3	Oracle	Драйвер для БД Oracle	http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html

2.2.1 Установка коннекторов на базе сервера коннекторов .NET

Для работы net-коннекторов (например, коннектора к Active Directory и MS Exchange) необходим специальный сервер, который должен быть установлен на компьютере, где работает сторонняя система (Active Directory, Exchange и др.). Таким образом, установка

net-коннекторов предполагает наличие установленного сервера коннекторов .NET (см. раздел 2.2.1.1), конфигурация которого импортирована в ПС (см. раздел 2.2.1.2).

2.2.1.1 Установка сервера коннекторов .NET

Перед установкой сервера коннекторов .NET необходимо установить ПО .NET Framework версии 4.0 или выше. Данное ПО должно быть установлено на компьютере, где должен работать сервер коннекторов.

Для установки и первичной настройки сервера коннекторов .NET используется два файла: **connid-dotnet-1.5.1.0.msi** (разработка предприятия-изготовителя) и архив **libs.zip**, который идет в составе сервера коннекторов .NET и содержит внешние библиотеки. Чтобы установить и настроить сервер коннекторов .NET необходимо выполнить следующие действия:

1) Установить сервер коннекторов .NET (**connid-dotnet-1.5.1.0.msi**), следуя указаниям мастера установки.

2) После того, как сервер коннекторов установлен, распаковать архив **libs.zip** в каталог, в который был установлен сервер коннекторов. В результате в данном каталоге должны быть расположены все файлы из архива **libs.zip**.

3) Остановить работающий процесс сервера коннекторов.

4) С помощью CLI, запущенного от имени пользователя с правами администратора, перейти в каталог, в который был установлен сервер коннекторов, и задать секретный ключ, выполнив команду

```
ConnectorServer /setkey <secret_key>
```

где **secret_key** - ключ, представляющий собой любую последовательность символов, например, **Secret**. Этот ключ необходимо будет указать при импорте конфигурации коннектора в теге <sharedSecret> (см. раздел 2.2.1.2).

- 5) Запустить сервер коннекторов одним из следующих способов:
- если сервер коннекторов установлен как служба **ConnId Connector Server** (в списке служб Windows имеется соответствующая служба), то запустить ее;
 - выполнить команду

```
ConnectorServer /run
```

Примечание

Выполнять команду следует, находясь в каталоге с установленным сервером коннекторов.

После установки сервера коннекторов его необходимо связать с ПС. Для этого следует импортировать конфигурацию сервера коннекторов (см. раздел 2.2.1.2).

2.2.1.2 Импорт конфигурации сервера коннекторов .NET

После установки сервера коннекторов его необходимо связать с ПС. Для этого необходимо импортировать конфигурацию сервера коннекторов, выполнив следующие действия:

- 1) В AGUI ПС выбрать пункт главного меню **Конфигурирование > Загрузить объект**.
- 2) Нажать кнопку **Выберите файл** и выбрать файл, содержащий параметры конфигурации сервера коннекторов;
- 3) Нажать кнопку **Import object** (см. Рис. 4).

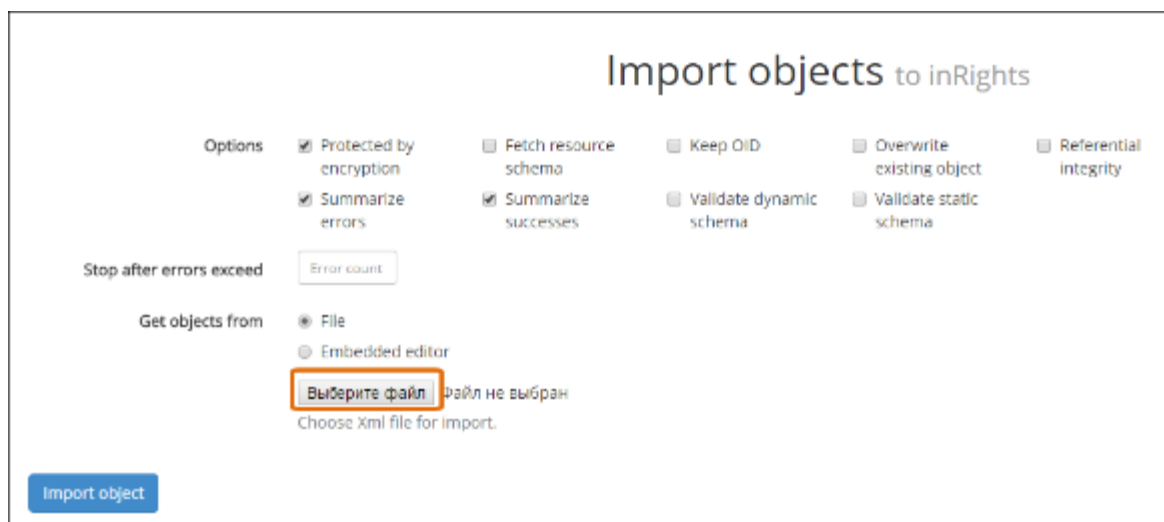


Рис. 4 – Импорт файла конфигурации сервера коннекторов в ПС

Пример содержимого файла конфигурации сервера коннекторов:

```
<object
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    version="22"
    xsi:type="c:ConnectorHostType">
    <name>localhost ICF connector (port 8759)</name>
    <description/>
    <metadata>
        <createTimestamp>2015-12-14T16:55:27.409+03:00</createTimestamp>
        <creatorRef oid="00000000-0000-0000-0000-000000000002" type="c:UserType"><!-- Administrator --></creatorRef>

    <createChannel>http://inrights.solarsecurity.ru/xml/ns/public/model/channels-3#objectImport</createChannel>
```



```
</metadata>  
<hostname>                </hostname>  
<port>8759</port>  
<sharedSecret>Secret</sharedSecret>  
</object>
```

В примере:

- **name** – имя сервера коннекторов;
- **hostname** – адрес сервера коннекторов;
- **Port** – порт сервера коннекторов;
- **sharedSecret** – секретный ключ, заданный на этапе

установки сервера коннекторов (см. раздел 2.2.1.1).

Чтобы убедиться в успешном импорте конфигурации сервера коннекторов, следует выбрать пункт главного меню **Информационные системы > Список информационных систем**. В блоке **Серверы коннекторов** должна появиться запись со сведениями о сервере коннекторов.

2.2.1.3 Настройка параметров журналирования операций на сервере коннекторов .NET

По умолчанию записи об операциях на сервере коннекторов .NET (логи) пишутся в один файл, удаление которого не предусмотрено. Соответственно, размер этого файла постоянно увеличивается, и в результате файл может занять все свободное место на сервере. Чтобы этого избежать, необходимо в файле конфигурации **ConnectorServer.exe.Config** настроить параметры журналирования, заменив содержимое раздела

```
<sharedListeners>  
  <add name="myListener"  
  
    type="System.Diagnostics.TextWriterTraceListener"  
    initializeData="connectorserver.log"
```

```
        traceOutputOptions="DateTime">
        <filter
type="System.Diagnostics.EventTypeFilter"
        initializeData="Verbose"/>
    </add>
</sharedListeners>
```

например, на следующее:

```
<sharedListeners>
  <add name="myListener"
    type="Microsoft.VisualBasic.Logging.FileLogTraceListener,
Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral,
    PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL"
    baseFileName="connectorserver.log"
    logFileCreationSchedule="Daily"
    traceOutputOptions="DateTime"
    Location="Custom"
    CustomLocation="C:\connectorlogs">
    <filter                                type="System.Diagnostics.EventTypeFilter"
initializeData="Verbose"/>
  </add>
</sharedListeners>
```

В этом случае каждый день будет создаваться новый файл журнала. Далее можно настроить безопасное удаление старых файлов.

2.2.1.4 Установка net-коннектора

Для установки net-коннектора необходимо:

1) Поместить исполняемые и вспомогательные файлы требуемого net-коннектора в каталог с установленным сервером коннекторов.

Примечание

Исполняемые файлы поставляются в виде архивов, в некоторых случаях они могут быть дополнены архивом с внешними библиотеками, необходимыми для работы коннектора.

В этом случае необходимо поместить файлы из данного архива в каталог с установленным сервером коннекторов.

Например, коннектор **ExchangeConnector**, который идет в составе ПС, содержит два файла: **Exchange.Connector-1.4.7.4.zip** и **libs.zip**.

Для его установки необходимо сначала распаковать архив **Exchange.Connector-1.4.7.4.zip** в каталог с установленным сервером коннекторов, а затем в тот же каталог распаковать архив **libs.zip**.

2) Перезапустить сервер коннекторов .NET.

3) В AGUI в разделе **Информационные системы > Список информационных систем** в блоке Серверы коннекторов выбрать сервер коннекторов и действие Исследовать (см. Рис. 5).



Рис. 5 – Обнаружение коннектора

2.2.2 Установка коннекторов на базе сервера коннекторов JAVA / внутреннего сервера коннекторов

Для работы java-коннекторов (например, коннектора к БД Oracle) необходим специальный сервер, который может быть внутренним (уже установлен в системе) или внешним (см. раздел 2.2.2.1).

Для установки и первичной настройки сервера коннекторов JAVA используется два файла: **connid-connector-server-java-1.4.4.0-SNAPSHOT.zip** (разработка предприятия-изготовителя) и архив **connid-connector-server-java-1.4.4.0-SNAPSHOT-libs.zip**, который идет в составе сервера коннекторов JAVA и содержит внешние библиотеки.

ВАЖНО! Для установки сервера коннекторов JAVA необходимо наличие ПО Java версии 1.8 или выше. Кроме того, к установке конкретного коннектора могут предъявляться дополнительные требования. Подробнее см. в соответствующей документации.

2.2.2.1 Установка сервера коннекторов JAVA

Для установки сервера коннекторов JAVA на ОС Linux следует:

- 1) На компьютере, где планируется запускать сервер коннекторов JAVA, создать пустой каталог (например, **jconnserv**).
- 2) Распаковать архив `connid-connector-server-java-1.4.4.0-SNAPSHOT.zip` в каталог **jconnserv**.
- 3) Распаковать архив `connid-connector-server-java-1.4.4.0-SNAPSHOT-libs.zip` в каталог **jconnserv**.
- 4) В каталоге **jconnserv** создать подкаталог **bundles**.
- 5) Используя `jar / unzip` или подобные средства, извлечь файл `pathorg/identityconnectors/framework/server/connectorserver.properties` из `jar`-архива `framework-internal-XX.jar`.
- 6) В CLI выполнить команду.

```
java -cp "connector-framework.jar:connector-framework-internal.jar:groovy-all.jar" \
org.identityconnectors.framework.server.Main
```

В результате выполнения команды на экране должны отобразиться строки:

```
Usage:
    Main -run -properties
    Main -setKey -key -properties
    Main -setDefault -properties
```

- 7) Сконфигурировать сервер коннекторов, выполнив в CLI команду

```
java -cp "connector-framework.jar:connector-  
framework-internal.jar:groovy-all.jar" \  
org.identityconnectors.framework.server.Main -  
setKey -key -properties connectorserver.properties
```

8) Выполнить запуск сервера коннекторов с помощью команды

```
java -cp "connector-framework.jar:connector-  
framework-internal.jar:groovy-all.jar" \  
org.identityconnectors.framework.server.Main -run \  
-properties connectorserver.properties
```

Для установки сервера коннекторов JAVA на Windows следует:

1) На компьютере, где планируется запускать сервер коннекторов JAVA, создать пустой каталог (например, **jconnserv**).

2) Распаковать архив **connid-connector-server-java-1.4.4.0-SNAPSHOT.zip** в каталог **jconnserv**.

3) Распаковать архив **connid-connector-server-java-1.4.4.0-SNAPSHOT-libs.zip** в каталог **jconnserv**.

4) В каталоге **jconnserv** создать подкаталог **bundles** (в каталоге **bundles** будут храниться java-коннекторы для работы с ПС).

5) В файле **jconnserv\conf\ConnectorServer.properties** установить необходимое значение для порта, на котором должен работать сервер коннекторов:

```
connectorserver.port=8760
```

6) В файле **jconnserv\conf\ConnectorServer.properties** заменить строку

```
connectorserver.loggerClass=org.identityconnectors.  
common.logging.slf4j.SLF4JLog
```

на

```
connectorserver.loggerClass=org.identityconnectors.  
common.logging.impl.JDKLogger
```

7) Установить секретный ключ, выполнив в CLI команду

```
bin\JavaConnectorServer.bat /setkey <секретный  
ключ>
```

Примечание

Выполнять команду следует, находясь внутри распакованного архива на уровне каталогов bin, bundles, conf, lib.

8) Выполнить запуск сервера коннекторов с помощью команды

```
bin\JavaConnectorServer.bat /run
```

Примечание

Для остановки работы сервера коннекторов необходимо нажать комбинацию клавиш **Ctrl+C**.

9) Для того чтобы установить службу **JavaConnectorServer**, в CLI выполнить команду:

```
JavaConnectorServer.bat /install <service_name>
```

где **service_name** – имя устанавливаемой службы, например, **JavaConnectorServer**. Если имя не указать, то службе будет присвоено имя по умолчанию: **JavaConnectorServer**.

10) Выполнить импорт конфигурации сервера коннекторов по аналогии с .NET-сервером коннекторов (см. раздел 2.2.1.2). При этом следует изменить значение порта в теге <port> на **8760**.

2.2.2.2 Настройка параметров журналирования операций на сервере коннекторов JAVA

По умолчанию записи об операциях на сервере коннекторов JAVA (логи) пишутся в один файл, удаление которого не предусмотрено. Соответственно, размер этого файла постоянно увеличивается, и в результате файл может занять все свободное место на сервере. Чтобы этого избежать, необходимо:

1) В каталоге, где установлен **Java Connector Server**, создать подкаталог **logs** (в logs будут сохраняться журнальные файлы);

2) В конфигурационном файле <каталог с установленным Java Connector Server>/conf/ConnectorServer.properties заменить строку

```
connectorserver.loggerClass=org.identityconnectors.  
common.logging.slf4j.SLF4JLog
```

на строку

```
connectorserver.loggerClass=org.identityconnectors.  
common.logging.impl.JDKLogger
```

3) В конфигурационном файле **<каталог с установленным Java Connector Server>/conf/logging.properties** заменить его содержимое (все строки, не начинающиеся с символа #), примерно на следующее:

```
handlers=java.util.logging.FileHandler  
java.util.logging.ConsoleHandler.formatter =  
java.util.logging.SimpleFormatter  
java.util.logging.FileHandler.pattern =  
logs/connectorserver%g.log  
java.util.logging.FileHandler.limit = 100000  
java.util.logging.FileHandler.count = 5  
java.util.logging.FileHandler.formatter =  
java.util.logging.SimpleFormatter  
java.util.logging.FileHandler.append = true  
.level=INFO
```

где:

- **handles** – список имен классов обработчиков, которые добавляются в корневой регистратор;
- **java.util.logging.ConsoleHandler.formatter** – имя класса, который отвечает за формат журнальных записей в консоли;
- **java.util.logging.FileHandler.pattern** – паттерн создания журнального файла;
- **java.util.logging.FileHandler.limit** – максимальный размер журнального файла в байтах. Как только журнальный файл достигнет максимального размера, будет создан новый файл;
- **java.util.logging.FileHandler.count** – количество повторений процесса создания новых файлов. Например, если значение параметра

равно 5, то на 6-ой раз самый первый файл удалится, а на его месте создастся новый. Таким образом обеспечивается ротация;

- **java.util.logging.FileHandler.formatter** – имя класса, который отвечает за формат журнальных записей в файлах;
- **java.util.logging.FileHandler.append** – параметр указывает, должен ли FileHandler присоединяться к уже существующим файлам (по умолчанию - false);
- **.level** – уровень журналирования (SEVERE, WARNING, CONFIG, INFO, ALL, OFF и т.д.).

2.2.2.3 Установка java-коннектора

Для установки java-коннектора следует:

- 1) Поместить исполняемые и вспомогательные файлы требуемого java-коннектора в каталог с установленным сервером коннекторов.

Примечание

Коннекторы к БД входят в состав ПС и предустановлены на внутреннем сервере коннекторов.

Примечание

Исполняемые файлы поставляются в виде архивов, в некоторых случаях они могут быть дополнены архивом с внешними библиотеками, необходимыми для работы коннектора.

В этом случае необходимо поместить файлы из данного архива в каталог с установленным сервером коннекторов.

- 2) При необходимости (для коннекторов к БД) скопировать соответствующий jdbc-драйвер в каталог с установленным сервером коннекторов (в случае использования внутреннего сервера коннекторов - каталоги **TOMCAT_HOME\lib** и **TOMCAT_HOME_ADMGU\lib**).

- 3) Перезапустить сервер коннекторов.

В AGUI в разделе **Информационные системы > Список информационных систем** в блоке Серверы коннекторов выбрать сервер коннекторов и действие Исследовать (см. Рис. 5).

3 ОПИСАНИЕ АДМИНИСТРАТИВНОГО ВЕБ-ИНТЕРФЕЙСА ПС

3.1 Вход в веб-интерфейс администратора

Для входа в веб-интерфейс **admin-gui** (далее AGUI) ПС следует в адресной строке браузера ввести **http://<host>:8081/admin-gui**, где host – адрес сервера, на который был установлен ПС. На экране отобразится окно авторизации (см. Рис. 6).

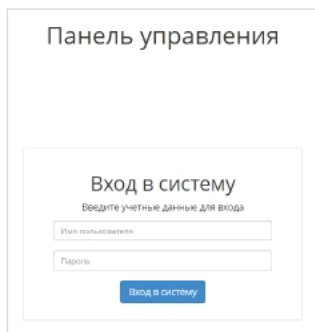


Рис. 6 – Окно авторизации

При первом входе в систему в окне авторизации следует указать логин и пароль по умолчанию: После этого рекомендуется изменить пароль (см. раздел 3.9), а затем заново авторизоваться уже с новым паролем.

3.2 Описание основных элементов интерфейса

После успешного входа в систему отображается раздел **Панель управления** (см. Рис. 7).

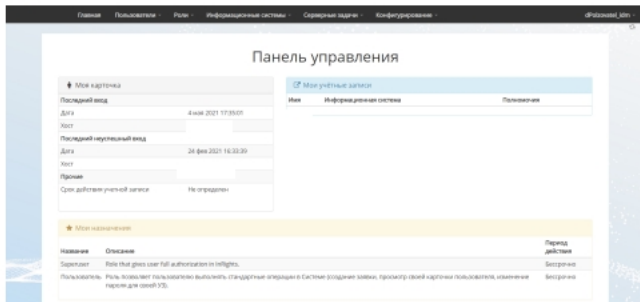


Рис. 7 – Раздел Панель управления

В верхней части интерфейса расположено главное меню, которое предоставляет доступ к следующим разделам AGUI (см. Рис. 7):

- **Главная (Панель управления)** – в разделе отображаются сведения о текущем пользователе и показатели работы ПС (см. раздел 3.3);
- **Пользователи** – предназначен для управления учетными записями пользователей. Кроме того, в разделе можно создавать и изменять элементы организационной структуры предприятия (см. раздел 3.4);
- **Роли** – предназначен для управления элементами ролевой модели, применяемой на предприятии (см. раздел 3.5);
- **Информационные системы** – предназначен для управления информационными системами (ИС), которые взаимодействуют с ПС. В этом разделе администратор может подключать внешние ИС и БД (например, Active Directory, БД MySQL, MS Exchange, LDAP, Lotus Notes и т.п.);

- **Серверные задачи** – предназначен для просмотра специальных задач, выполняемых на сервере ПС, и управления ими (см. раздел 3.7);

- **Конфигурирование** – предназначен для настройки ПС и управления его функционированием (см. раздел 3.8).

Кроме того, в правой верхней части AGUI ПС расположено меню администратора (**administrator**), предоставляющее доступ к функциям:

- смены пароля администратора, а также паролей его учетных записей (**administrator > Reset passwords**, см. раздел 3.9);
- завершения сеанса работы текущего пользователя (**administrator > Log out**).

Для выбора одной из операций, которые можно выполнять над объектами системы (например, для удаления или экспорта объектов), в AGUI имеется элемент **Меню действий** (см. Рис. 8).

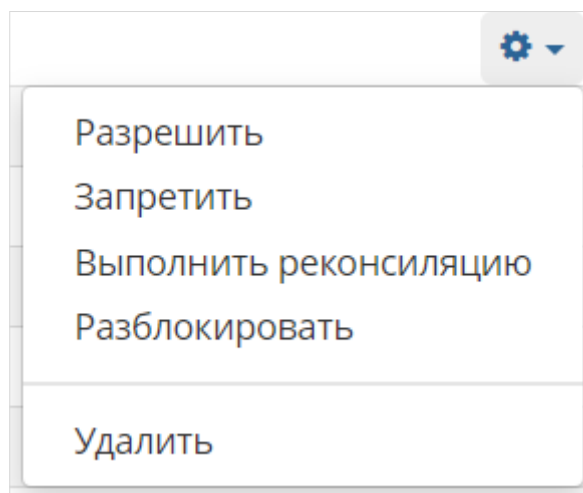


Рис. 8 – Примеры меню действий

3.3 Панель управления ПС: получение сведений о текущем пользователе и состоянии системы

В разделе **Панель управления ПС** отображаются общие сведения о текущем пользователе и состоянии ПС:

- в блоке **Моя карточка** – данные о последнем входе (как успешном, так и неуспешном) текущего пользователя в систему и сроке действия его учетной записи;
- в блоке **Мои назначения** – назначения, которые были сделаны для текущего пользователя (сведения о том, какие роли назначены пользователю);
- в блоке **Мои учетные записи** – учетные записи текущего пользователя в целевых информационных системах (ресурсах).

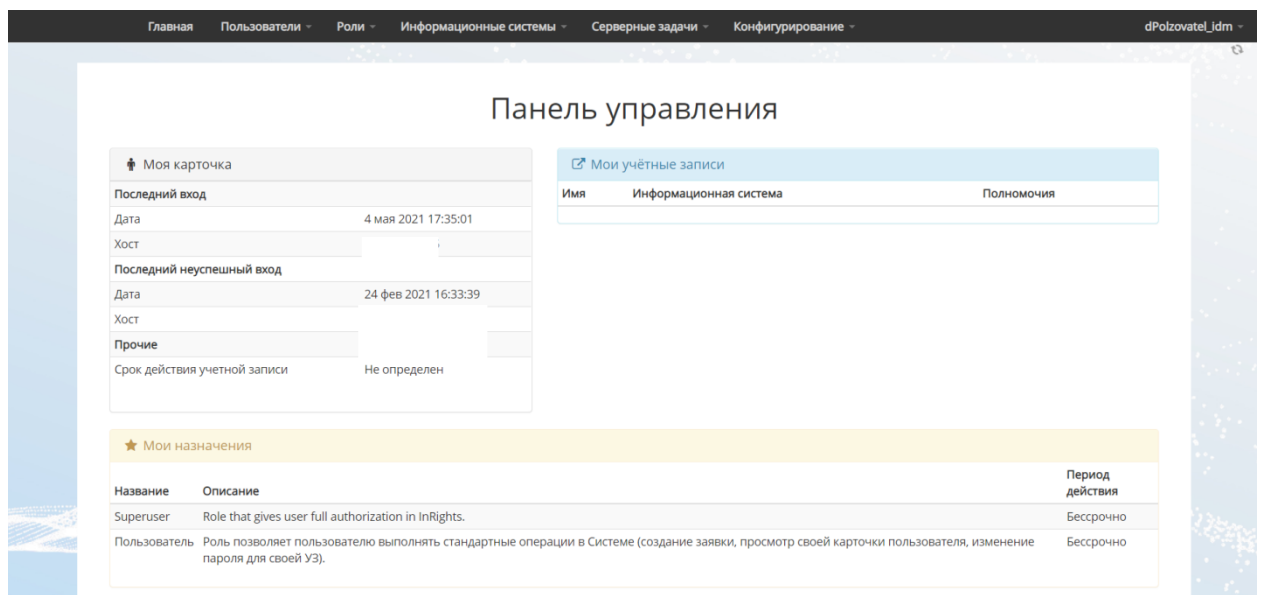


Рис. 9 – Раздел Панель управления ПС

3.4 Пользователи: управление пользователями, задание организационной структуры предприятия

3.4.1 Управление учетными записями пользователей

Вся информация о работнике компании (ФИО, подразделение, должность, рабочий телефон и др., а также сведения об аккаунтах и правах доступа работника в различных целевых системах компании) содержится в профиле пользователя (объект системы User). Существует возможность настроить разные правила обработки для разных работников (например, штатные работники и внештатные).

3.4.1.1 Управление списком пользователей: просмотр, редактирование, удаление

Управление учетными записями пользователей выполняется в разделе **Пользователи** (см. Рис. 10). Для перехода в этот раздел необходимо в главном меню выбрать пункт **Пользователи > Список пользователей**.

Наименование	Имя	Фамилия	ФИО	Email	Учетные записи
AIvanova	Анна Мария	Иванова	Иванова Анна Мария Ивановна	A.ivanova@mirat.ru	1
AILenin	Антон	Ленин	Ленин Антон Ираклиевич	A.lenin@greenatom.ru	1
AIPelmeshkin	Анатолий	Пельмешкин	Пельмешкин Анатолий Иванович	A.pelmeshkin@ra.ru	1
AIPichugina	Анастасия	Пичугина	Пичугина Анастасия Ивановна	A.pichugina@mirat.ru	1
AIPodyomov	Андрей	Подъёмов	Подъёмов Андрей Игоревич	A.podyomov@ra.ru	1
AIVardyshev	Андрей	Вардышев	Вардышев Андрей Игоревич	A.vardyshev@ra.ru	2
AKategoriya	А	Категория	Категория А Команда	A.kategoriya@ra.ru	2
AKotov	Александр	Котов	Котов Александр	A.kotov@greenatom.ru	2
AKSiluanov	Альберт	Силуанов	Силуанов Альберт Кузьмич	A.siluanov@mirat.ru	1
alximikp	Авдотья	Сидорова	Сидорова Авдотья Павловна	A.sidorova@ra.ru	1

Рис. 10 – Пользователи ПС

3.4.1.1.1 Управление списком пользователей: просмотр, поиск, блокировка и удаление пользователей

В разделе **Пользователи** (см. Рис. 10) отображается таблица с данными существующих в системе учетных записей пользователей ПС. Администратор может выполнять поиск, блокировку, удаление и другие операции с данными пользователей.

3.4.1.1.1.1 Поиск пользователей

Для поиска конкретного пользователя следует:

1) В соответствующем списке выбрать один или несколько критериев поиска:

- **Идентификатор** – наименование пользователя (Name);
- **Имя** – имя пользователя (Given Name);

- **Фамилия** – фамилия пользователя (Family Name);
 - **ФИО** - фамилия, имя, отчество (Full Name).
- 2) В поле **Найти** ввести требуемое значение;
- 3) Нажать кнопку **Найти**.

Пример поиска данных и отображения его результатов приведен на рисунке ниже (см. Рис. 11).

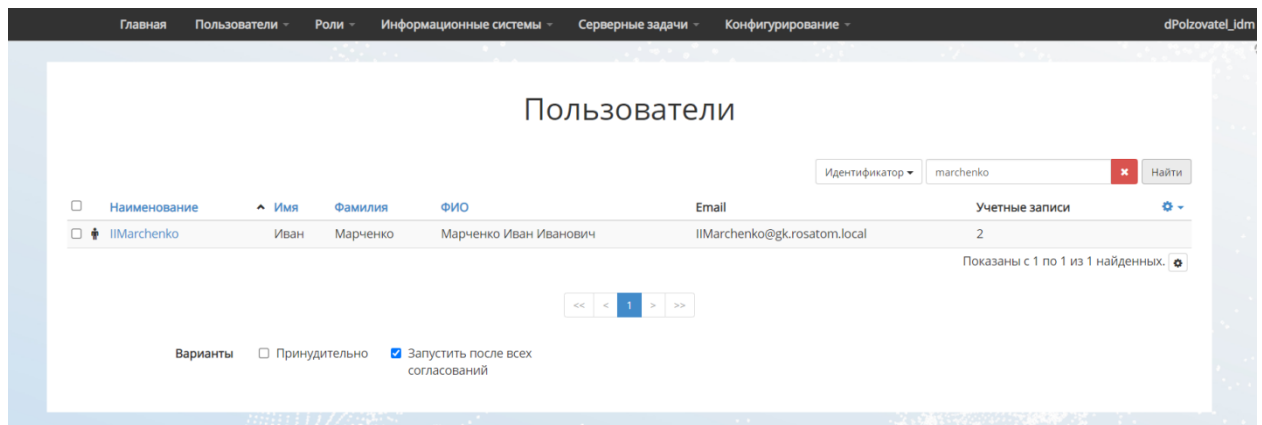


Рис. 11 – Результат поиска пользователя

3.4.1.1.1.2 Групповые операции с данными пользователей: изменение статуса УЗ, проверка данных, разблокировка пользователей

Администратор может выполнить с данными одного или нескольких пользователей следующие операции (см. Рис. 12):

- изменить административный статус УЗ пользователей на **Разрешено (Разрешить)**;
- изменить административный статус УЗ пользователей на **Запрещено (Запретить)**;
- проверить актуальность и целостность данных (**Выполнить реконсильяцию**);
- разблокировать пользователей, которые были **заблокированы (Разблокировать)**.

Для этого следует:

- 1) Выбрать требуемые учетные записи, установив соответствующие флажки.

2) В меню действий выбрать соответствующий пункт (см. Рис. 12). В результате на экране отобразится блок с информацией об успешном или неуспешном выполнении операции.

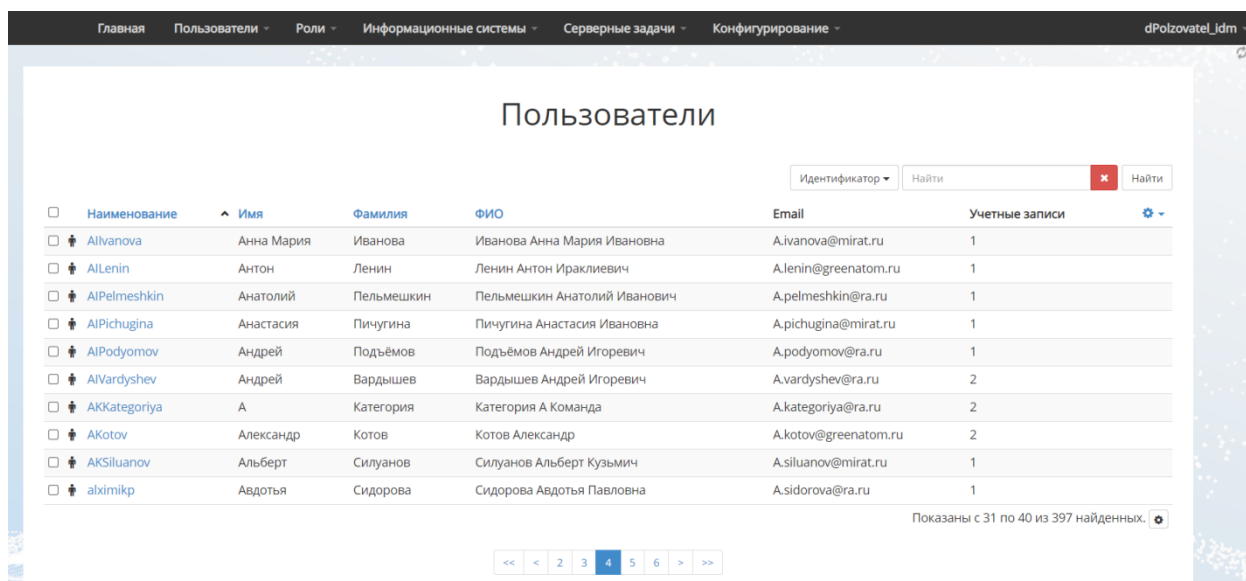


Рис. 12 – Групповые операции с данными пользователей

3.4.1.1.1.3 Удаление учетных записей пользователей

Для удаления учетных записей пользователей следует:

- 1) Выбрать требуемые учетные записи, установив соответствующие флажки.
- 2) В меню действий выбрать пункт **Удалить** (см. Рис. 13).
- 3) В отобразившемся диалоговом окне подтвердить удаление.

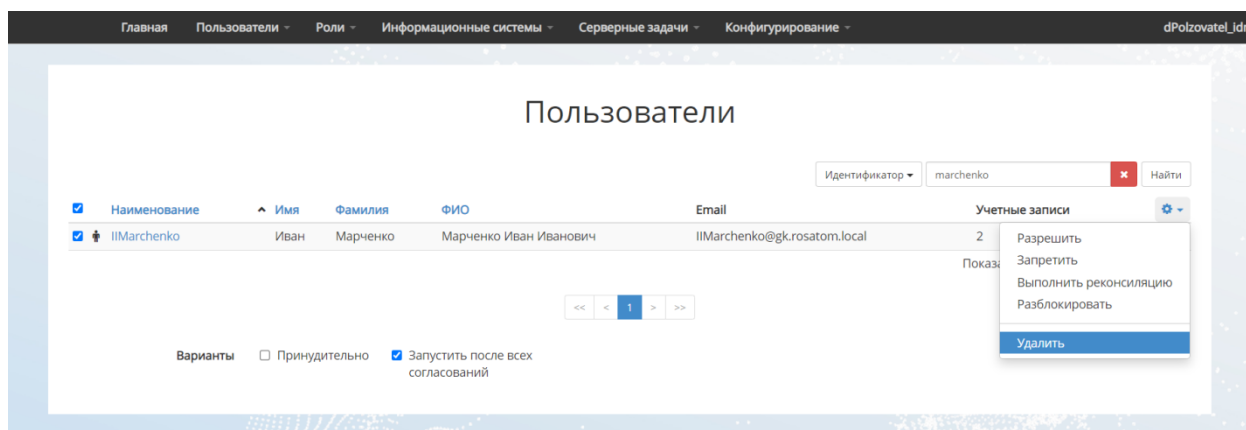


Рис. 13 – Удаление пользователей

3.4.1.1.2 Карточка пользователя: просмотр и редактирование параметров УЗ пользователя

Для просмотра и/или редактирования параметров учетной записи пользователя следует:

1) В списке пользователей выбрать требуемую УЗ, перейдя по соответствующей ссылке.

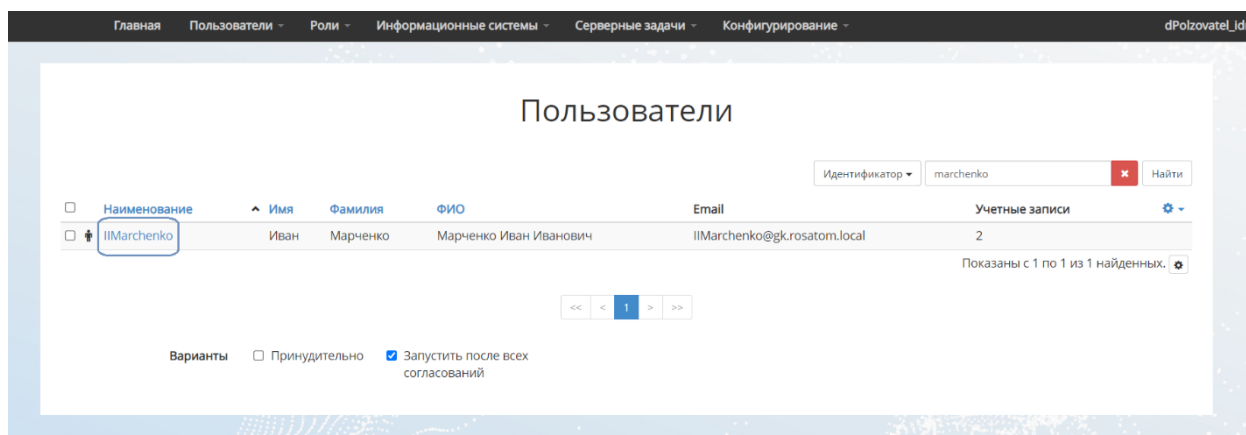


Рис. 14 – Редактирование учетной записи пользователя: шаг 1

2) При необходимости в блоке **Карточка пользователя** отредактировать требуемые данные о пользователе.

3) При необходимости в блоке **Учетные записи** изменить данные имеющихся учетных записей пользователя, а также добавить или удалить УЗ.

4) При необходимости в блоке **Назначения** изменить данные имеющихся назначений, а также добавить или удалить назначения (см. раздел 3.4.1.1.2.2).

Примечание

В блоке Назначения администратор может управлять только прямыми назначениями.

5) При необходимости установить/снять флажки:

- **Принудительно** – запустить реконсильацию зависимых объектов;

- **Выполнить реконсиляцию** – сравнить реальные атрибуты учетных записей с атрибутами пользователя в ПС;
 - **Запустить после всех согласований** – применить изменения после всех согласований;
 - **Показать результаты** – оставаться на странице после сохранения (после нажатия кнопки **Сохранить**).
- 6) Для сохранения сделанных изменений нажать кнопку **Сохранить** (см. Рис. 15).

Рис. 15 – Редактирование учетной записи пользователя: шаги 2-6

3.4.1.1.2.1 Карточка пользователя, блок Учетные записи: управление УЗ пользователя

В блоке **Учетные записи** администратор может управлять учетными записями пользователя во внешних системах. Поскольку по отношению к ПС внешние ИС и БД являются компонентами среды функционирования, описание процесса управления внешними учетными записями не входит в рамки этого документа.

3.4.1.1.2.2 Карточка пользователя, блок Назначения: изменение назначений

В блоке **Назначения** администратор может управлять полномочиями, которые были предоставлены пользователю.

3.4.1.1.2.2.1 Редактирование назначения

Для редактирования имеющегося назначения следует:

- 1) Отобразить параметры требуемого назначения, нажав на соответствующую ссылку (см. Рис. 16).
- 2) Отредактировать необходимые значения.
- 3) Установить флажок **Показать результаты** и нажать кнопку **Сохранить**.

The screenshot shows a web interface for editing user assignments. On the left, there is a sidebar with a tree view containing 'Назначения' and 'Пользователь'. The main area displays a form for editing an assignment. The form has the following fields:

- Description: A text input field.
- Tenant: A dropdown menu with 'None' selected and edit/delete icons.
- Organization Unit/Project: A dropdown menu with 'None' selected and edit/delete icons.
- Activation: A dropdown menu with 'Undefined' selected.
- Административный статус: A dropdown menu.
- Начало действия: A date picker showing '17'.
- Окончание действия: A date picker showing '17'.
- Target: A dropdown menu with 'Пользователь' selected.

Рис. 16 – Редактирование параметров назначения

3.4.1.1.2.2.2 Задание связи учетной записи, роли или подразделения с назначением

Администратор может выполнить следующие операции:

- назначить учетную запись;
- назначить роль;
- назначить подразделения.

Для выполнения конкретной операции необходимо:

- 1) Выбрать требуемые назначения, установив соответствующие флажки.

- 2) В меню действий выбрать требуемый пункт (см. Рис. 17).
- 3) В отобразившемся окне выбрать требуемые объекты (учетные записи / роли / подразделения) и нажать кнопку **Assign** (см. Рис. 17).
- 4) Установить флажок **Показать результаты** и нажать кнопку **Сохранить**.

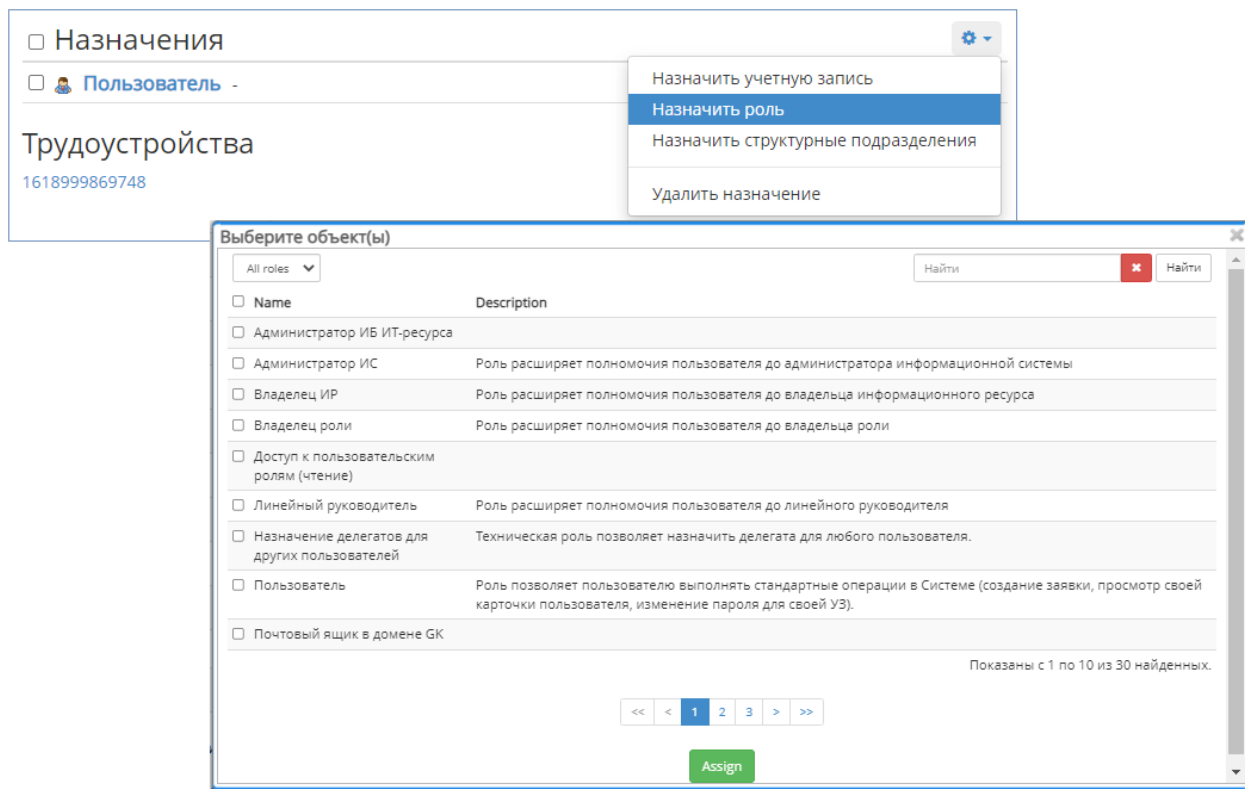


Рис. 17 – Назначение роли

3.4.1.1.2.2.3 Отмена назначения

Для отмены сделанного назначения следует:

- 1) Выбрать требуемые назначения, установив соответствующие флажки;
- 2) В меню действий выбрать пункт **Удалить назначение** (см. Рис. 18);
- 3) В отобразившемся диалоговом окне подтвердить отмену (удаление) назначения.
- 4) Установить флажок **Показать результаты** и нажать кнопку **Сохранить**.

Карточка пользователя

Name: IMarchenko

Full Name: Марченко Иван Иванович

Given Name: Иван

Family Name: Марченко

Additional Name: Иванович

Work start date: 21.04.21

Email Address: IMarchenko@gk.rosatom.local

Cost Center: getfromhr

HR Status: Активен

dateOfBirth: 21.10.67

firstName_en: Ivan I. Marchenko

Activation: Administrative Status: Разрешено

Варианты: ☐ Принудительно ☐ Выполнить реконсолиацию ☒ Запустить после всех согласований ☒ Показать результаты

Назад Показать все назначения Сохранить

Учетные записи

AD - gk.rosatom.local default, IMarchenko

AD - inter.rosatom.local default, IMarchenko

Назначения

Пользователь -

Почтовый ящик в домене GK

Подтвердите удаление

Вы действительно хотите удалить 1 назначений?

Yes No

Target: Почтовый ящик в домене GK

Трудоустройства: 161899969748

Рис. 18 – Удаление назначения

3.4.1.1.2.2.4 Просмотр всех назначений

Для просмотра всех назначений (как прямых, так и непрямых) следует нажать кнопку **Показать все назначения**, после чего на экране отобразится окно **View direct & indirect assignments**, содержащее сведения обо всех назначениях, относящихся к пользователю (см. Рис. 19).

Карточка пользователя

Name: IMarchenko

Full Name: Марченко Иван Иванович

Given Name: Иван

Family Name: Марченко

Additional Name: Иванович

Work start date: 21.04.21

Email Address: IMarchenko@gk.rosatom.local

Cost Center: getfromhr

HR Status: Активен

dateOfBirth: 21.10.67

firstName_en: Ivan I. Marchenko

Activation: Administrative Status: Разрешено

Варианты: ☐ Принудительно ☐ Выполнить реконсолиацию ☒ Запустить после всех согласований ☐ Показать результаты

Назад Показать все назначения Сохранить

Учетные записи

AD - gk.rosatom.local default, IMarchenko

AD - inter.rosatom.local default, IMarchenko

View direct & indirect assignments

Name	Type	Description	Tenant	Org	Kind	Intent
Назначение делегатов для других пользователей	Indirect	Техническая роль позволяет назначить делегата для любого пользователя.				
Пользователь	Direct	Роль позволяет пользователю выполнять стандартные операции в Системе (создание заявок, просмотр своей карточки пользователя, изменение пароля для своей УЗ).				
Техническая роль: просмотр аккаунтов своих подчиненных	Indirect					
Учетная запись в домене gk.rosatom.local	Direct					
Учетная запись в домене inter.rosatom.local	Direct					
AD - gk.rosatom.local	Indirect				ACCOUNT	default
AD - inter.rosatom.local	Indirect				ACCOUNT	default
Помощник секретаря	Direct	Сотрудник секретариата				

Показаны с 1 по 10 из 10 найденных.

Рис. 19 – Просмотр всех назначений

3.4.2 Задание организационной структуры предприятия

3.4.2.1 Просмотр и изменение элементов организационной структуры предприятия

Управление единицами организационной структуры предприятия выполняется в разделе **Организационная структура** (см. Рис. 20). Для перехода в этот раздел необходимо в главном меню выбрать пункт **Пользователи > Иерархия подразделений**.

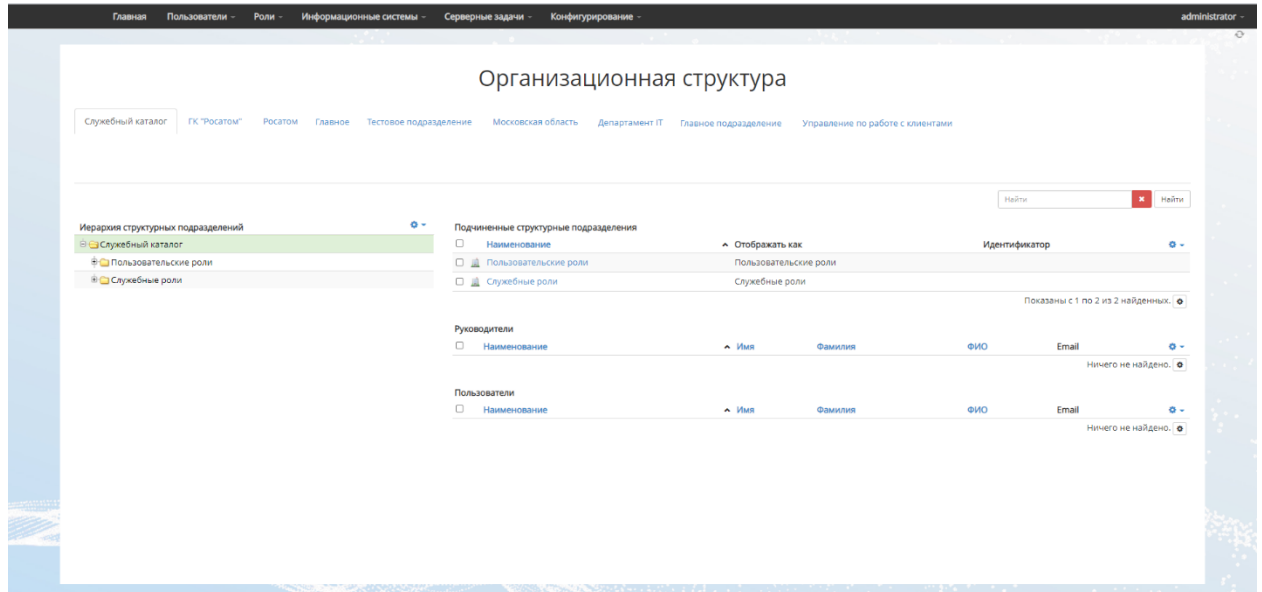


Рис. 20 – Организационная структура

В разделе **Организационная структура** имеются вкладки, которые определяются как основные (верхние) отделы организации (например, **Главное**, офис в **Московской области** – Московская область и т.п.). На каждой такой вкладке, в свою очередь, собрана информация по дочерним подразделениям конкретного отдела. В блоке **Иерархия подразделений** в виде дерева отображается список всех подразделений, относящихся к конкретному отделу.

Для просмотра / редактирования данных какого-либо подразделения следует:

- 1) При необходимости перейти на соответствующую вкладку.
- 2) В блоке **Иерархия подразделений** раскрыть содержимое требуемого элемента дерева и выбрать необходимое подразделение, нажав на соответствующий элемент дерева.

3) В блоке **Подчиненные подразделения** выбрать требуемое подразделение, перейдя по соответствующей ссылке.

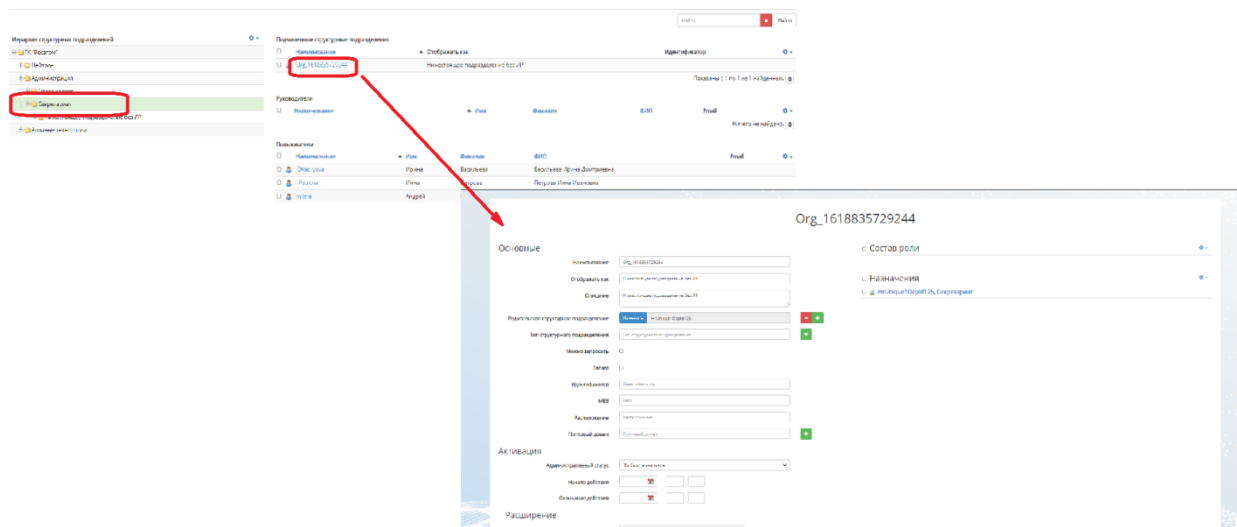


Рис. 21 – Редактирование подразделения: шаги 1-3

4) При необходимости отредактировать конкретные данные о подразделении;

5) При необходимости в блоках **Состав роли** и **Назначения** изменить имеющиеся у подразделения права доступа.

6) При необходимости установить/снять флажки:

- **Принудительно** – запустить реконсиляцию зависимых объектов;
- **Выполнить реконсиляцию** – сравнить реальные атрибуты подразделения с атрибутами в ПС;
- **Запустить после всех согласований** – применить изменения после всех согласований;
- **Показать результаты** – оставаться на странице после сохранения (после нажатия кнопки **Save**).

7) Для сохранения сделанных изменений нажать кнопку **Save** (см. Рис. 22).

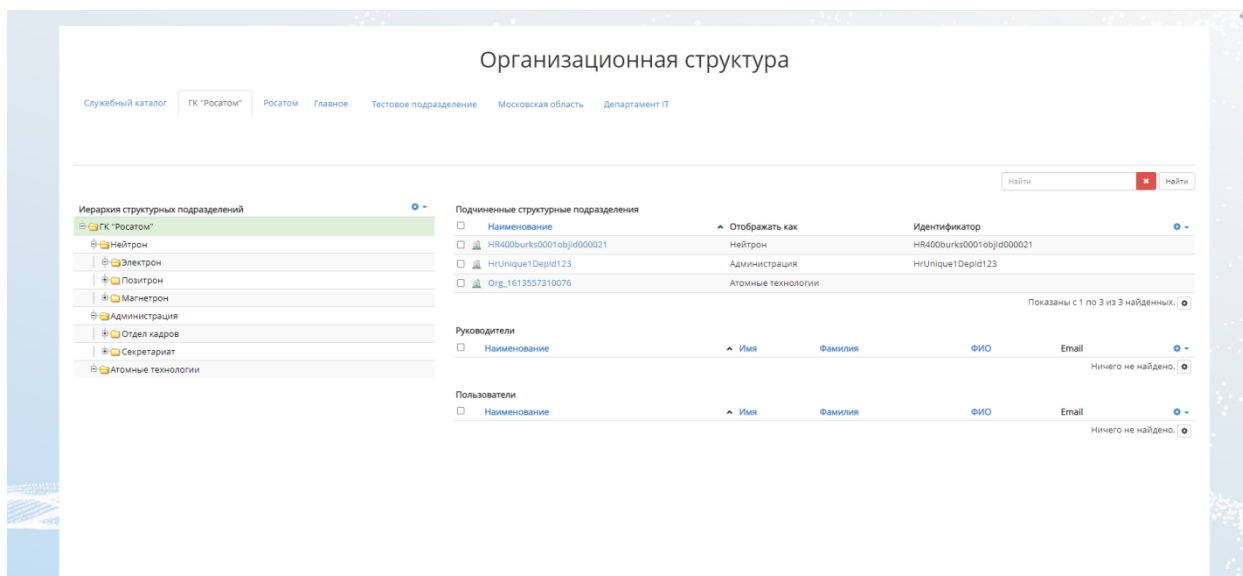


Рис. 22 – Редактирование подразделения: шаги 4-5

3.5 Роли: задание ролевой модели и управление ролями пользователей

Управление доступом на основе ролей – это политика избирательного управления доступом, при которой права доступа субъектов системы на объекты группируются с учетом специфики их применения, образуя роли. Роль представляет собой совокупность прав доступа на объекты системы. То есть роль – это набор прав доступа, который назначается пользователю, в результате чего пользователь получает полномочия на выполнение конкретных действий, заданных в параметрах роли. Ролевая модель позволяет реализовать гибкие правила разграничения доступа.

ПС позволяет администратору настраивать ролевую модель, создавая роли и управляя списком ролей, а также содержит набор предустановленных по умолчанию ролей, обеспечивающих разделение полномочий до момента настройки ролевой модели ПС:

Таблица 5 – Предустановленные роли, обеспечивающие разделение полномочий

Роль	Полномочия
Пользователь	Базовая роль, позволяющая выполнять стандартные операции в системе
Администратор СЦУДП	Просмотр всех страниц разделов и выполнение любых действий

3.5.1 Создание ролей

Для создания роли следует:

- 1) В главном меню выбрать пункт **Роли > Новая роль**.
- 2) В разделе **Создать новую роль** (см. Рис. 23) задать значения параметров:
 - **Название** (название роли, например, Работник отдела бухучета);
 - **Описание** (описание роли).

Примечание

Описание всех параметров, которые можно задать при создании роли, приведено в таблице ниже (см. Таблица 5 –).

- 3) При необходимости установить/снять флажки:
 - Принудительно – запустить реконсиляцию зависимых объектов;
 - Выполнить реконсиляцию – сравнить реальные атрибуты учетных записей с атрибутами пользователя в ПС;
 - Запустить после всех согласований – применить изменения после всех согласований;
 - Показать результаты – оставаться на странице после сохранения (после нажатия кнопки **Save**).
- 4) Нажать кнопку **Save**.

Рис. 23 – Управление списком ролей: просмотр, редактирование, удаление ролей

В таблице ниже (см. Таблица 6 –) приведено описание параметров, которые можно задать при создании роли.

Таблица 6 – Параметры (свойства), задаваемые при создании роли

Параметр	Описание
ОСНОВНЫЕ	
Название	Наименование роли
Отображать как	Отображаемое наименование роли
Описание	Подробное описание роли
Тип	Тип роли, например: Системная, Бизнес, Техподдержка
Можно запрашивать	При установке флажка роль будет отображаться в интерфейсе пользователя в каталоге ролей, которые можно назначать пользователям. Для системных ролей устанавливать флажок не рекомендуется
Идентификатор	Идентификатор роли
Уровень риска	Индикатор уровня риска,

Владелец	Бизнес-владелец роли. Атрибут позволяет назначить владельца роли. Владелец роли может быть как пользователь (User), так и подразделение (Org. Unit)
Согласующий	Участник маршрута согласования. Участником может быть как пользователь (User), так и подразделение (Org. Unit). Можно указать несколько участников
Максимально назначено	Максимальное количество назначений роли
Минимально назначено	Минимальное количество назначений роли
АКТИВАЦИЯ	
Административный статус	Административный статус роли: • Разрешено – роль активна (можно назначать пользователям); • Запрещено – роль неактивна; • Архивировано – роль неактивна, находится в архиве ролей.
Начало действия	Начало действия роли (дата и время)
Окончание действия	Окончание действия роли (дата и время)

Кроме того, при создании роли можно задать:

- **Состав роли** – это совокупность прав доступа, которая будет выдана пользователям или организационным единицам в результате присвоения им этой роли;
- **Назначения** – дополнительные аспекты поведения роли. Например, создание групп Active Directory.

3.5.2 Управление списком ролей: просмотр, редактирование, удаление ролей

Управление списком существующих в системе ролей выполняется в разделе **Список ролей** (см. Рис. 24). Для перехода в этот раздел необходимо в главном меню выбрать пункт **Роли > Список ролей**.

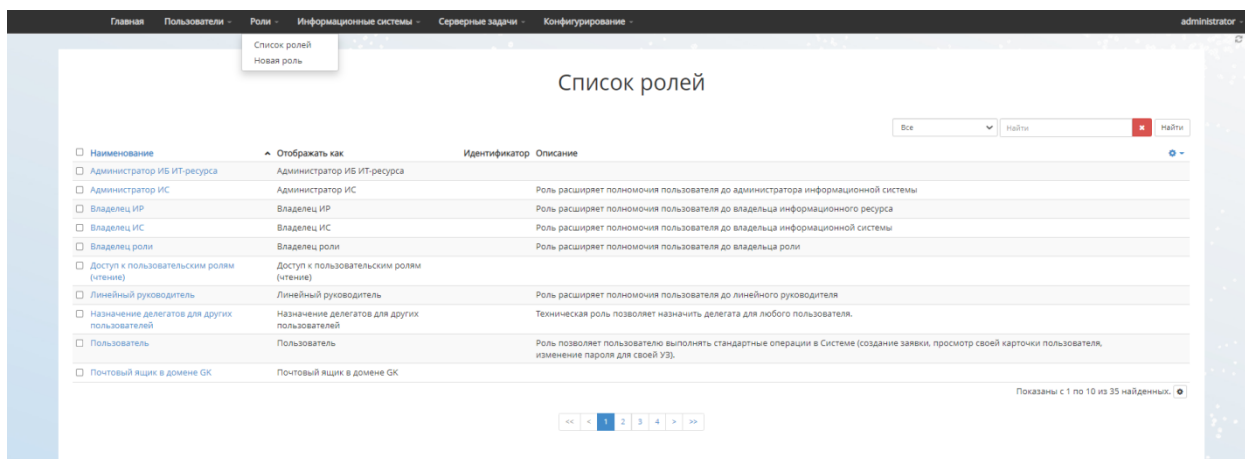


Рис. 24 – Список ролей

3.5.2.1 Управление списком ролей: просмотр, фильтрация, поиск и удаление ролей

В разделе **Список ролей** (см. Рис. 24) отображается таблица с данными существующих в системе ролей. Администратор может:

- отфильтровать данные в таблице, отобразив только требуемые роли. Для этого следует выбрать необходимое значение в списке: **Все**, **Можно запрашивать**, **Нельзя запрашивать** (см. Рис. 25).

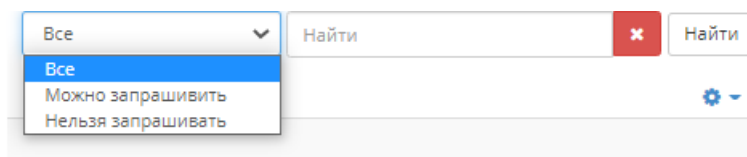


Рис. 25 – Фильтр для отображения конкретных ролей

- найти данные в таблице. Для этого следует ввести требуемое значение в поле **Найти** и нажать соответствующую кнопку. Пример поиска данных приведен на рисунке ниже (см. Рис. 26).

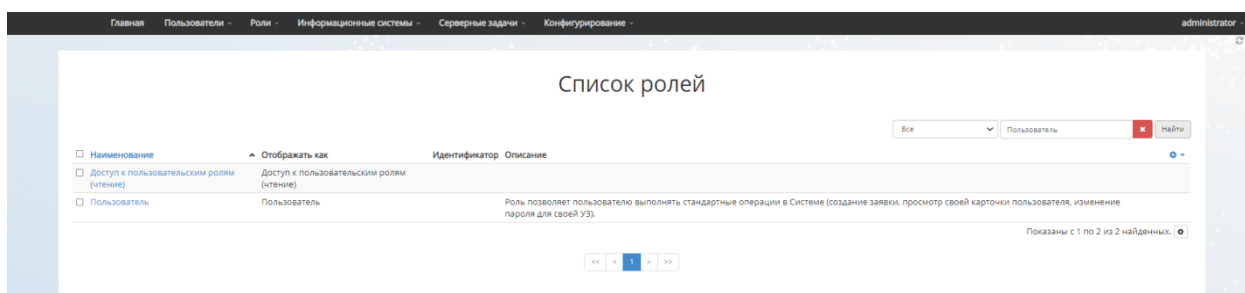


Рис. 26 – Поиск ролей

- удалить роль/роли. Для этого следует:
 - 1) Выбрать требуемые роли, установив соответствующие флажки.
 - 2) В меню действий выбрать пункт **Удалить** (см. Рис. 27).
 - 3) В отобразившемся диалоговом окне подтвердить удаление.

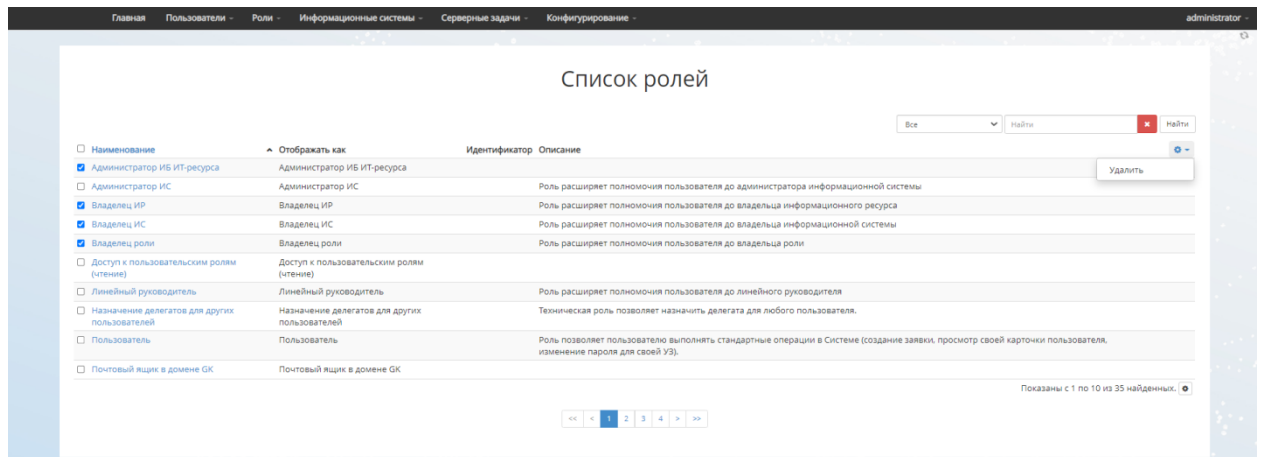


Рис. 27 – Удаление ролей

3.5.2.2 Просмотр и редактирование свойств роли

Для просмотра и/или редактирования свойств роли следует:

- 1) В списке ролей выбрать требуемую роль, перейдя по соответствующей ссылке.

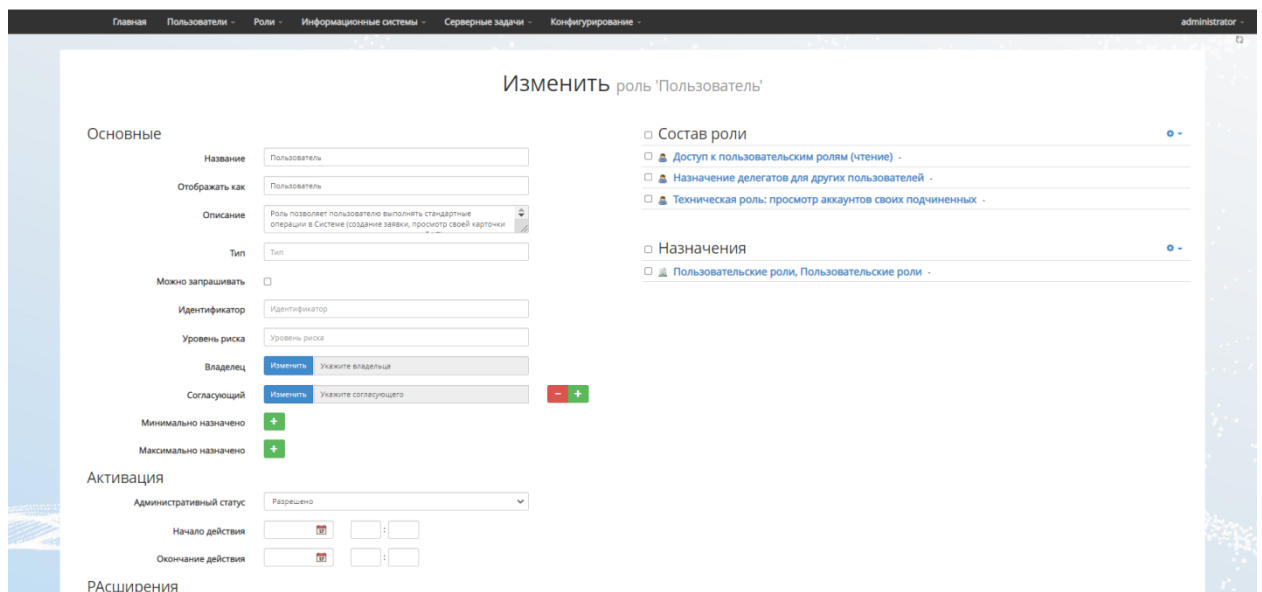


Рис. 28 – Редактирование роли: шаг 1

2) При необходимости отредактировать параметры роли и/или изменить ее состав и назначения.

3) Для сохранения сделанных изменений нажать кнопку **Сохранить** (см. Рис. 29).

The screenshot shows a web application interface for editing a role. The top navigation bar includes links: Главная, Пользователи, Роли, Информационные системы, Серверные задачи, and Конфигурирование. The user is logged in as 'administrator'. The form contains the following fields:

- passportNumber
- passportSeries
- placeOfBirth
- regionOfBirth
- roomNumber
- sapLogin
- telephoneNumber
- uniqEmployeeId
- vacEndDate (with a calendar icon)
- vacStartDate (with a calendar icon)
- vipUser (dropdown menu, currently set to 'Не определено')
- workplaceNumber
- workplacePresence

At the bottom, there are checkboxes for 'Варианты':

- ☐ Принудительно
- ☐ Выполнить реконсильацию
- ☐ Выполнить реконсильацию затронутых объектов
- ☒ Запустить после всех согласований
- ☐ Показать результаты

Buttons at the bottom: 'Назад' and 'Сохранить'.

Рис. 29 – Редактирование роли: шаги 2-3

3.6 Информационные системы: управление ИС, взаимодействующими с ПС

Информационная система (как сущность ПС) - это объект, определяющий настройки соединения и правила взаимодействия ПС с конкретной информационной системой.

3.6.1 Создание ИС

Для создания ИС следует:

1) В главном меню выбрать пункт **Информационные системы>Новая информационная система**.

2) В разделе **Новая информационная система > Основные свойства** задать значения базовых параметров:

- **Название системы** – наименование системы, например, **Орг. структура**;
- **Описание** – описание системы (поле не является обязательным для заполнения);
- **Сервер коннекторов** – внешний сервер, на котором работают коннекторы. При необходимости значение выбирается из списка серверов.

Примечание

В списке серверов коннекторов отображаются ранее установленные в системе серверы коннекторов. Описание процесса установки приведено в документе выше (см. раздел 2).

- **Коннектор** – коннектор, с помощью которого система будет взаимодействовать с ПС. Значение выбирается из списка, перечень коннекторов в списке зависит от выбранного сервера коннекторов. Если сервер не выбран, в списке содержатся коннекторы, работающие на внутреннем сервере.

Рис. 30 – Создание ИС: задание основных свойств

3) Нажать кнопку **Вперед**.

4) В разделе **Новая информационная система** > **Конфигурация** (см. Рис. 31) задать значения конфигурационных параметров. Набор конфигурационных параметров зависит от того, к какому внешнему ресурсу (AD, LDAP, Exchange, таблица БД и т.д.) выполняется подключение (см. разделы 4.1.2.1 и 4.1.2.3).

Рис. 31 – Создание ресурса: задание конфигурационных параметров

5) При необходимости нажать кнопку **Сохранить и проверить соединение**. В результате нажатия кнопки система выполнит процедуру подключения к ресурсу, после чего на экране отобразится сообщение с информацией об успешном или неуспешном подключении.

6) Нажать кнопку **Вперед**.

7) При необходимости в разделе **Новая информационная система > Схема** (см. Рис. 32) проверить корректность схемы исходных данных. На вкладке **Схема** в блоке **Object class** можно выбрать объекты схемы, в результате чего отобразится информация о выбранном объекте (блок **Details**), а также атрибуты объекта (блок **Attributes**). На вкладке **Xml** можно просмотреть xml-код схемы данных.

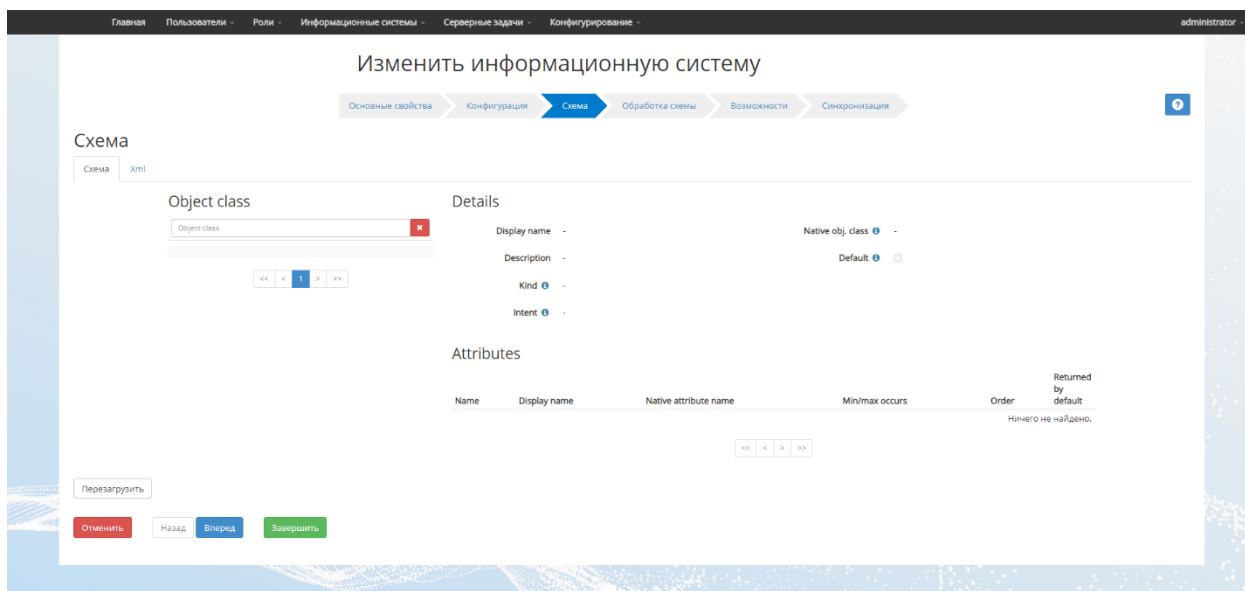


Рис. 32 – Создание ресурса: проверка схемы данных

8) Нажать кнопку **Вперед**.

9) При необходимости в разделе **Новая информационная система Обработка схемы** (см. Рис. 33) настроить соответствие исходной информации данным ПС (точнее, соответствие полей данных ПС полям в подключаемой системе).

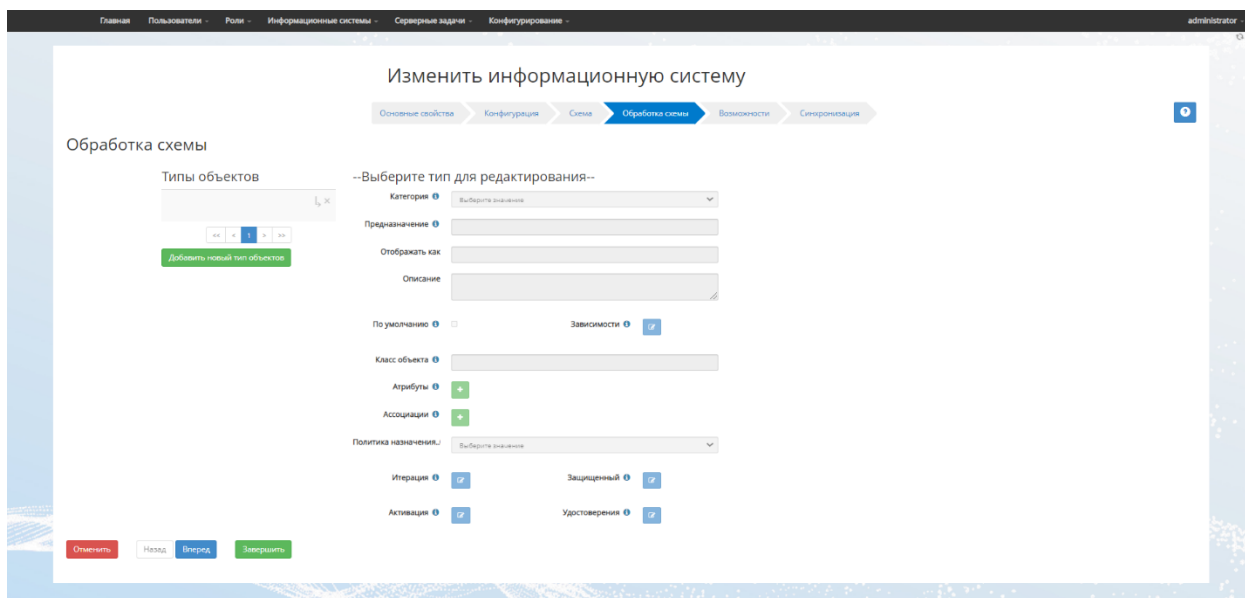


Рис. 33 – Создание ИС: настройка соответствия данных ПС данным в управляемой системе

10) Нажать кнопку **Вперед**.

11) Пропустить этап настройки дополнительных возможностей (раздел **Новая информационная система> Возможности**), нажав кнопку **Вперед**.

12) В разделе **Новая информационная система> Синхронизация** в блоке **Объекты синхронизации** выбрать требуемый объект и задать необходимые параметры синхронизации, определяющие способы и методы обработки информации во время синхронизации:

- **Focus** – тип загружаемого объекта (при подключении ресурса в систему загружаются данные):

- o UserType (пользователь);
- o OrgType (организационная структура);
- o RoleType (роль).

- **Reaction** – поведение системы в различных ситуациях:

- o UNMATCHED – система не нашла объекта с указанным идентификатором;

- o LINKED – в системе уже присутствует объект с указанным идентификатором, и он уже связан с исходным значением в источнике данных;

- o DELETED – система определила, что объект был удалён в источнике данных, но он ещё присутствует в системе;

- o UNLINKED – в системе уже присутствует объект с указанным идентификатором, но объект не связан с исходным значением в источнике данных.

Изменить информационную систему

Синхронизация

Объекты синхронизации --Select sync object to edit--

Name:

Description:

Kind:

Intent:

Object class:

Focus:

Enabled: ☐ Reconcile: ☐

Correlation:

Confirmation:

Condition:

Object Template:

Opportunistic:

Reaction:

Buttons: Отменить, Назад, Вперед, Завершить

Рис. 34 – Создание ИС: задание параметров синхронизации

13) Нажать кнопку **Завершить**.

3.6.2 Управление списком ИС: просмотр, поиск, редактирование и удаление ИС

Управление списком информационных систем выполняется в разделе **Информационные системы, подключенные к ПС** (см. Рис. 35). Для перехода в этот раздел необходимо в главном меню выбрать пункт **Информационные системы > Список информационных систем**.

Информационные системы

Название	Пакет	Версия	Последний статус	Содержание
Данные из ЕКБД	com.evolveum.polygon.connector-databaseable	1.4.0.49	✓	Учетные записи
Получатели из ЕКБД	com.evolveum.polygon.connector-databaseable	1.4.0.49	✓	Учетные записи
Пользователи из ЕКБД	com.evolveum.polygon.connector-databaseable	1.4.0.49	✓	Учетные записи
СУИТ	rusalsecurity.connectors.universal-connector	1.0	✓	Учетные записи
Трудустройства из ЕКБД	com.evolveum.polygon.connector-databaseable	1.4.0.49	✓	Учетные записи
AD - gluosatom.local	Exchange.Connector	1.4.7.5	✓	Учетные записи
AD - inter.rosatom.local	Exchange.Connector	1.4.7.5	✓	Учетные записи
ReverseProxyEOSDO	rusalsecurity.connectors.universal-connector	1.0	✓	Учетные записи

Показаны с 1 по 8 из 8 найденных.

Серверы коннекторов

Название	Хост	Порт	Таймаут	Защищено
Connector Server for gluosatom.local	10.201.48.79	8759	60000	☐
Connector Server for inter.rosatom.local	10.201.48.74	8759	60000	☐

Рис. 35 – Раздел Информационные системы, подключенные к ПС

3.6.2.1 Поиск ИС

Для поиска конкретной ИС следует:

- 1) В поле **Найти** ввести название (или часть названия) искомой ИС.
- 2) Нажать кнопку **Найти**.

Пример поиска данных и отображения его результатов приведен на рисунке ниже (см. Рис. 36).

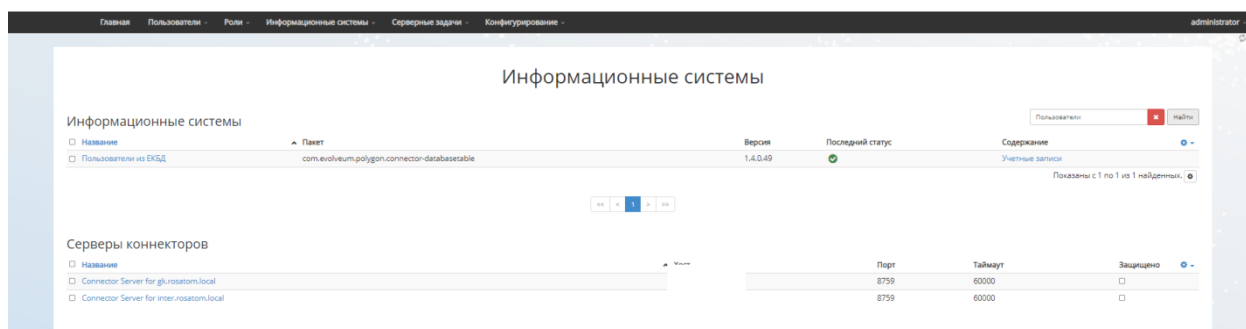


Рис. 36 – Результат поиска ресурса

3.6.2.2 Операции с ИС: удаление, изменение данных

Администратор может выполнить с конкретным ресурсом следующие операции (см. Рис. 37):

- просмотреть (см. раздел 3.6.2.2.1) и при необходимости изменить (см. раздел 3.6.2.2.2) параметры ИС в интерфейсе (**Изменить**);
- изменить параметры ИС, отредактировав его XML-описание (**Изменить XML**, см. раздел 3.6.2.2.3);
- удалить информацию о том, когда последний раз была выполнена синхронизация данных (**Удалить токен синхронизации**, см. раздел 3.6.2.2.4);
- удалить ИС (**Удалить**, см. раздел 3.6.2.2.5).

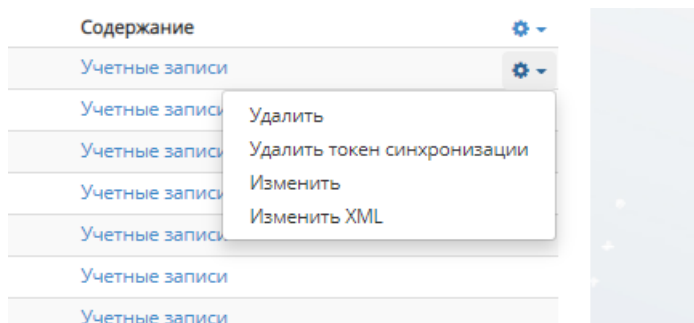


Рис. 37 – Меню действий с ресурсом

3.6.2.2.1 Просмотр параметров ИС

Для просмотра параметров ИС следует в списке ресурсов (**Информационные системы > Список информационных систем**) выбрать требуемую ИС, перейдя по соответствующей ссылке (см. Рис. 38).

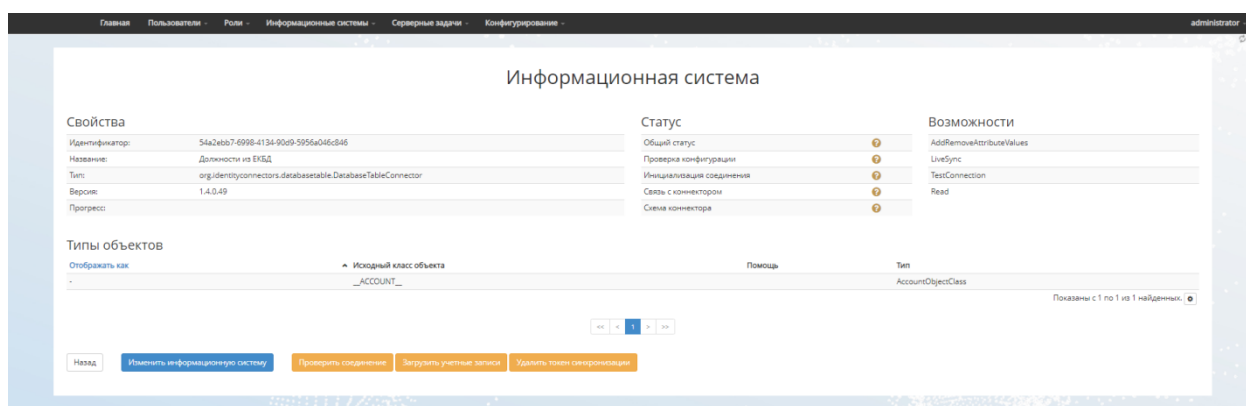


Рис. 38 – Просмотр параметров ресурса

Для перехода в режим редактирования ресурса следует нажать кнопку **Изменить информационную систему** (см. раздел 3.6.2.2.2).

Для проверки соединения ПС с ИС / загрузки учетных записей из ИС в ПС / удаления токена синхронизации (подробнее о токене синхронизации см. в разделе 3.6.2.2.4) следует нажать соответствующую кнопку. После этого на экран будет выдано сообщение о результате операции.

3.6.2.2.2 Редактирование параметров ИС

Для редактирования параметров ресурса следует:

1) В списке ресурсов (**Информационные системы > Список информационных систем**) выбрать требуемую информационную систему, перейдя по соответствующей ссылке (см. Рис. 38).

2) Нажать кнопку **Изменить информационную систему**;

3) Последовательно в подразделах **Основные свойства, Конфигурация, схема, Обработка схемы, Возможности, Синхронизация** изменить/добавить/удалить необходимые данные (описание параметров см. в разделе 3.6.1). Для перехода к следующему подразделу нажать кнопку Вперед.

Для сохранения сделанных изменений и выхода из режима редактирования ресурса нажать кнопку **Завершить** (см. Рис. 39).

Рис. 39 – Редактирование параметров ИС

3.6.2.2.3 Редактирование XML-описания ИС

Для редактирования XML-описания ИС следует:

1) В списке ресурсов (**Информационные системы > Список информационных систем**) в меню действий с конкретной ИС выбрать пункт Изменить XML (см. Рис. 40).

2) В разделе **Object Details** установить флажок **Изменить**.

3) Отредактировать необходимые данные.

4) Нажать кнопку **Save**.

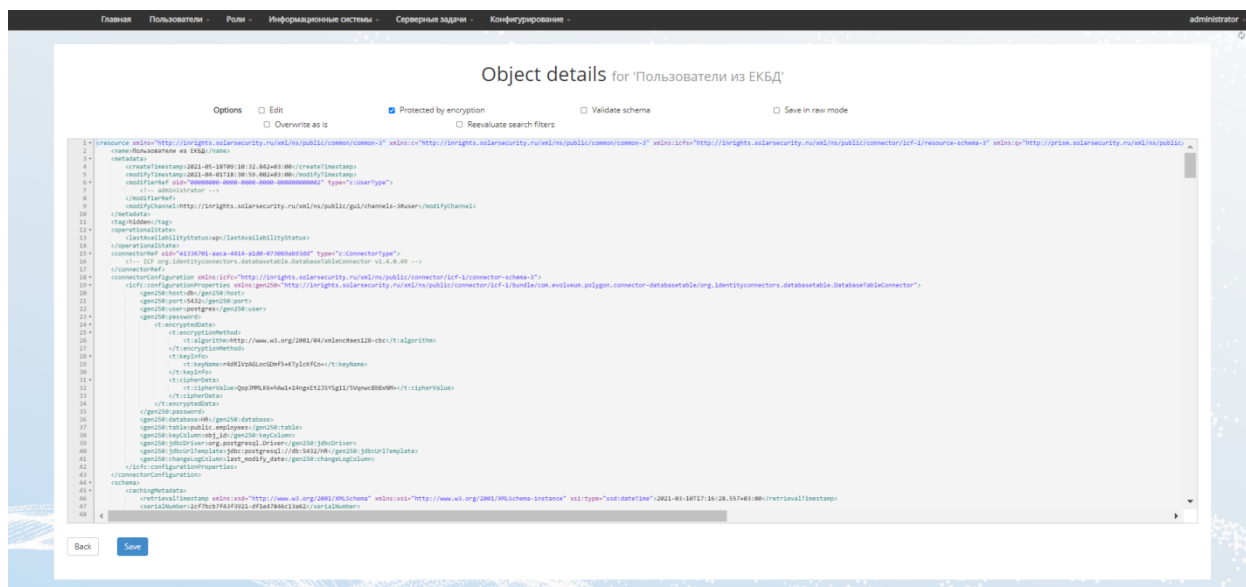


Рис. 40 – Редактирование XML-описания ИС

3.6.2.2.4 Удаление токена синхронизации

Токен синхронизации — это строка, содержащая информацию о том, когда последний раз была выполнена синхронизация данных между ПС и конкретной ИС. Токен синхронизации дает возможность загружать в ПС только те данные, которые обновились в управляемой системе после указанного в токене времени. Если же необходимо загрузить все данные, то токен синхронизации требуется удалить.

Для удаления токена синхронизации, относящегося к конкретной ИС, следует в меню действий с ресурсом выбрать пункт **Удалить токен синхронизации** (см. Рис. 41).

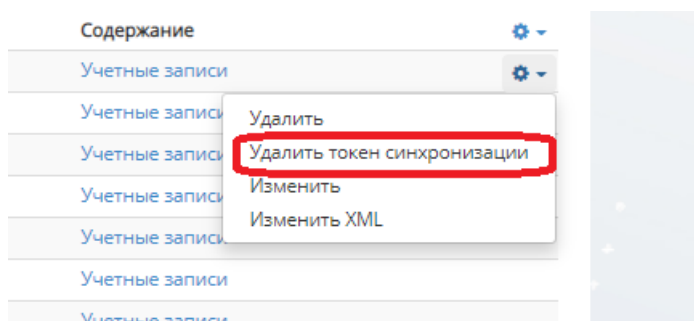


Рис. 41 – Удаление токена синхронизации

Кроме того, токен синхронизации можно удалить, перейдя в режим просмотра параметров ресурса и нажав кнопку **Удалить токен синхронизации** (см. раздел 3.6.2.2.1).

3.6.2.2.5 Удаление ИС

Для удаления конкретного ресурса необходимо в меню действий с ИС (см. Рис. 37) выбрать пункт **Удалить** и в отобразившемся диалоговом окне подтвердить удаление.

Кроме того, можно удалить сразу несколько ресурсов. Для этого следует:

- 1) Выбрать требуемые ресурсы, установив соответствующие флажки (см. Рис. 42);
- 2) В меню общих действий с ресурсами выбрать пункт **Удалить** (см. Рис. 42);
- 3) В отобразившемся диалоговом окне подтвердить удаление.

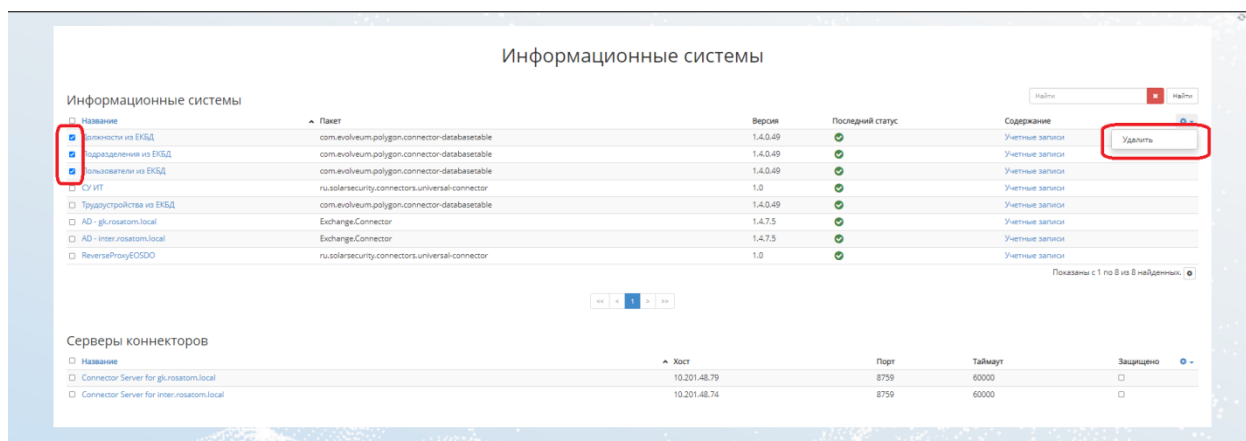


Рис. 42 – Удаление ресурсов

3.7 Серверные задачи: управление задачами, выполняемыми на сервере ПС

Серверные задачи выполняются на сервере ПС в фоновом режиме. К таким задачам можно отнести, например, синхронизацию и реконсильацию данных. При создании и настройке серверных задач

администратор может указать, например, период времени, через который требуется запустить синхронизацию данных.

3.7.1 Управление списком задач

Управление списком задач выполняется в разделе **Серверные задачи** (см. Рис. 43). Для перехода в этот раздел необходимо в главном меню выбрать пункт **Серверные задачи > Список задач**.

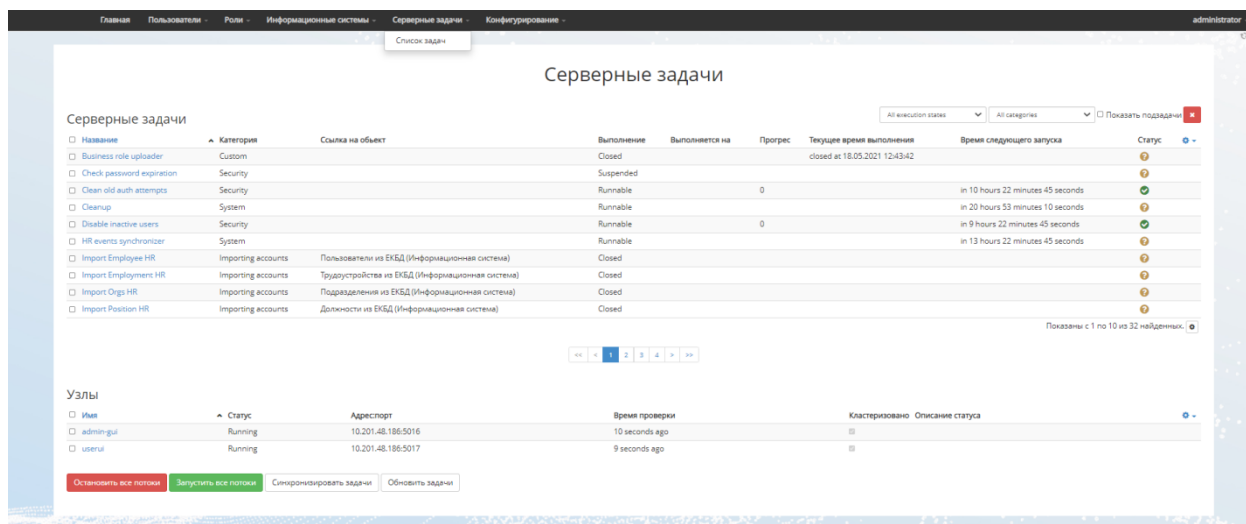


Рис. 43 – Раздел Серверные задачи

В разделе Серверные задачи (см. Рис. 43) в таблице **Серверные задачи** отображаются данные об имеющихся в системе задачах, в таблице Узлы – данные об узлах (серверах), где выполняются задачи:

- **adminguiNode** – сервер, где установлен AGUI;
- **useruiNode** – сервер, где установлен пользовательский GUI.

Данные в таблице **Tasks** можно отфильтровать по статусу выполнения и/или по категории, выбрав необходимые значения в соответствующих списках (**All execution states** и **All categories**). Кроме того, можно показать/скрыть подзадачи, установив/сняв флажок **Show subtasks**.

С помощью кнопок **Остановить все потоки** и **Запустить все потоки** администратор может остановить/запустить все потоки выполнения задач на всех узлах.

С помощью кнопок **Синхронизировать задачи** и **Обновить задачи** администратор может синхронизировать и обновить данные по задачам, что обеспечивает внутреннюю согласованность данных в БД.

3.7.1.1 Групповые операции с задачами: запуск, приостановка и возобновление работы

Администратор может выполнить с одной или несколькими задачами следующие операции (см. Рис. 44):

- приостановить выполнение задач (**Остановить**);
- возобновить выполнение задач (**Продолжить**);
- запустить (**Запустить сейчас**).

Для этого следует:

1) Выбрать требуемые задачи, установив соответствующие флажки.

2) В меню действий выбрать соответствующий пункт (см. Рис. 44). В результате на экране отобразится блок с информацией об успешном или неуспешном выполнении операции.

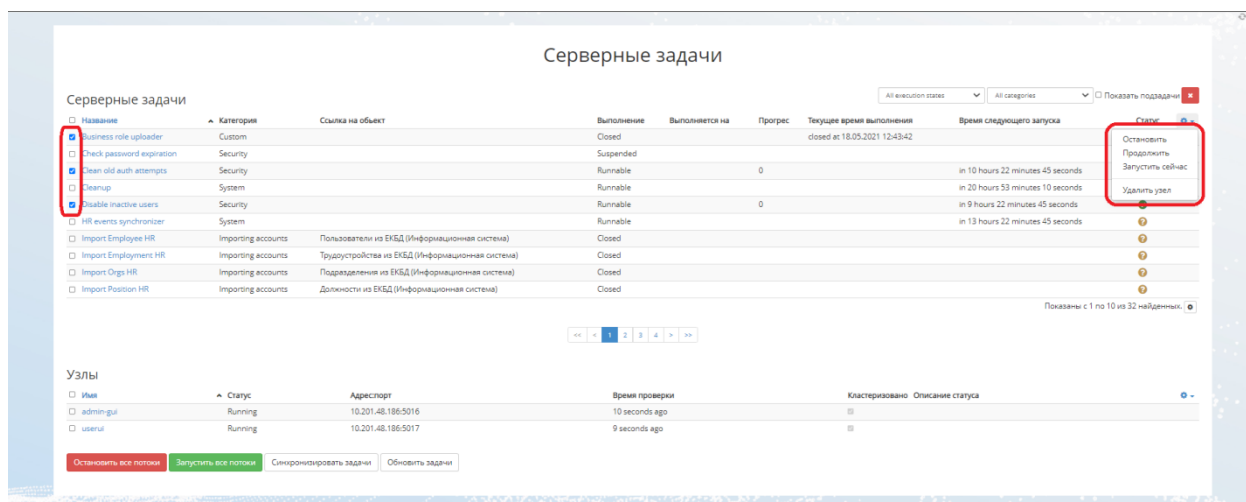


Рис. 44 – Групповые операции с задачами

3.7.1.2 Удаление задач

Для удаления задач следует:

1) В таблице **Tasks** выбрать требуемые задачи, установив соответствующие флажки.

2) В меню действий выбрать пункт **Удалить узел** (см. Рис. 45).

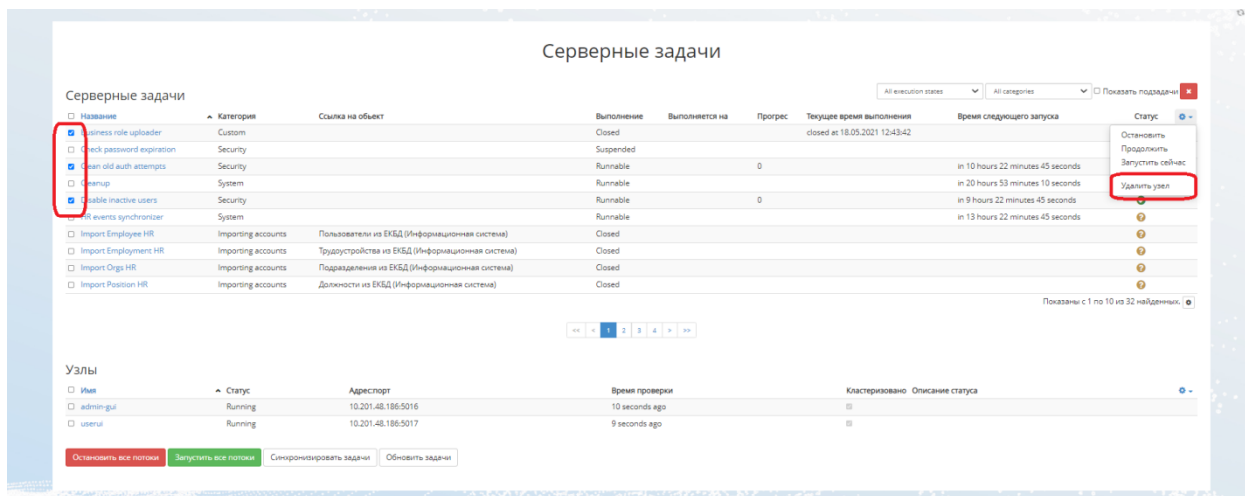


Рис. 45 – Удаление задач

3.7.1.3 Управление задачами, выполняющимися на конкретном сервере (узле)

Администратор может выполнить следующие операции (см. Рис. 46):

- остановить планировщик задач (**Остановить планировщик задач**);
- остановить планировщик задач и выполнение задач (**Остановить планировщик задач и задачи**);
- запустить планировщик задач (**Запустить планировщик задач**);
- отключить узел (**Удалить**). Отключение узла означает, что узел не будет участвовать в обработке фоновых задач.

Для управления задачами, выполняющимися на конкретном сервере (узле), следует:

1) В таблице **Узлы** выбрать требуемые узлы, установив соответствующие флажки;

2) В меню действий выбрать соответствующий пункт (см. Рис. 46). В результате на экране отобразится блок с информацией об успешном или неуспешном выполнении операции.

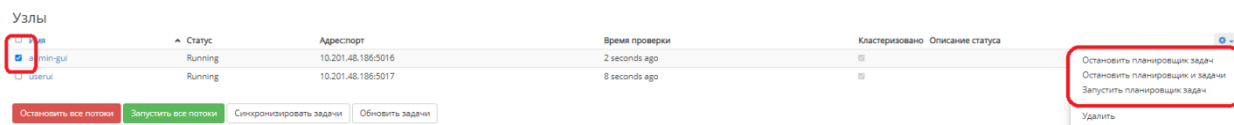


Рис. 46 – Групповые операции с задачами, выполняющимися на конкретном сервере

3.7.2 Редактирование параметров задачи

Для редактирования параметров задачи следует:

1) В списке задач выбрать требуемую задачу, перейдя по соответствующей ссылке (см. Рис. 47).

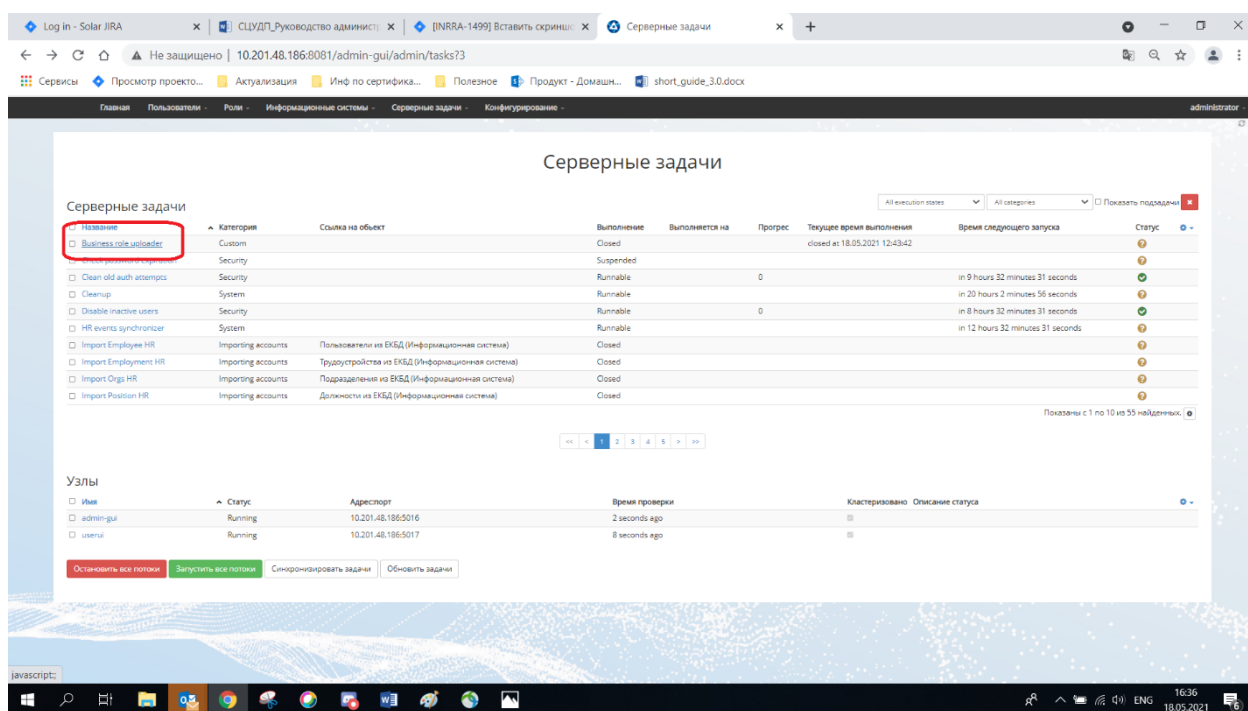


Рис. 47 – Выбор задачи

2) В отобразившемся разделе **Task details** нажать кнопку **Edit**.

Task details

Basic

Task name: Business role uploader

Description: Task to upload business roles (ApplicationObject) from Excel files. Task parameters can be changed via /debug page.

OID: task.ra.uploader.businessroles

Identifier: task.ra.uploader.businessroles

Category: Custom

Parent task:

Handler URI: http://nrights.solarsecurity.ru/ml/ins/publicmodel/scripting/groovy

Custom param: /home/tomcat/2Scripts/BusinessRolesLoader.groovy

Execution status: Closed at 18.05.2021 12:43:42

Resource reference: Выберите значение

Kind: Choose one

Intent:

Object class:

Worker threads:

Thread action

Thread stop action: Close

Operation result

Token	Operation	Status	Message
1000000000000011391	ru.solarsecurity.inrights.task.api.Task.run	UNKNOWN	

Options: ☐ Dry run

Buttons: Back, **Edit**, Run now

Рис. 48 – Переход к редактированию задачи

- 3) Отредактировать требуемые параметры задачи (см. Рис. 49);
- 4) Нажать кнопку **Save** (см. Рис. 49).

Task details

Basic

Task name: Business role uploader

Description: Task to upload business roles (ApplicationObject) from Excel files. Task parameters can be changed via /debug page.

OID: task.ra.uploader.businessroles

Identifier: task.ra.uploader.businessroles

Category: Custom

Parent task:

Handler URI: http://nrights.solarsecurity.ru/ml/ins/publicmodel/scripting/groovy

Custom param: /home/tomcat/2Scripts/BusinessRolesLoader.groovy

Execution status: Closed at 18.05.2021 12:43:42

Resource reference: Выберите значение

Kind: Choose one

Intent:

Object class:

Worker threads:

Thread action

Thread stop action: Close

Operation result

Token	Operation	Status	Message
1000000000000011391	ru.solarsecurity.inrights.task.api.Task.run	UNKNOWN	

Options: ☐ Dry run

Buttons: Back, **Save**, Run now

Рис. 49 – Редактирование параметров задачи

3.8 Конфигурирование: настройка работы ПС

Администратору доступны следующие действия:

- импорт объектов в систему (см. раздел 3.8.1);
- управление объектами репозитория (см. раздел 3.8.2);
- просмотр и изменение параметров работы ПС (см. раздел

3.8.3);

- управление ресурсными объектами;

Примечание

Поскольку по отношению к ПС внешние ИС и БД (т.е. ресурсные объекты) являются компонентами среды функционирования, описание процесса управления ресурсными объектами не входит в рамки этого документа.

3.8.1 Загрузка (импорт) объектов в систему

ПС предоставляет администратору возможность импорта (загрузки) объектов в систему. При этом можно импортировать объекты из указанного администратором файла или ввести XML-описание объекта в специальное поле.

Для загрузки объекта следует:

1) В главном меню выбрать пункт **Конфигурирование> Загрузить объект**.

2) В разделе **Загрузка объектов** в блоке **Загрузить объекты из** выбрать требуемый способ импорта объекта:

- File;
- Embedded editor.

3) В зависимости от выбранного в п.2 способа, указать требуемый файл или ввести соответствующее XML-описание объекта.

4) При необходимости в блоке **Параметры** установить / снять требуемые флажки:

- **Fetch resource schema** – перестроить схему ресурса после импорта объекта;
- **Keep OID** – сохранить существующий OID в случае повторного импорта;
- **Overwrite existing object** – перезаписать существующие объекты;
- **Referential integrity** – проверить валидность ссылок на другие объекты перед импортом;

- **Summarize successes** – отобразить подробную информацию о результатах импорта;
- **Validate dynamic schema** – проверить валидность динамической схемы перед импортом объекта;
- **Validate static schema** – проверить валидность статической схемы перед импортом объекта;
- **Summarize errors** – отобразить подробную информацию об ошибках импорта.

5) При необходимости в поле **Остановка после превышения количества ошибок** указать лимит (количество) ошибок, после достижения которого процесс загрузки объектов должен завершиться (прерваться).

6) Нажать кнопку **Добавить объект**.

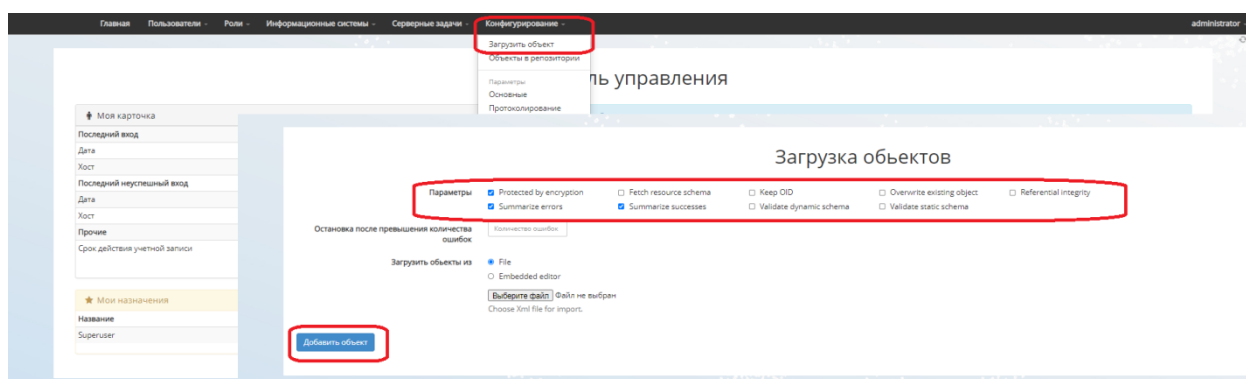


Рис. 50 – Импорт объектов в систему

3.8.2 Управление объектами репозитория

Администратор может экспортировать (в формат XML) или удалять объекты, хранящиеся в репозитории системы, а также просматривать и редактировать XML-файл, содержащий параметры конкретного объекта. Указанные операции выполняются в разделе **Объекты из хранилища** (см. Рис. 51). Для перехода в этот раздел необходимо в главном меню выбрать пункт **Конфигурирование>Объекты в репозитории**.

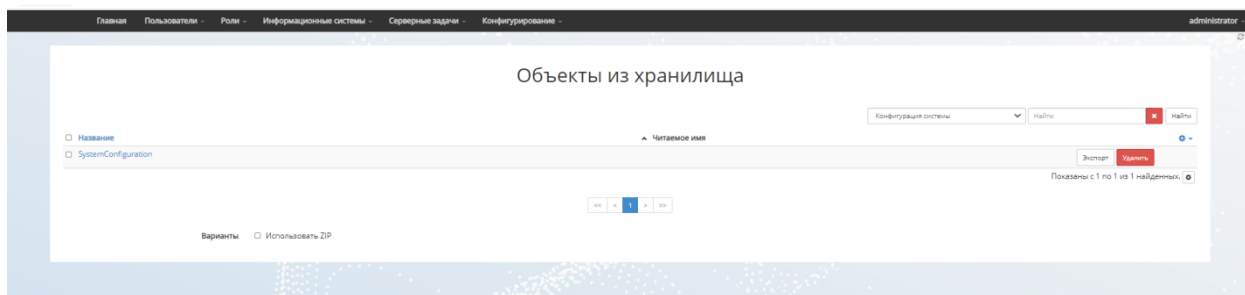


Рис. 51 – Расширенные настройки системы

Для удобства пользователя все объекты объединены в группы. В таблице ниже (см. Таблица 7 –) приведены общие сведения о группах объектов.

Таблица 7 – Перечень совместимых ОС

Группа объектов	Описание
Access certification campaign	Список периодических проверок (например, валидности назначений ролей)
Access certification definition	Шаблоны периодических проверок (например, валидности назначений ролей)
Lookup table	Список механизмов управления выбором и маппингом предустановленных значений
SoD правило	Список правил разграничения полномочий, вызывающих конфликты разграничения полномочий
Value policy	Список политик, определяющих возможные значения атрибутов (например, парольная политика)
Абстрактная роль	Список объектов RoleType и OrgType
Версии маршрутов согласования	Список версий маршрутов согласования
Должность	Список должностей организационной структуры
Доступ информационному ресурсу	Список доступов к информационному ресурсу
Задача	Список всех задач

Запрос	Список заявок на согласование
Информационная система	Список информационных систем (как источников данных, так и целевых)
Информационный ресурс	Список информационных ресурсов
Компания	Список компаний
Коннектор	Список коннекторов
Конфигурация	Список конфигураций, отвечающих за журналирование событий безопасности
Конфигурация системы	Список конфигураций системы
Логическая система	Список логических систем
Маршрут согласования	Список маршрутов согласования, определяющих «путь» заявки в процессе согласования
Назначение	Список назначений
Обобщенный объект	Список обобщенных объектов
Организация	Список объектов OrgType
Отчет	Список шаблонов отчетов
Политика безопасности	Список политик безопасности
Пользователь	Список пользователей
Ресурсный объект	Список теневых объектов
Роль	Список объектов RoleType
Спецификация динамических атрибутов	Список схем динамических атрибутов, расширяющих стандартную схему объектов
Сформированный отчет	Список сформированных отчетов
Трудоустройство	Список трудоустройств пользователей
Узел	Список инсталляций системы
Фокусный объект	Список фокусных объектов
Формы	Список шаблонов форм для настройки вида форм объектов

Хост коннекторов	Список коннектор-хостов
Шаблон объекта	Список шаблонов объектов

3.8.2.1 Поиск объектов

Для поиска объекта следует в разделе **Объекты из хранилища (Конфигурирование>Объекты в репозитории)**:

- 1) Выбрать требуемую группу объектов (поиск ведется только в пределах группы).
- 2) В поле **Найти** ввести требуемое значение.
- 3) Нажать кнопку **Найти** (см. Рис. 52).

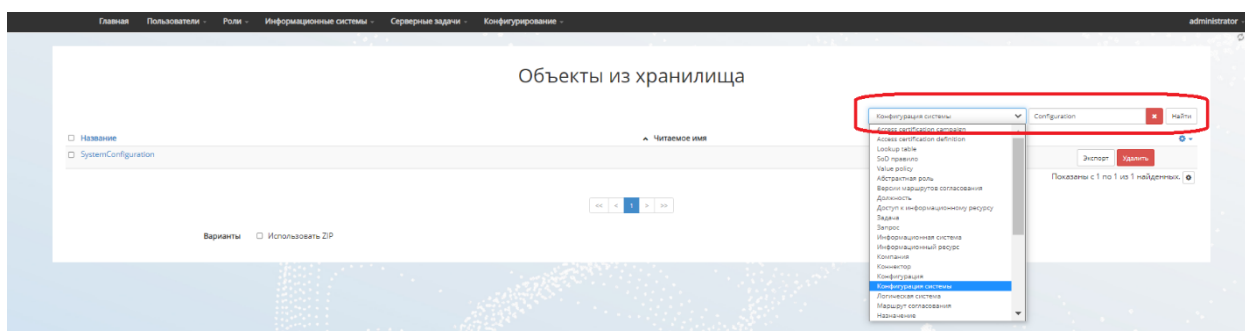


Рис. 52 – Поиск объекта

3.8.2.2 Экспорт объектов репозитория

Для экспорта конкретного объекта следует в разделе **Объекты из хранилища (Конфигурирование>Объекты в репозитории)**:

- 1) 1. Выбрать требуемую группу объектов (см. Рис. 53).
- 2) При необходимости установить флажок **Использовать ZIP** (в этом случае в результате экспорта будет получен ZIP-архив, в котором содержится файл формата xml).
- 3) Нажать кнопку **Экспорт** справа от названия соответствующего объекта.

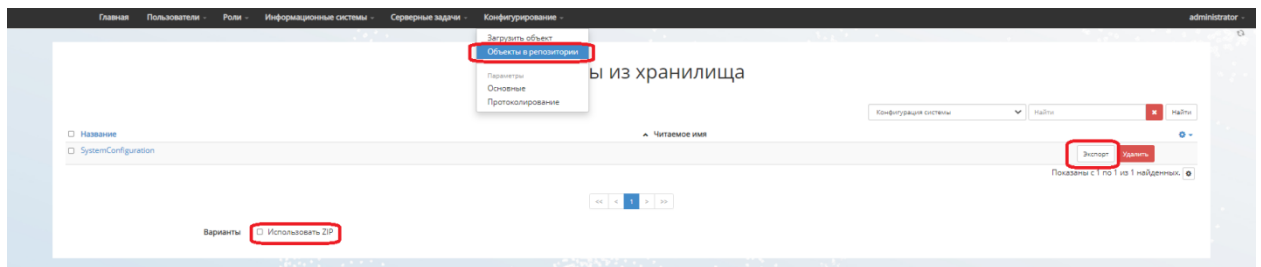


Рис. 53 – Экспорт объекта

Для экспорта нескольких объектов группы следует:

- 1) 1. Выбрать требуемую группу объектов.
- 2) Выбрать требуемые объекты, установив соответствующие флажки.
- 3) При необходимости установить флажок **Использовать ZIP**.
- 4) В меню действий выбрать пункт **Экспорт выбранных** (см. Рис. 54).

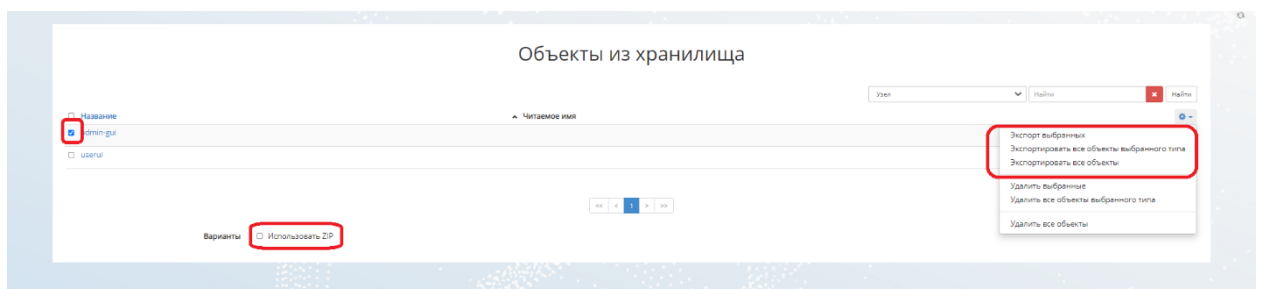


Рис. 54 – Экспорт выбранных объектов

Для экспорта всех объектов группы следует:

- 1) Выбрать требуемую группу объектов.
- 2) При необходимости установить флажок **Использовать ZIP**.
- 3) В меню действий выбрать пункт **Экспортировать все объекты выбранного типа**.

Для экспорта всех объектов следует (вне зависимости от того, какая группа объектов выбрана):

- 1) При необходимости установить флажок **Использовать ZIP**.
- 2) В меню действий выбрать пункт **Экспортировать все объекты**.

3.8.2.3 Удаление объектов репозитория

Удаление объектов репозитория выполняется аналогично экспорту (см. раздел 3.8.2.2), при этом используются кнопка Удалить и пункты меню действий Удалить выбранные и **Удалить все объекты выбранного типа** (см. Рис. 55). Кроме того, после нажатия кнопки **Удалить** / выбора соответствующих пунктов меню необходимо в диалоговом окне подтвердить удаление.

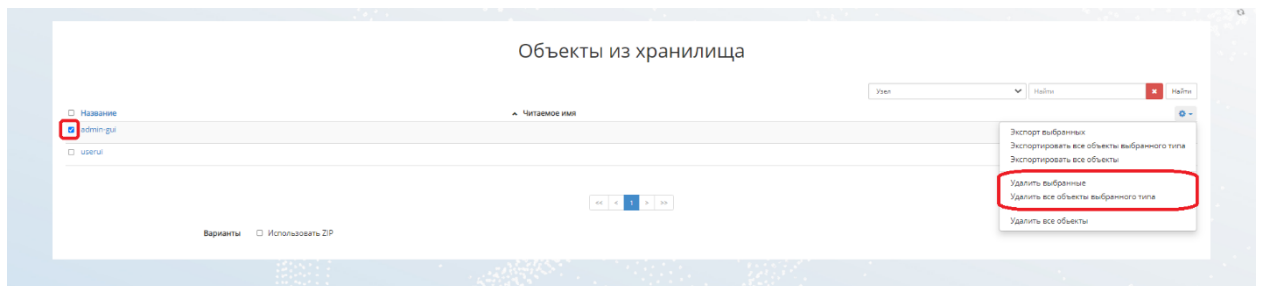


Рис. 55 – Элементы, используемые для удаления объектов

3.8.2.4 Просмотр и редактирование XML-описания объекта

Для просмотра и редактирования XML-описания конкретного объекта следует в разделе Объекты из хранилища (**Конфигурирование>Объекты в репозитории**):

- 1) Выбрать требуемую группу объектов;
- 2) В списке объектов выбрать требуемый объект, перейдя по соответствующей ссылке (см. Рис. 56 – Редактирование XML-описания объекта: шаги 1-2;

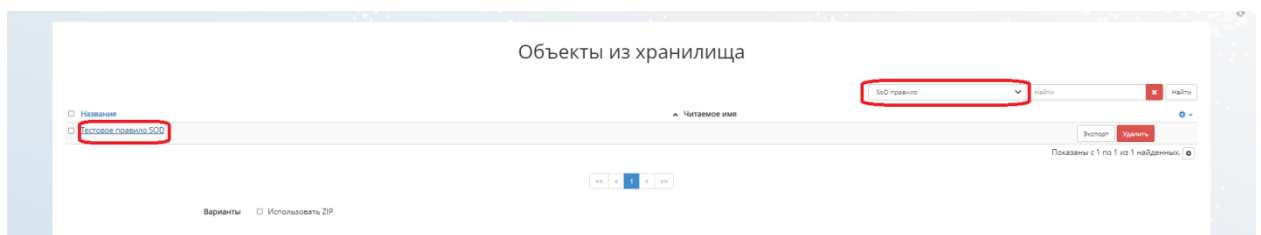


Рис. 56 – Редактирование XML-описания объекта: шаги 1-2

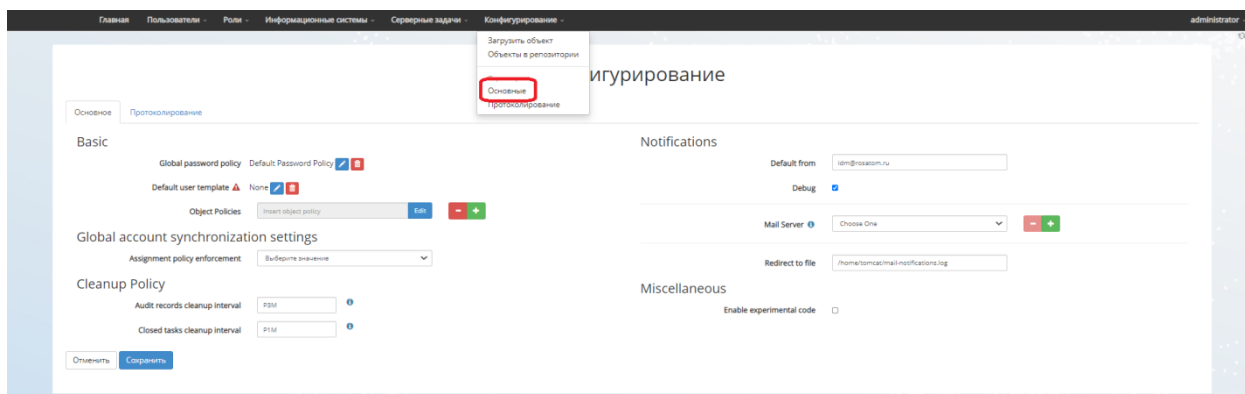


Рис. 58 – Раздел Конфигурирование

На вкладке **Основное** собрана информация об основных параметрах системы (см. 3.8.3.1), вкладка **Протоколирование** предназначена для работы с параметрами журналирования (см. 3.8.3.2).

Примечание

Для перехода сразу в раздел **Конфигурирование** > **Протоколирование** необходимо в главном меню выбрать пункт **Конфигурирование** > **Протоколирование**.

3.8.3.1 Просмотр и изменение основных параметров работы ПС

В разделе **Конфигурирование** на вкладке **Основное** (см. Рис. 58) администратор может просматривать и редактировать значения основных параметров работы ПС (см. Таблица 8 –). Для сохранения изменений необходимо нажать кнопку **Save**.

Таблица 8 – Основные параметры работы системы

Наименование параметра	Описание
Basic	
Global password policy	Глобальная парольная политика. Используется по умолчанию, если не указана иная парольная политика для объекта

Default user template	Пользовательский шаблон по умолчанию
Object Policies	Объектные политики. Выбор шаблона по умолчанию для любого типа объектов
Global account synchronization settings	
Assignment policy enforcement	Политика назначений
Cleanup Policy	
Audit records cleanup interval	Интервал очистки записей аудита
Closed tasks cleanup interval	Интервал очистки закрытых задач
Notifications	
Default from	Отправитель по умолчанию
Debug	Отладка
Mail Server	Почтовый сервер
Redirect to file	Перенаправлять в файл
Miscellaneous	
Enable experimental code	Использовать экспериментальный код

3.8.3.2 Просмотр и изменение параметров журналирования процессов работы ПС

В разделе **Конфигурирование** на вкладке **Протоколирование** (**Конфигурирование** > **Протоколирование**, см. Рис. 59) администратор может просматривать и редактировать значения параметров журналирования процессов работы ПС. Для сохранения изменений необходимо нажать кнопку **Save**.

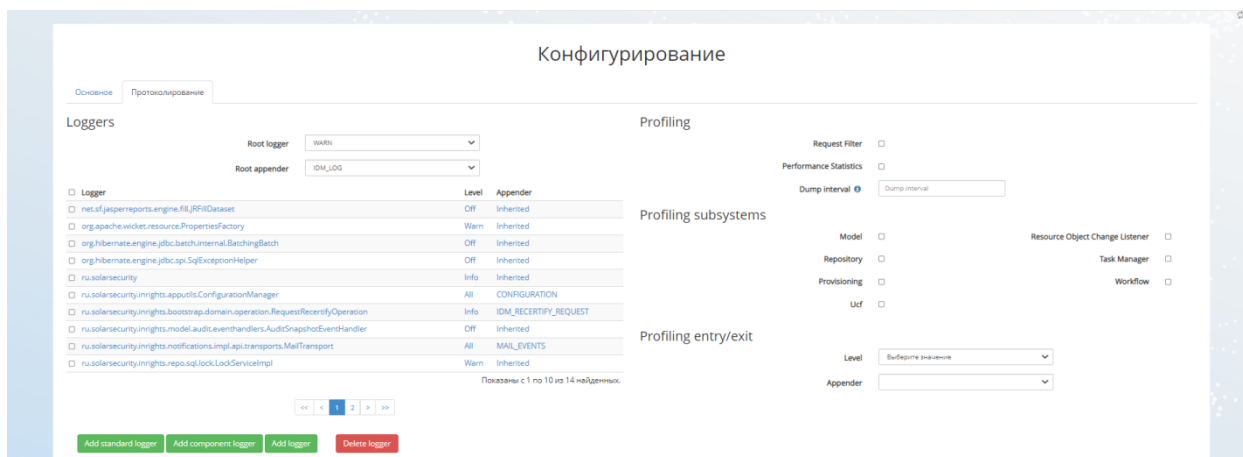


Рис. 59 – Раздел Конфигурирование вкладка Протоколирование

3.8.3.2.1 Блок **Loggers**: задание компонентов/процессов системы, информацию о работе которых необходимо получить

В блоке **Loggers** (см. Рис. 60) можно задать:

- Основной уровень журналирования (выбирается из списка **Root logger**) для компонентов – уровень журналирования, который по умолчанию будет задан для всех новых компонентов (**Inherited**);
- Основной журнальный файл, в который следует сохранять информацию о работе компонентов системы (выбирается из списка **Root appender**);
- Компоненты/процессы, о работе которых следует получать информацию. При этом для каждого компонента можно указать уровень журналирования и журнальный файл.

Loggers

Root logger WARN

Root appender IDM_LOG

☐ Logger	Level	Appender
☐ net.sf.jasperreports.engine.fill.JRFillDataset	Off	Inherited
☐ org.apache.wicket.resource.PropertiesFactory	Warn	Inherited
☐ org.hibernate.engine.jdbc.batch.internal.BatchingBatch	Off	Inherited
☐ org.hibernate.engine.jdbc.spi.SqlExceptionHelper	Off	Inherited
☐ ru.solarsecurity	Info	Inherited
☐ ru.solarsecurity.inrights.apputils.ConfigurationManager	All	CONFIGURATION
☐ ru.solarsecurity.inrights.bootstrap.domain.operation.RequestRecertifyOperation	Info	IDM_RECERTIFY_REQUEST
☐ ru.solarsecurity.inrights.model.audit.eventhandlers.AuditSnapshotEventHandler	Off	Inherited
☐ ru.solarsecurity.inrights.notifications.impl.api.transports.MailTransport	All	MAIL_EVENTS
☐ ru.solarsecurity.inrights.repo.sql.lock.LockServiceImpl	Warn	Inherited

Показаны с 1 по 10 из 14 найденных.

<<
<
1
2
>
>>

Add standard logger
Add component logger
Add logger
Delete logger

Рис. 60 – Блок Loggers

Для добавления требуемых компонентов/процессов следует воспользоваться кнопками **Add standart logger**, **Add component logger** или **Add logger**.

Нажав кнопку **Delete logger**, можно удалить заранее выбранные (с помощью установки соответствующих флажков) компоненты/процессы.

ВАЖНО! Для сохранения сделанных изменений необходимо нажать кнопку **Save**.

Для изменения уровня журналирования и/или выбора другого журнального файла для уже заданного компонента/процесса следует:

- 1) В списке компонентов/процессов выбрать требуемый компонент/процесс, перейдя по соответствующей ссылке (см. Рис. 61).
- 2) В соответствующих списках выбрать требуемые значения **Level** и/или **Appender** (см. Рис. 61).
- 3) Нажать кнопку **Save**.

Logger	Level	Appender
☐ net.sf.jasperreports.engine.fill.JRFillDataset	Off	Inherited
☐ org.apache.wicket.resource.PropertiesFactory	Warn	Inherited
☐ org.hibernate.engine.jdbc.batch.internal.BatchingBatch	Off	Inherited
☐ org.hibernate.engine.jdbc.spi.SqlExceptionHelper	Off	Inherited
☐ ru.solarsecurity	Info	Inherited
☐ ru.solarsecurity.inrights.apputils	☐ org.hibernate.engine.jdbc.spi.SqlExceptionHelper	OFF None selected
☐ ru.solarsecurity.inrights.bootstrap	☐ ru.solarsecurity	Info Inherited
☐ ru.solarsecurity.inrights.model	☐ ru.solarsecurity.inrights.apputils.ConfigurationManager	All CONFIGURATION
☐ ru.solarsecurity.inrights.notification	☐ ru.solarsecurity.inrights.bootstrap.domain.operation.RequestRecertifyOperation	Info IDM_RECERTIFY_REQUEST
☐ ru.solarsecurity.inrights.repo	☐ ru.solarsecurity.inrights.model.audit.eventhandlers.AuditSnapshotEventHandler	Off Inherited
	☐ ru.solarsecurity.inrights.notifications.impl.api.transports.MailTransport	All MAIL_EVENTS
	☐ ru.solarsecurity.inrights.repo.sql.lock.LockServiceImpl	Warn Inherited

Рис. 61 – Изменение уровня журналирования / задание журнального файла

3.8.3.2.2 Блок Audit: задание параметров аудита событий ПС

В блоке **Audit** (см. Рис. 62) можно указать необходимость протоколирования событий ПС.

Audit

Audit log

☒

Details

☒

Appender

ACCESS_SYNC

Рис. 62 – Блок Audit

Для этого следует:

- 1) Установить флажки **Audit log** и **Details** (если требуется подробное протоколирование);
- 2) В списке **Appender** выбрать журнальный файл, в который следует сохранять данные аудита событий.

ВАЖНО! Для сохранения сделанных изменений необходимо нажать кнопку **Save**.

3.8.3.2.3 Блок Appenders: задание параметров вывода информации о работе компонентов/процессов системы

В блоке **Appenders** (см. Рис. 63) можно задать:

- параметры консолей для вывода информации о работе системы;

- журнальные файлы, которые будут отображаться в списках **Root logger** и **Appender**.

Appenders						
☐ Name	Pattern	File path	File pattern	Max. history	Max. file size [kB]	Appending
☐ ACCESS_SYNC	%date [%X(subsystem)] [%thread] %level [%logger]: %msg%n	\$(catalina.base)/logs/access_sync.log	\$(catalina.base)/logs/access_sync-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ ACCESS_SYNC	%date [%X(subsystem)] [%thread] %level [%logger]: %msg%n	\$(catalina.base)/logs/access_sync.log	\$(catalina.base)/logs/access_sync-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ CONFIGURATION	%date [%msg%n]	\$(catalina.base)/logs/idm_config.log	\$(catalina.base)/logs/idm-config-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ IDM-PROFILE_LOG	%date [%level: %msg%n]	\$(catalina.base)/logs/idm-profile.log	\$(catalina.base)/logs/idm-profile-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ IDM_LOG	%date [%X(subsystem)] [%thread] %level [%logger]: %msg%n	\$(catalina.base)/logs/idm.log	\$(catalina.base)/logs/idm-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ IDM_RECONF_REQUEST	%date [%X(subsystem)] [%thread] %level [%logger]: %msg%n	\$(catalina.base)/logs/idm_reconf_request.log	\$(catalina.base)/logs/idm_reconf_request-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ IDM_REQUEST_CONVERTER_LOG	%date [%level: %msg%n]	\$(catalina.base)/logs/idm-request-converter.log	\$(catalina.base)/logs/idm-request-converter-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
☐ MAIL_EVENTS	%date [%X(subsystem)] [%thread] %level [%logger]: %msg%n	\$(catalina.base)/logs/mail_events.log	\$(catalina.base)/logs/mail_events-%d{yyyy-MM-dd}.%i.log	10	100MB	☐
Add console appender Add file appender Delete appender						

Рис. 63 – Блок Appenders

Для добавления данных консоли / журнального файла следует воспользоваться кнопками **Add console appender** / **Add file appender**.

Нажав кнопку **Delete appender**, можно удалить заранее выбранные (с помощью установки соответствующих флажков) консоли/журнальные файлы.

ВАЖНО! Для сохранения сделанных изменений необходимо нажать кнопку **Сохранить**.

Для изменения данных уже заданных консолей или журнальных файлов следует:

- 1) В списке консолей/журнальных файлов выбрать требуемую консоль/требуемый файл, перейдя по соответствующей ссылке (см. Рис. 64).

- 2) Отредактировать необходимые значения.

- 3) Нажать кнопку **Сохранить**.

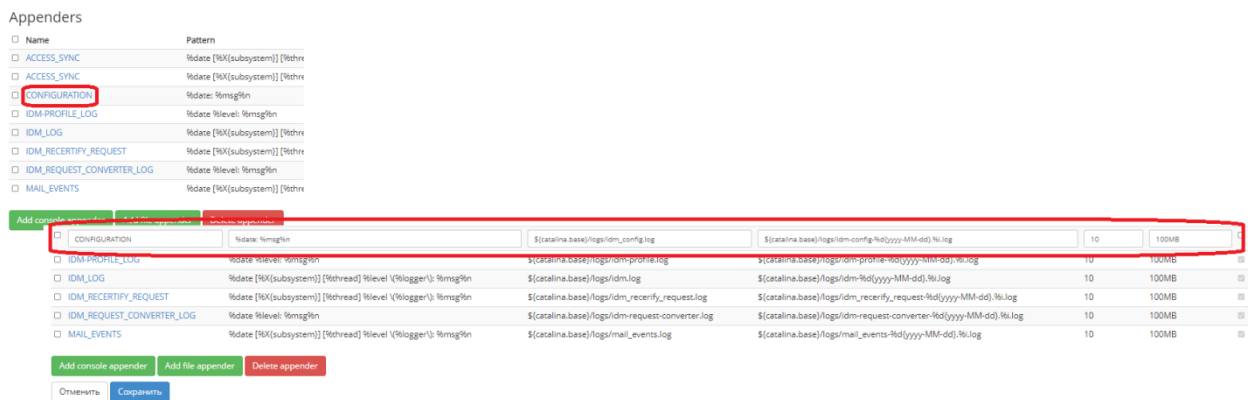


Рис. 64 – Редактирование данных консоли / журнального файла

3.8.3.2.4 Блоки Profiling, Profiling subsystems, Profiling entry/exit: задание параметров журналирования работы профайлера

Профайлер – это инструмент, с помощью которого в ПС обеспечивается сбор данных о производительности системы. В блоках **Profiling, Profiling subsystems, Profiling entry/exit** можно задать параметры профайлера (см. Рис. 65). Для этого следует:

- 1) При необходимости в блоке **Profiling** установить флажки:
 - **Request Filter** – записывать в журнал время выполнения запросов к БД;
 - **Performance Statistics** – записывать в журнал статистические данные по производительности.

2) При необходимости в поле **Dump interval** задать промежуток времени (интервал) в минутах, через который данные будут записываться в журнал (по умолчанию 30 минут).

3) В блоке **Profiling subsystems** выбрать подсистемы, которые требуется мониторить (модели данных, репозитории и т.д.), установив соответствующие флажки.

4) В блоке **Profiling entry/exit** в соответствующих списках выбрать уровень журналирования и журнальный файл, в который требуется записывать данные.

Profiling

Request Filter ☒

Performance Statistics ☒

Dump interval ⓘ

Profiling subsystems

Model	☒	Resource Object Change Listener	☒
Repository	☒	Task Manager	☐
Provisioning	☐	Workflow	☐
Ucf	☐		

Profiling entry/exit

Level

Appender

Рис. 65 – Задание параметров работы профайлера

ВАЖНО! Для сохранения сделанных изменений необходимо нажать кнопку **Сохранить**.

3.9 Изменение паролей УЗ администратора

Для смены пароля администратора следует:

- 1) В меню администратора выбрать пункт **administrator > Сменить пароль**
- 2) В разделе **Мои пароли** в блоке **Пароль** дважды указать новый пароль.
- 3) Выбрать учетную запись администратора, установив соответствующий флажок.

Примечание

Для смены пароля доступа в AGUI следует выбрать соответствующую учетную запись (см. Рис. 66).

- 4) Нажать кнопку **Сохранить**.

Мои пароли

Пароль

Учетные записи

* Важно. When "СЦУДП" account is checked, user password is changed. Besides that, all account credentials are updated based on policies defined in resources definitions.

Имя	Ресурсы	Включено
СЦУДП	Хранилище СЦУДП	☒

Назад Сохранить

Рис. 66 – Смена пароля для доступа администратора в AGUI

4 АДМИНИСТРИРОВАНИЕ ПС

Администрирование системы предполагает различные операции, выполнение которых требует понимания процесса взаимодействия ПС с ИС (см. раздел 1.3).

4.1 Загрузка кадровых данных

4.1.1 Настройка подключения к стандартным источникам данных

Параметры взаимодействия ПС с ИС задаются при настройке соответствующего ресурса. Поэтому для подключения к источнику данных необходимо создать ресурс. При создании ресурса задаются значения конфигурационных параметров, набор которых зависит от того, к какому внешнему ресурсу (AD, таблица БД и т.д.) выполняется подключение.

Создать и настроить ресурс можно как в специальном разделе AGUI (**Информационные системы > Новая информационная система**, подробнее см. раздел 3.6.1), так и с помощью загрузки или редактирования соответствующего XML-описания ресурса.

Примечание

Загрузить описание ресурса можно в разделе **Загрузить объект (Информационные системы > Загрузить описание**

информационной системы, при этом загрузка данных ИС соответствует импорту данных любого объекта ПС (см. раздел 3.8.1).

Отредактировать XML-описание ресурса можно, выбрав требуемый ресурс в разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории**, см. раздел 3.8.2.4).

4.1.1.1 Настройка подключения к БД

Для подключения ПС к БД требуется создать ресурс. При создании ресурса в интерфейсе (Информационные системы>Новая информационная система, подробнее см. раздел 3.6.1) в подразделе **Новая информационная система> Конфигурация** необходимо задать следующие параметры:

- **Name Quoting** - символ, который используется для кавычирования названия колонок (по умолчанию при запросах названия колонок в запросах не кавычируются);
 - **Host** - имя хоста, на котором располагается БД;
 - **TCP Port** - порт, по которому требуется подключиться к БД;
 - **User** - пользователь, от имени которого выполняется подключение к БД;
- **User Password** - пароль, используемый для подключения к БД;
- **Database** - имя БД, к которой выполняется подключение;
- **Table** - таблица (или представление), являющаяся источником исходных данных;
 - **Key Column** - колонка, значения из которой будут считаться ключами для объектов;
 - **Password Column** - колонка, которая содержит информацию с паролем;

- **JDBC Driver** - имя JDBC-драйвера, который используется для подключения (jar-файл с данным драйвером должен быть прописан в classpath приложения или коннектор-хоста);
- **JDBC Connection URL** - URL для подключения к БД.

The screenshot shows a 'Configuration' tab with a sub-tab 'configurationProperties'. The form is titled 'Орг.структура из HR'. It contains the following fields and values:

Field	Value
Name Quoting	
Host	localhost
TCP Port	3306
User	gubber
User Password	
Database	hr
Table	hr_departments
Key Column	department_id
Password Column	
JDBC Driver	com.mysql.jdbc.Driver
JDBC Connection URL	jdbc:mysql://localhost:3306/hr
Enable writing empty string	Не определено
Rethrow all SQLExceptions	Не определено
Native Timestamps	Не определено
All native	Не определено
Validate Connection Query	
Change Log Column (Sync)	
Datasource Path	
Initial JNDI Properties	
Suppress Password	Не определено

Рис. 67 – Настройка подключения к БД: задание значений конфигурационных параметров

Создать и настроить ресурс можно также с помощью загрузки (см. раздел 3.8.1) или редактирования (см. раздел 3.8.2.4) XML-файла, содержащего в секции **connectorConfiguration** необходимые конфигурационные параметры, например:

```
<connectorConfiguration
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/connector-schema-3">

  <icfc:configurationProperties
xmlns:gen290="http://inrights.solarsecurity.ru/xml/
ns/public/connector/icf-1/bundle/

    com.evolveum.polygon.connector-
databasetable/org.identityconnectors.databasetable.
DatabaseTableConnector">

      <gen290:host>localhost</gen290:host>
      <gen290:port>3306</gen290:port>
      <gen290:user>gubber</gen290:user>
      <gen290:password></gen290:password>
      <gen290:database>hr</gen290:database>
      <gen290:table>hr_departments</gen290:table>

<gen290:keyColumn>department_id</gen290:keyColumn>

<gen290:jdbcDriver>com.mysql.jdbc.Driver</gen290:jd
bcDriver>

<gen290:jdbcUrlTemplate>jdbc:mysql://localhost:3306
/hr</gen290:jdbcUrlTemplate>

    </icfc:configurationProperties>
  </connectorConfiguration>
```

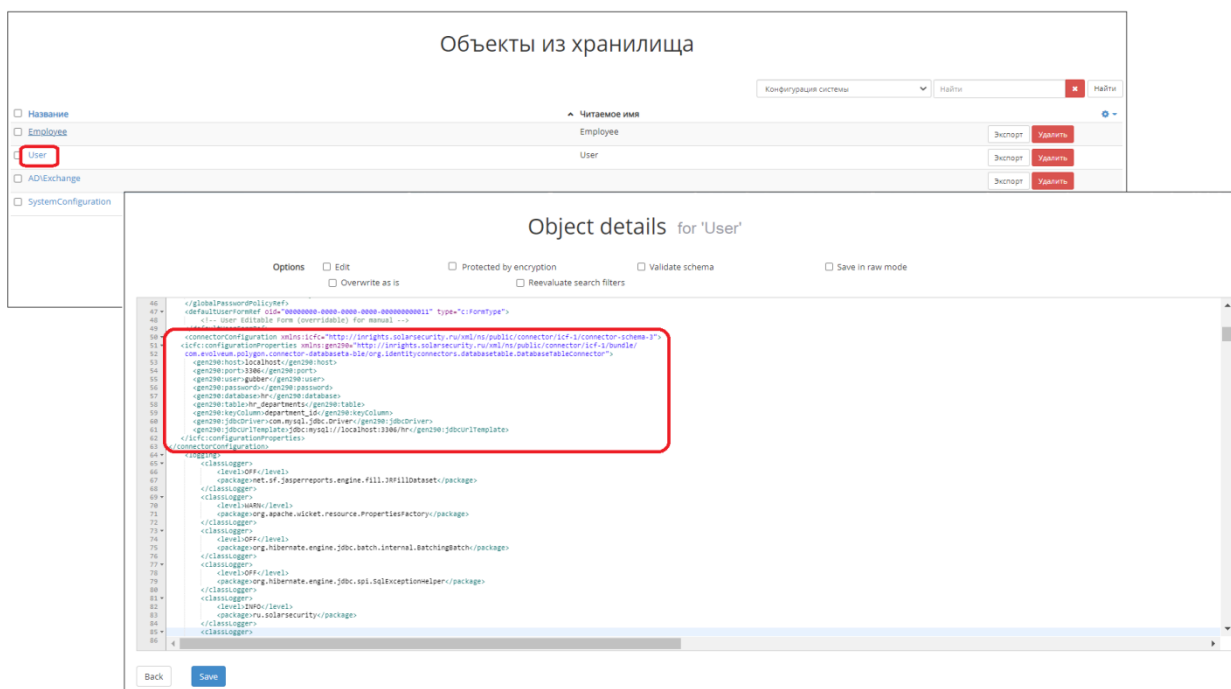



Рис. 68 – Настройка подключения к БД: редактирование XML-файла

4.1.1.2 Настройка подключения к AD

Для подключения ПС к **Active Directory** требуется создать ресурс. При создании ресурса в интерфейсе (**Информационные системы**>**Новая информационная система**, подробнее см. раздел 3.6.1) в подразделе **Новая информационная система**>**Конфигурация** необходимо задать следующие параметры:

- Directory Administrator's Account - имя пользователя для доступа к AD;
- Directory Administrator's Password - пароль для доступа к AD;
- Object Class for User Objects - класс объектов в AD, соответствующий типу объектов Пользователь (по умолчанию - User);
- Container - объект, который будет корневым объектом для поиска;
- Create Home Directory - параметр, указывающий, следует ли создавать родительский объект, если он отсутствует;

- Active Directory Domain Controller Hostname - хост, на котором располагается контроллер AD;
- Search Child Domains - параметр, указывающий, следует ли искать объекты в дочерних доменах;
- Domain Name - имя windows-домена;
- Sync Global Catalog Server - имя сервера глобального каталога. Значение параметра следует указывать только в том случае, если используется поиск по дочерним доменам;
- Sync Domain Controller - контроллер домена, используемый во время ActiveSync. Значение параметра следует указывать только в том случае, если не используется поиск по дочерним доменам.

The screenshot shows the 'Конфигурация' (Configuration) tab in the Exchange Admin Center. The 'AD' (Active Directory) configuration page is displayed, showing various settings for connecting to an Active Directory domain controller. The settings include:

- Directory Administrator's Account: administrator
- Version of Exchange (2007, 2010, or 2013):
- Directory Administrator's Password: (masked)
- URI of remote PowerShell used to manage Exchange (if applicable):
- Object Class for User Objects: User
- ProtectedMailbox enable: Не определено
- Container: OU=idm, DC=isim, DC=local
- Create Home Directory: Да
- Active Directory Domain Controller Hostname: isim
- Search Child Domains: Нет
- Domain Name: isim.local
- Sync Global Catalog Server: isim.local
- Sync Domain Controller: isim.local

Рис. 69 – Настройка подключения к Active Directory

Создать и настроить ресурс можно также с помощью загрузки (см. раздел 3.8.1) или редактирования (см. раздел 3.8.2.4) XML-файла, содержащего в секции `connectorConfiguration` необходимые конфигурационные параметры, например:

```
<connectorConfiguration
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/connector-schema-3">
  <icfc:resultsHandlerConfiguration>

<icfc:enableCaseInsensitiveFilter>true</icfc:enable
CaseInsensitiveFilter>
  </icfc:resultsHandlerConfiguration>
  <icfc:configurationProperties
xmlns:gen836="http://inrights.solarsecurity.ru/xml/
ns/public/connector/icf-1/bundle/

ActiveDirectory.Connector/Org.IdentityConnectors.Ac
tiveDirectory.ActiveDirectoryConnector">

<gen836:DirectoryAdminName>administrator</gen836:Di
rectoryAdminName>

<gen836:DirectoryAdminPassword></gen836:DirectoryAd
minPassword>
  <gen836:ObjectClass>User</gen836:ObjectClass>

<gen836:Container>OU=inrights,DC=isim,DC=local</gen
836:Container>

<gen836:CreateHomeDirectory>true</gen836:CreateHome
Directory>

<gen836:LDAPHostName>isim.local</gen836:LDAPHostNam
e>

<gen836:SearchChildDomains>false</gen836:SearchChil
dDomains>
  <gen836:DomainName>isim</gen836:DomainName>

<gen836:SyncGlobalCatalogServer>isim</gen836:SyncGl
obalCatalogServer>
```

```
<gen836:SyncDomainController>isim</gen836:SyncDomainController>
    </icfc:configurationProperties>
</connectorConfiguration>
```

4.1.1.3 Настройка подключения к текстовому файлу

Для подключения ПС к текстовому файлу (как источнику данных) требуется создать ресурс. При создании ресурса в интерфейсе **(Информационные системы>Новая информационная система,** подробнее см. раздел 3.6.1) в подразделе **Новая информационная система> Конфигурация** необходимо задать следующие параметры:

- Field delimiter - символ или набор символов, являющихся разделителями полей в файле;
- File path - путь к файлу-источнику данных;
- Encoding - кодировка, в которой сохранён файл;
- Name attribute - наименование атрибута объекта;
- Unique identifier attribute - уникальный идентификатор объекта.

The screenshot shows the 'Новая информационная система' (New Information System) configuration window. The 'Конфигурация' (Configuration) tab is selected, and the 'configurationProperties' section is expanded. The following settings are visible:

- Bind password:** Two empty text input fields.
- VLV ordering rule:** One empty text input field.
- Primary identifier attribute:** One empty text input field.
- Operational attributes:** One empty text input field with a green '+' button to the right.
- Object classes to synchronize:** One empty text input field with a green '+' button to the right.
- Bind DN:** One empty text input field.
- Base contexts to synchronize:** One empty text input field with a green '+' button to the right.
- Attributes to synchronize:** One empty text input field with a green '+' button to the right.
- Connection security:** One empty text input field.
- Referral strategy:** One empty text input field.
- Schema quirks mode:** A dropdown menu currently showing 'Не определено' (Not defined).
- Changelog block size:** One empty text input field.

Рис. 70 – Настройка подключения к текстовому файлу

Создать и настроить ресурс можно также с помощью загрузки (см. раздел 3.8.1) или редактирования (см. раздел 3.8.2.4) XML-файла, содержащего в секции **connectorConfiguration** необходимые конфигурационные параметры, например:

```
<connectorConfiguration
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/connector-schema-3">
  <icfc:configurationProperties
xmlns:gen375="http://inrights.solarsecurity.ru/xml/
ns/public/connector/icf-1/bundle/
  com.evolveum.polygon.connector-
csvfile/com.evolveum.polygon.csvfile.CSVFileConnect
or">
    <gen375:encoding>UTF-8</gen375:encoding>

<gen375:fieldDelimiter>;</gen375:fieldDelimiter>

<gen375:filePath>C:\Users\i.ivanov\work\projects\pr
oject\deploy\hr\hr.csv</gen375:filePath>

<gen375:nameAttribute>lastname</gen375:nameAttribut
e>

<gen375:uniqueAttribute>id</gen375:uniqueAttribute>
  </icfc:configurationProperties>
</connectorConfiguration>
```

4.1.2 Настройка объектов: подразделений, должностей, пользователей

Для настройки любого объекта необходимо в AGUI в разделе **Информационные системы > Новая информационная система** (см. раздел 3.6.1) создать и настроить ресурс, с помощью которого будут импортироваться данные из доверенного источника в информационное пространство ПС. При этом при задании параметров синхронизации в

подразделе **Новая информационная система** > **Синхронизация** следует:

1) В списке **Focus** указать соответствующий тип загружаемого объекта:

- **UserType** (пользователь);
- **OrgType** (организационная структура, подразделение);
- **RoleType** (роль, должность).

2) При необходимости в списке **Object Template** задать соответствующий шаблон объекта (описание и примеры шаблонов приведены в разделе 4.1.3).

Новая информационная система

Основные свойства > Конфигурация > Схема > Обработка связей > Возможности > Синхронизация

Синхронизация

Объекты синхронизации
Add Synchronization Object

--Select sync object to edit--

Name

Description

Kind

Intent

Object class

Focus

Enabled ☐ Reconcile ☐

Correlation

Confirmation Condition

Object Template

Opportunistic ☐ True | Unset | False

Reaction

Рис. 71 – Настройка объектов: задание параметров синхронизации

После создания ресурса и его первичной настройки необходима более точная настройка ресурса. Для этого следует отредактировать его XML-описание в разделе **Конфигурирование** > **Объекты в репозитории** (см. Рис. 72).

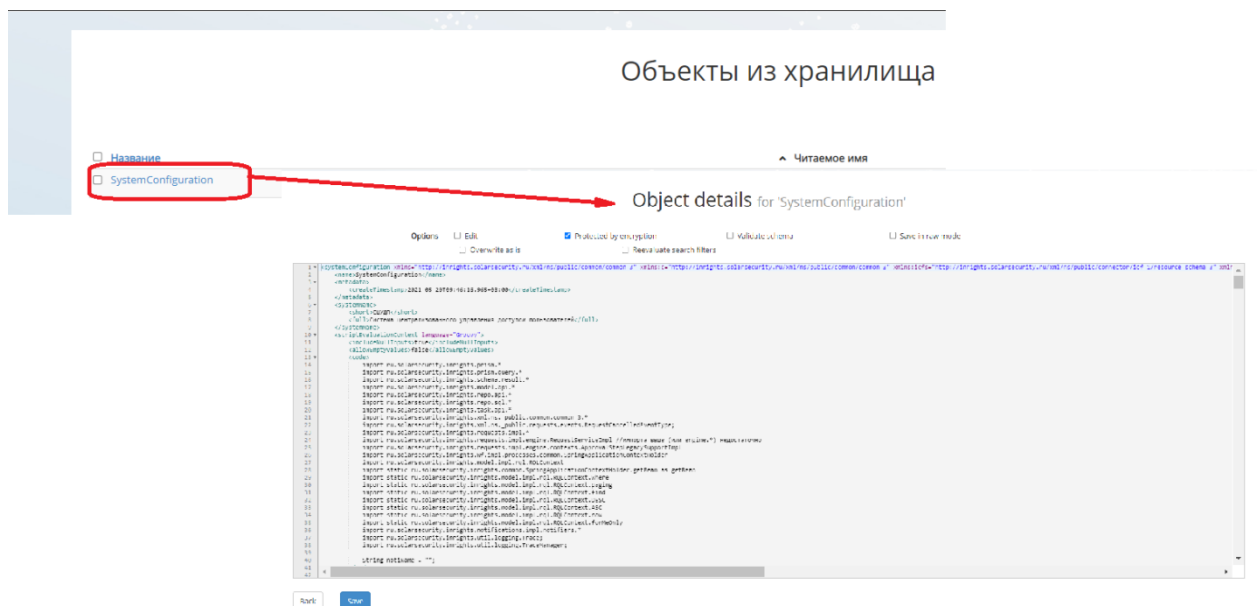


Рис. 72 – Настройка ресурса: редактирование XML-описания

В разделах 4.1.2.1, 4.1.2.2 и 4.1.2.3 приведены примеры XML-описаний объектов. Каждому шагу настройки объекта соответствует определенный блок в XML-описании объекта: блок настроек подключения коннектора (**connectorConfiguration**), блок настроек соответствия схемы и данных внутри системы (**schemaHandling**), блок настроек синхронизации (**synchronization**).

В свою очередь:

- блок **schemaHandling** содержит настройки атрибутов объекта (разделы attribute):
 - о **c:ref** - ссылка на источник данных (значение указано в разделе schema в xml-структуре документа);
 - о **displayName** - наименование соответствующего поля в AGUI;
 - о **inbound** - правило синхронизации атрибута из управляемой системы в ПС:
 - о **authoritative** - если задано true, то значение атрибута может как устанавливаться, так и удаляться. В противном случае операция удаления значения недоступна;

о **strength** - политика установки значения из управляемой системы:

- weak - указывает, что значение атрибута может быть задано только при пустом поле;
- normal - значение может быть задано, если оно не менялось в ПС;
- strong - значение задается всегда.

о **target** - поле, отображающееся при настройке объекта в AGUI, в которое вводится значение атрибута.

- **блок synchronization** содержит:

о блок **correlation**, в котором указываются правила сопоставления исходных данных и данных внутри ПС.

блоки **reaction**, в которых задается поведение системы в тех или иных случаях (описание см. в разделе 3.6.1 (пункт 12) и шаблон, на основании которого будет создаваться объект (примеры шаблонов см. в разделе 4.1.3).

4.1.2.1 Настройка организационной структуры и подразделений (OrgType)

В этом разделе приведены примеры настройки организационной структуры:

1) Пример настройки параметров подключения коннектора:

```
<connectorConfiguration
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/connector-schema-3">
    <icfc:configurationProperties
xmlns:gen659="http://inrights.solarsecurity.ru/xml/
ns/public/connector/icf-1/bundle
/com.evolveum.polygon.connector-
databasetable/org.identityconnectors.databasetable.
DatabaseTableConnector">
        <gen659:host>localhost</gen659:host>
```



```
<gen659:port>3306</gen659:port>
<gen659:user>gubber</gen659:user>
  <connectorConfiguration
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/
  connector-schema-3">
    <icfc:configurationProperties
xmlns:gen659="http://inrights.solarsecurity.ru/xml/
ns/public/connector/icf-1/bundle/
    com.evolveum.polygon.connector-
databasetable/org.identityconnectors.databasetable.
DatabaseTableConnector">
      <gen659:host>localhost</gen659:host>
      <gen659:port>3306</gen659:port>
      <gen659:user>gubber</gen659:user>
      <gen659:password>
        <t:encryptedData>
          <t:encryptionMethod>
<t:algorithm>http://www.w3.org/2001/04/xmlenc#aes12
8-cbc</t:algorithm>
          </t:encryptionMethod>
          <t:keyInfo>
<t:keyName>ARkSknaANjlIKmWwh8J2p6Nmqq0=</t:keyName>
          </t:keyInfo>
          <t:cipherData>
<t:cipherValue>VZZgLAyr904fb9h3MIATDEuP6vpNTzjcUeWb
7gU5Sow=</t:cipherValue>
          </t:cipherData>
        </t:encryptedData>
      </gen659:password>
      <gen659:database>hr</gen659:database>
      <gen659:table>hr_departments</gen659:table>
```

```
<gen659:keyColumn>department_id</gen659:keyColumn>

<gen659:jdbcDriver>com.mysql.jdbc.Driver</gen659:jdb
bcDriver>

<gen659:jdbcUrlTemplate>jdbc:mysql://localhost:3306
/hr</gen659:jdbcUrlTemplate>

    </icfc:configurationProperties>
</connectorConfiguration>
```

Примечание

Для задания пароля необходимо удалить код внутри тега gen659:password и указать новый пароль, при этом в начале и конце строки не должно быть пробелов:

```
<gen659:password>new_pass</gen659:password>
```

2) Пример настройки соответствия схемы и данных внутри системы:

```
<schemaHandling>
  <objectType>
    <kind>account</kind>
    <displayName>Подразделение</displayName>
    <default>true</default>
    <objectClass
xmlns:qn582="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
      qn582:AccountObjectClass</objectClass>
    <attribute>
      <c:ref
xmlns:qn782="http://inrights.solarsecurity.ru/xml/n
s/public/connector/icf-1/resource-schema-3">
        qn782:uid</c:ref>
      <displayName>Идентификатор</displayName>
      <exclusiveStrong>false</exclusiveStrong>
      <tolerant>true</tolerant>
    <inbound>
```

```
<name>Org identifier</name>
<authoritative>true</authoritative>
<exclusive>false</exclusive>
<strength>normal</strength>
<target>
    <c:path>$focus/identifier</c:path>
</target>
</inbound>
</attribute>
...
</objectType>
</schemaHandling>
```

3) Пример настройки синхронизации:

```
<synchronization>
    <objectSynchronization>
        <name>Department sync</name>

<objectClass>ri:AccountObjectClass</objectClass>
    <kind>account</kind>
    <focusType>c:OrgType</focusType>
    <enabled>true</enabled>
    <correlation>
        <q:equal>
            <q:path>identifier</q:path>
            <expression>

<c:path>$account/attributes/uid</c:path>
            </expression>
        </q:equal>
    </correlation>
    <reconcile>true</reconcile>
    <opportunistic>true</opportunistic>
    <reaction>
```

```
<name>Создать подразделение</name>
<situation>unmatched</situation>
<synchronize>true</synchronize>
<reconcile>true</reconcile>
<objectTemplateRef oid="127298b1-6795-
412e-8a46-ff5c6364cc99" type="ObjectTemplateType"/>
<action>
  <name>Создать подразделение</name>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#addFocus</handlerUri>
  <order>before</order>
</action>
</reaction>
<reaction>
  <name>Синхронизировать изменения</name>
  <situation>linked</situation>
  <synchronize>true</synchronize>
  <reconcile>false</reconcile>
  <objectTemplateRef oid="127298b1-6795-
412e-8a46-ff5c6364cc99" type="ObjectTemplateType"/>
</reaction>
<reaction>
  <name>Удалить</name>
  <situation>deleted</situation>
  <reconcile>false</reconcile>
  <action>
    <name>Удалить</name>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#deleteFocus</handlerUri>
  <order>before</order>
  </action>
</reaction>
```

```
<reaction>
  <name>Link</name>
  <situation>unlinked</situation>
  <reconcile>false</reconcile>
  <action>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#link</handlerUri>
  <order>before</order>
</action>
</reaction>
</objectSynchronization>
</synchronization>
```

4.1.2.2 Настройка должностей (**RoleType**), задание связи с подразделениями

В этом разделе приведен пример настройки должностей:

```
<resource
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

oid="0b579196-f6f5-4e76-82da-e10c8437b683"
```

```
version="109">
<name>Должности из HR</name>
<metadata>
  <createTimestamp>2016-06-
01T11:46:11.709+03:00</createTimestamp>
  <creatorRef oid="00000000-0000-0000-0000-
000000000002" type="c:UserType"><!-- administrator
--></creatorRef>

<createChannel>http://inrights.solarsecurity.ru/xml
/ns/public/model/channels-
3#objectImport</createChannel>
  <modifyTimestamp>2016-08-
10T14:35:19.681+03:00</modifyTimestamp>
  <modifierRef
xmlns:tns="http://inrights.solarsecurity.ru/xml/ns/
public/common/common-3"
    oid="00000000-0000-0000-0000-
000000000002"
    type="tns:UserType"><!--
administrator --></modifierRef>

<modifyChannel>http://inrights.solarsecurity.ru/xml
/ns/public/gui/channels-3#user</modifyChannel>
</metadata>
<operationalState>

<lastAvailabilityStatus>up</lastAvailabilityStatus>
</operationalState>
  <connectorRef oid="476ae395-ca9d-464d-9c27-
a8b86ddb8d2" type="c:ConnectorType">
<!-- ICF
org.identityconnectors.databasetable.DatabaseTableC
onnecter v1.4.0.49 -->
  <filter>
    <q:equal>
      <q:path>c:connectorType</q:path>
```

```
<q:value>org.identityconnectors.databasetable.DatabaseTableConnector</q:value>
    </q:equal>
</filter>
</connectorRef>
<connectorConfiguration
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/connector-schema-3">
    <icfc:configurationProperties
xmlns:gen325="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/bundle/
    com.evolveum.polygon.connector-databasetable/org.identityconnectors.databasetable.DatabaseTableConnector">
        <gen325:host>localhost</gen325:host>
        <gen325:port>3306</gen325:port>
        <gen325:user>gubber</gen325:user>
        <gen325:password>
            <t:encryptedData>
                <t:encryptionMethod>
<t:algorithm>http://www.w3.org/2001/04/xmlenc#aes128-cbc</t:algorithm>
                </t:encryptionMethod>
                <t:keyInfo>
<t:keyName>ARkSknaANjlIKmWwh8J2p6Nmqq0=</t:keyName>
                </t:keyInfo>
                <t:cipherData>
<t:cipherValue>XlJNeHgTd0WV+exWRv/4WVN+acW/mJr0/gRqeTkJ8j4=</t:cipherValue>
                </t:cipherData>
            </t:encryptedData>
        </gen325:password>
```

```

    <gen325:database>hr</gen325:database>
    <gen325:table>hr_positions</gen325:table>

<gen325:keyColumn>position_id</gen325:keyColumn>

<gen325:jdbcDriver>com.mysql.jdbc.Driver</gen325:jdb
bcDriver>

<gen325:jdbcUrlTemplate>jdbc:mysql://localhost:3306
/hr</gen325:jdbcUrlTemplate>
    </icfc:configurationProperties>
</connectorConfiguration>
<schema>
    < cachingMetadata>
        < retrievalTimestamp
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="xsd:dateTime">2015-05-
07T14:46:37.201+03:00</retrievalTimestamp>
        < serialNumber>ab3fa04f4f1892c0-
765aeaf0141df656</serialNumber>
    </ cachingMetadata>
    < definition>
        < xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:a="http://prism.solarsecurity.ru/xml/ns/publi
c/annotation-3"

xmlns:tns="http://inrights.solarsecurity.ru/xml/ns/
public/resource/instance-3"

xmlns:ra="http://inrights.solarsecurity.ru/xml/ns/p
ublic/resource/annotation-3"

        elementFormDefault="qualified"

```



```

targetNamespace="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3">

    <xsd:import
namespace="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"/>

    <xsd:import
namespace="http://prism.solarsecurity.ru/xml/ns/public/annotation-3"/>

    <xsd:import
namespace="http://inrights.solarsecurity.ru/xml/ns/public/resource/annotation-3"/>

    <xsd:complexType
name="AccountObjectClass">
        <xsd:annotation>
            <xsd:appinfo>
                <ra:resourceObject/>

<ra:identifier>icfs:uid</ra:identifier>

<ra:secondaryIdentifier>icfs:name</ra:secondaryIdentifier>

<ra:displayNameAttribute>icfs:name</ra:displayNameAttribute>

<ra:namingAttribute>icfs:name</ra:namingAttribute>

<ra:nativeObjectClass>__ACCOUNT__</ra:nativeObjectClass>

                <ra:kind>account</ra:kind>
                <ra:default>true</ra:default>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element minOccurs="0"
ref="icfs:uid">
                <xsd:annotation>

```

```

                                <xsd:appinfo>
                                    <a:displayName>ICF
UID</a:displayName>

                                </xsd:appinfo>
                                </xsd:annotation>
                                </xsd:element>
                                <xsd:element name="position_name"
type="xsd:string"/>
                                <xsd:element ref="icfs:name">
                                    <xsd:annotation>
                                        <xsd:appinfo>
                                            <a:displayName>ICF
NAME</a:displayName>

                                        </xsd:appinfo>
                                        </xsd:annotation>
                                        </xsd:element>
                                        <xsd:element name="id"
type="xsd:int"/>
                                        <xsd:element minOccurs="0"
name="department_id" type="xsd:string"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                                </xsd:schema>
                            </definition>
                        </schema>
                    <schemaHandling>
                        <objectType>
                            <kind>account</kind>
                            <displayName>Должность</displayName>
                            <default>true</default>
                            <objectClass
xmlns:qn582="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">

```

```
qn582:AccountObjectClass</objectClass>
<attribute>
  <c:ref
xmlns:qn782="http://inrights.solarsecurity.ru/xml/n
s/public/connector/icf-1/resource-schema-3">
    qn782:uid</c:ref>

<displayName>Идентификатор</displayName>

<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <inbound>
    <name>Org identifier</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
    <strength>normal</strength>
    <target>
      <c:path>$focus/name</c:path>
    </target>
  </inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn686="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn686:position_name</c:ref>
    <displayName>Наименование</displayName>

<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <fetchStrategy>implicit</fetchStrategy>
  <inbound>
    <name>Org name</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
```

```
<strength>normal</strength>
<target>

<c:path>$focus/displayName</c:path>
  </target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn849="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn849:department_id</c:ref>
    <displayName>Родительское
подразделение</displayName>

<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <inbound>
    <name>Org mapping</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
    <strength>normal</strength>
    <target>

<c:path>$focus/extension/parentOrgIdentifier</c:pat
h>
  </target>
</inbound>
</attribute>
</objectType>
</schemaHandling>
<capabilities>
  < cachingMetadata>
    < retrievalTimestamp
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="xsd:dateTime">2015-05-
07T14:46:37.201+03:00</retrievalTimestamp>
    <serialNumber>84472c93bc43a964-
13d40eabac8d3b76</serialNumber>
    </cachingMetadata>
    <native
xmlns:cap="http://inrights.solarsecurity.ru/xml/ns/
public/resource/capabilities-3">
        <cap:addRemoveAttributeValues/>
        <cap:liveSync/>
        <cap:testConnection/>
        <cap:create/>
        <cap:read/>
        <cap:update/>
        <cap:delete/>
        <cap:script>
            <cap:host>
                <cap:type>connector</cap:type>
            </cap:host>
        </cap:script>
    </native>
</capabilities>
<synchronization>
    <objectSynchronization>
        <name>Position sync</name>

<objectClass>ri:AccountObjectClass</objectClass>
    <kind>account</kind>
    <enabled>true</enabled>
    <correlation>
        <q:equal>

```

```
<q:path>name</q:path>
<expression>
  <c:path>$focus/name</c:path>
</expression>
</q:equal>
</correlation>
<reconcile>true</reconcile>
<opportunistic>true</opportunistic>
<reaction>
  <name>Создать подразделение</name>
  <situation>unmatched</situation>
  <synchronize>true</synchronize>
  <reconcile>true</reconcile>
  <objectTemplateRef oid="127298b1-6795-
412e-8a46-ff5c6364cc44" type="ObjectTemplateType"/>
  <action>
    <name>Создать подразделение</name>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#addFocus</handlerUri>
    <order>before</order>
  </action>
</reaction>
<reaction>
  <name>Синхронизировать изменения</name>
  <situation>linked</situation>
  <synchronize>true</synchronize>
  <reconcile>false</reconcile>
  <objectTemplateRef oid="127298b1-6795-
412e-8a46-ff5c6364cc44" type="ObjectTemplateType"/>
</reaction>
<reaction>
  <name>Удалить</name>
  <situation>deleted</situation>
```

```

<reconcile>false</reconcile>
<action>
  <name>Удалить</name>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#deleteFocus</handlerUri>
  <order>before</order>
</action>
</reaction>
<reaction>
  <name>Link</name>
  <situation>unlinked</situation>
  <reconcile>false</reconcile>
  <action>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#link</handlerUri>
  <order>before</order>
  </action>
</reaction>
</objectSynchronization>
</synchronization>
</resource>

```

4.1.2.3 Настройка профилей пользователей (UserType), задание связи с должностями и подразделениями

В этом разделе приведен пример настройки профиля пользователя:

```

<resource
xmlns="http://inrights.solarsecurity.ru/xml/ns/publ
ic/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/publi
c/query-3"

```

```
xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"
```

```
xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"
```

```
xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"
```

```
xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"
```

```
oid="37c21384-6996-45eb-8b27-3820e35c5004"
```

```
version="303">
```

```
<name>Пользователи из HR (расшир)</name>
```

```
<operationalState>
```

```
<lastAvailabilityStatus>up</lastAvailabilityStatus>
```

```
</operationalState>
```

```
<connectorRef oid="476ae395-ca9d-464d-9c27-a8b86ddb8d2" type="c:ConnectorType">
```

```
<!-- ICF
```

```
org.identityconnectors.databasetable.DatabaseTableConnector v1.4.0.49 -->
```

```
<filter>
```

```
<q:equal>
```

```
<q:path>c:connectorType</q:path>
```

```
<q:value>org.identityconnectors.databasetable.DatabaseTableConnector</q:value>
```

```
</q:equal>
```

```
</filter>
```

```
</connectorRef>
```

```
<connectorConfiguration
```

```
xmlns:icfc="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/connector-schema-3">
```



```
<icfc:configurationProperties
xmlns:gen311="http://inrights.solarsecurity.ru/xml/
ns/public/connector/icf-1/bundle/

    com.evolveum.polygon.connector-
databasetable/org.identityconnectors.databasetable.
DatabaseTableConnector">

    <gen311:host>localhost</gen311:host>
    <gen311:port>3306</gen311:port>
    <gen311:user>gubber</gen311:user>
    <gen311:password>
        <t:encryptedData>
            <t:encryptionMethod>

<t:algorithm>http://www.w3.org/2001/04/xmlenc#aes12
8-cbc</t:algorithm>
            </t:encryptionMethod>
            <t:keyInfo>

<t:keyName>ARkSknaANjlIKmWwh8J2p6Nmqq0=</t:keyName>
            </t:keyInfo>
            <t:cipherData>

<t:cipherValue>eloT75Crio4hNHQUK0mrCDAz2511YKCwxQrr
+QAhaLU=</t:cipherValue>
            </t:cipherData>
        </t:encryptedData>
    </gen311:password>
    <gen311:database>hr</gen311:database>

<gen311:table>isim_employees_view</gen311:table>

<gen311:keyColumn>EMP_NUMBER</gen311:keyColumn>

<gen311:jdbcDriver>com.mysql.jdbc.Driver</gen311:jd
bcDriver>
```

```
<gen311:jdbcUrlTemplate>jdbc:mysql://localhost:3306/hr</gen311:jdbcUrlTemplate>

<gen311:changeLogColumn>EMP_MODIFY</gen311:changeLogColumn>

    </icfc:configurationProperties>
</connectorConfiguration>
<schema>
    < cachingMetadata>
        < retrievalTimestamp
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:type="xsd:dateTime">2015-05-15T14:06:27.190+03:00</retrievalTimestamp>
        < serialNumber>b93b30504c8b1119-c2a016d58b2ea599</serialNumber>
    </ cachingMetadata>
    < definition>
        < xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:a="http://prism.solarsecurity.ru/xml/ns/public/annotation-3"

xmlns:tns="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

xmlns:ra="http://inrights.solarsecurity.ru/xml/ns/public/resource/annotation-3"

        elementFormDefault="qualified"

        targetNamespace="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3">
```

```

        <xsd:import
namespace="http://inrights.solarsecurity.ru/xml/ns/
public/connector/icf-1/resource-schema-3"/>

        <xsd:import
namespace="http://prism.solarsecurity.ru/xml/ns/pub
lic/annotation-3"/>

        <xsd:import
namespace="http://inrights.solarsecurity.ru/xml/ns/
public/resource/annotation-3"/>

        <xsd:complexType
name="AccountObjectClass">
            <xsd:annotation>
                <xsd:appinfo>
                    <ra:resourceObject/>

<ra:identifier>icfs:uid</ra:identifier>

<ra:secondaryIdentifier>icfs:name</ra:secondaryIden
tifier>

<ra:displayNameAttribute>icfs:name</ra:displayNameA
ttribute>

<ra:namingAttribute>icfs:name</ra:namingAttribute>

<ra:nativeObjectClass>__ACCOUNT__</ra:nativeObjectC
lass>

                <ra:kind>account</ra:kind>
                <ra:default>true</ra:default>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element minOccurs="0"
ref="icfs:uid">
                <xsd:annotation>
                    <xsd:appinfo>

```

```

                                <a:displayName>ICF
UID</a:displayName>

<a:access>read</a:access>

                                </xsd:appinfo>
                                </xsd:annotation>
                                </xsd:element>
                                <xsd:element
name="EMP_MIDDLE_NAME" type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="EMP_STATUS" type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="EMP_DATEOFBIRTH" type="xsd:string"/>
                                <xsd:element name="EMP_STREET"
type="xsd:string"/>
                                <xsd:element
name="EMP_DESCRIPTION" type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="EMP_KEY" type="xsd:string"/>
                                <xsd:element name="DEP_KEY"
type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="EMP_WORKEND" type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="EMP_MANAGER_KEY" type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="EMP_WORKSTART" type="xsd:string"/>
                                <xsd:element
name="EMP_MOBILE_PHONE" type="xsd:string"/>
                                <xsd:element minOccurs="0"
name="DEP_NAME" type="xsd:string"/>
                                <xsd:element name="POS_KEY"
type="xsd:string"/>
                                <xsd:element ref="icfs:name">
                                <xsd:annotation>
                                <xsd:appinfo>

```

```

NAME</a:displayName>
        </xsd:appinfo>
    </xsd:annotation>
</xsd:element>
    <xsd:element name="EMP_LAST_NAME"
type="xsd:string"/>
    <xsd:element name="EMP_MODIFY"
type="xsd:string"/>
    <xsd:element
name="EMP_FIRST_NAME" type="xsd:string"/>
    <xsd:element name="EMP_TELEPHONE"
type="xsd:string"/>
    <xsd:element minOccurs="0"
name="EMP_MANAGER_NAME" type="xsd:string"/>
    <xsd:element minOccurs="0"
name="POS_TITLE_NAME" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
</definition>
</schema>
<schemaHandling>
    <objectType>
        <kind>account</kind>
        <displayName>Пользователь</displayName>
        <default>true</default>
        <objectClass
xmlns:qn761="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3">
            qn761:AccountObjectClass</objectClass>
        <attribute>
            <c:ref
xmlns:schemaRef="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3">
                schemaRef:uid</c:ref>
            </attribute>
        </objectClass>
    </objectType>
</schemaHandling>

```

```
<displayName>Идентификатор</displayName>

<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <inbound>
    <name>Идентификатор</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
    <strength>normal</strength>
    <target>

<c:path>$focus/employeeNumber</c:path>
  </target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn936="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn936:EMP_LAST_NAME</c:ref>
    <displayName>Фамилия</displayName>

<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <inbound>
    <name>Фамилия</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
    <strength>normal</strength>
    <target>

<c:path>$focus/familyName</c:path>
  </target>
</inbound>
```

```
</attribute>
<attribute>
  <c:ref
xmlns:qn969="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn969:EMP_MIDDLE_NAME</c:ref>
    <displayName>Отчество</displayName>

<exclusiveStrong>false</exclusiveStrong>
<tolerant>true</tolerant>
<inbound>
  <name>Отчество</name>
  <authoritative>true</authoritative>
  <exclusive>false</exclusive>
  <strength>normal</strength>
  <target>

<c:path>$focus/additionalName</c:path>
  </target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn969="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn969:EMP_MANAGER_KEY</c:ref>
    <displayName>Руководитель</displayName>

<exclusiveStrong>false</exclusiveStrong>
<tolerant>true</tolerant>
<inbound>
  <name>Руководитель</name>
  <authoritative>true</authoritative>
  <exclusive>false</exclusive>
  <strength>normal</strength>
```

```
<target>
  <c:path>$focus/managerId</c:path>
</target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn652="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn652:EMP_FIRST_NAME</c:ref>
    <displayName>Имя</displayName>

<exclusiveStrong>false</exclusiveStrong>
    <tolerant>true</tolerant>
    <inbound>
      <name>Имя</name>
      <authoritative>true</authoritative>
      <exclusive>false</exclusive>
      <strength>normal</strength>
      <target>
        <c:path>$focus/givenName</c:path>
      </target>
    </inbound>
  </attribute>
  <attribute>
    <c:ref
xmlns:qn652="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
      qn652:DEP_KEY</c:ref>
      <displayName>ID
подразделения</displayName>

<exclusiveStrong>false</exclusiveStrong>
      <tolerant>true</tolerant>
      <inbound>
```



```
<name>ID подразделения</name>
<authoritative>true</authoritative>
<exclusive>false</exclusive>
<strength>normal</strength>
<target>

<c:path>$focus/organizationalUnit</c:path>
  </target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn788="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn788:POS_TITLE_NAME</c:ref>

<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <inbound>
    <name>Должность</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
    <strength>normal</strength>
    <target>
      <c:path>$focus/title</c:path>
    </target>
  </inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn788="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn788:EMP_STATUS</c:ref>

<exclusiveStrong>false</exclusiveStrong>
```

```
<tolerant>true</tolerant>
<inbound>
  <name>Cтaтyс</name>
  <authoritative>true</authoritative>
  <exclusive>false</exclusive>
  <strength>normal</strength>
  <expression>
    <script>
      <code>
        import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.HrStatusType;

        if(basic.isEmpty(input))
return null;

if(input.toString().equals('EMP')) return
HrStatusType.ACTIVE;

        else
if(input.toString().equals('REJ')) return
HrStatusType.FIRED;

      </code>
    </script>
  </expression>
  <target>
    <c:path>$focus/hrStatus</c:path>
  </target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn788="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn788:EMP_WORKSTART</c:ref>
</exclusiveStrong>false</exclusiveStrong>
<tolerant>true</tolerant>
```

```
<inbound>
  <name>Дата приема на работу</name>
  <authoritative>true</authoritative>
  <exclusive>false</exclusive>
  <strength>normal</strength>
  <target>
    <c:path>$focus/workstart</c:path>
  </target>
</inbound>
</attribute>
<attribute>
  <c:ref
xmlns:qn788="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">
    qn788:EMP_WORKEND</c:ref>
<exclusiveStrong>false</exclusiveStrong>
  <tolerant>true</tolerant>
  <inbound>
    <name>Дата увольнения</name>
    <authoritative>true</authoritative>
    <exclusive>false</exclusive>
    <strength>normal</strength>
    <target>
      <c:path>$focus/workend</c:path>
    </target>
  </inbound>
</attribute>
</objectType>
</schemaHandling>
<capabilities>
  <achingMetadata>
    <retrievalTimestamp
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="xsd:dateTime">2015-05-
15T14:06:27.193+03:00</retrievalTimestamp>

    <serialNumber>13c84d3b369dc376-
4189f72e2b252f64</serialNumber>

    </cachingMetadata>

    <native
xmlns:cap="http://inrights.solarsecurity.ru/xml/ns/
public/resource/capabilities-3">

        <cap:addRemoveAttributeValues/>
        <cap:liveSync/>
        <cap:testConnection/>
        <cap:create/>
        <cap:read/>
        <cap:update/>
        <cap:delete/>
        <cap:script>
            <cap:host>
                <cap:type>connector</cap:type>
            </cap:host>
        </cap:script>
    </native>
</capabilities>
<synchronization>
    <objectSynchronization>
        <name>Sync account</name>
        <objectClass
xmlns:qn911="http://inrights.solarsecurity.ru/xml/n
s/public/resource/instance-3">

qn911:AccountObjectClass</objectClass>

        <kind>account</kind>

        <focusType>UserType</focusType>
```

```
<enabled>true</enabled>
<correlation>
  <q:description>correlate by employee
id</q:description>
  <q:equal>
    <q:path>c:employeeNumber</q:path>
    <expression>
<c:path>$shadow/attributes/icfs:uid</c:path>
    </expression>
  </q:equal>
</correlation>
<reconcile>true</reconcile>
<opportunistic>false</opportunistic>
<reaction>
  <name>Add user</name>
  <situation>unmatched</situation>
  <synchronize>true</synchronize>
  <reconcile>false</reconcile>
  <objectTemplateRef oid="246f7b2e-20cd-
4377-8848-a1ebb5c0bcc6" type="ObjectTemplateType"/>
  <action>
    <name>Add user</name>

<handlerUri>http://inrights.solarsecurity.ru/xml/ns
/public/model/action-3#addFocus</handlerUri>
  <order>before</order>
</action>
</reaction>
<reaction>
  <name>Synchronize</name>
  <situation>linked</situation>
  <synchronize>true</synchronize>
  <reconcile>false</reconcile>
```

```

</reaction>
<reaction>
  <name>Link</name>
  <situation>unlinked</situation>
  <synchronize>true</synchronize>
  <reconcile>false</reconcile>
</reaction>
</objectSynchronization>
</synchronization>
</resource>

```

4.1.3 Задание шаблонов объектов

4.1.3.1 Описание шаблонов объектов

Шаблоны объектов описывают стандартные правила создания объектов. В шаблоне в блоках задаются параметры объектов (mapping-параметры). Каждый блок состоит из следующих частей:

- один или несколько блоков **source**. Этот блок описывает исходные данные для вычислений и состоит из двух секций (name - имя параметра, которое будет использоваться для вычислений; c:path - источник данных);
- один блок **expression**, где указывается выражение, которое должно возвращать значение, вычисляемое на основе данных, описанных в source;
- один блок **target**, где указывается, куда записывается вычисленное значение;
- один или ни одного блока **condition**. Этот блок определяет условие, при котором значение вычисляется и сохраняется.

Шаблон объекта можно создать в AGUI в разделе **Конфигурирование > Загрузить объект** (см. раздел 3.8.1).

4.1.3.2 Примеры шаблонов объектов

В разделе приведены примеры шаблонов объектов:

1) Шаблон роли (RoleType), обеспечивающий привязку должности к подразделению:

```
<objectTemplate
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

        oid="127298b1-6795-412e-8a46-ff5c6364cc44"

        version="92">
    <name>Шаблон роли HR</name>
    <metadata>
        <createTimestamp>2016-05-31T13:24:07.957+03:00</createTimestamp>
        <creatorRef oid="00000000-0000-0000-0000-000000000002" type="c:UserType"><!-- administrator --></creatorRef>

    <createChannel>http://inrights.solarsecurity.ru/xml/ns/public/model/channels-3#objectImport</createChannel>
    </metadata>
    <mapping>
```

```

<source>
    <name>parentOrgIdentifier</name>

<c:path>$focus/extension/parentOrgIdentifier</c:pat
h>

</source>
<source>
    <name>positionName</name>
    <c:path>$user/name</c:path>
</source>
<expression>
    <assignmentTargetSearch>
        <targetType>c:OrgType</targetType>
        <filter>
            <q:equal>
                <q:path>c:identifier</q:path>
                <expression>

<q:path>$parentOrgIdentifier</q:path>
                </expression>
            </q:equal>
        </filter>
    </assignmentTargetSearch>
</expression>
<target>
    <c:path>assignment</c:path>
</target>
</mapping>
</objectTemplate>

```

2) Шаблон организационной структуры (OrgType):

```

<objectTemplate
xmlns="http://inrights.solarsecurity.ru/xml/ns/publ
ic/common/common-3"

```


`xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"`

`xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"`

`xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"`

`xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"`

`xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"`

`oid="127298b1-6795-412e-8a46-ff5c6364cc99"`

`version="214">`

`<name>Шаблон орг. структуры</name>`

`<mapping>`

`<source>`

`<name>parentOrgIdentifier</name>`

`<c:path>$user/extension/parentOrgIdentifier</c:path>`

`</source>`

`<expression>`

`<assignmentTargetSearch>`

`<targetType>c:OrgType</targetType>`

`<filter>`

`<q:equal>`

`<q:path>c:identifier</q:path>`

`<expression>`

`<q:path>$parentOrgIdentifier</q:path>`

`</expression>`

`</q:equal>`

```

</filter>
<createOnDemand>true</createOnDemand>
<populateObject>
  <populateItem>
    <expression>

<c:path>$parentOrgIdentifier</c:path>
    </expression>
    <target>
      <c:path>identifier</c:path>
    </target>
  </populateItem>
  <populateItem>
    <expression>

<c:path>$parentOrgIdentifier</c:path>
    </expression>
    <target>
      <c:path>name</c:path>
    </target>
  </populateItem>
</populateObject>
</assignmentTargetSearch>
</expression>
<target>
  <c:path>assignment</c:path>
</target>
</mapping>
</objectTemplate>

```

В данном шаблоне настраивается построение иерархии организационной структуры. Источник элементов иерархии указывается в параметре **extension/parentOrgIdentifier**. По значению этого параметра выполняется поиск объектов типа **OrgType**. При

отсутствии соответствующего объекта создаётся новый объект. После чего устанавливается связь с родительским объектом.

3) Шаблон профиля пользователя (UserType):

```
<objectTemplate
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

  oid="246f7b2e-20cd-4377-8848-a1ebb5c0bcc6"
  version="266">
    <name>Стандартный пользователь</name>
    <description>Формирует имя для входа на основе
      ФИО пользователя.
      Генерирует пароль (при импорте из
      HR). Заполняет полное имя.
      Привязывает к подразделению (если
      заполнен organizationalUnit). Assignment роли
      Сотрудник
    </description>
    <metadata>
      <createTimestamp>2015-05-
      28T14:21:21.856+03:00</createTimestamp>
```

```

        <creatorRef oid="00000000-0000-0000-0000-
0000000000002" type="c:UserType"><!-- administrator
--></creatorRef>

<createChannel>http://inrights.solarsecurity.ru/xml
/ns/public/model/channels-
3#objectImport</createChannel>

    </metadata>
    <iteration>
        <maxIterations>9999</maxIterations>
    </iteration>
    <mapping> <!-- Правило, устанавливающее значение
familyName + ' ' + givenName + ' ' + additionalName
в поле fullName,
                при условии что ни одно из полей не
является пустым-->
        <source> <!-- поле, используемое в выражении
-->
            <name>familyName</name> <!-- Название
поля, используемое в выражении -->
            <c:path>$user/familyName</c:path> <!--
поле откуда брать значение -->
        </source>
        <source>
            <name>givenName</name>
            <c:path>$user/givenName</c:path>
        </source>
        <source>
            <name>additionalName</name>
            <c:path>$user/additionalName</c:path>
        </source>
        <expression> <!-- выражение, которое
формирует финальное значение -->
            <script>
                <code>familyName + ' ' + givenName + '
' + additionalName</code>

```

```
</script>
</expression>
<target> <!-- поле, куда записывается
значение -->
    <c:path>$user/fullName</c:path>
</target>
<condition> <!-- условие при котором
вычислять и записывать значение -->
    <script>
        <code>givenName != null & &
familyName != null & & additionalName !=
null</code>
    </script>
</condition>
</mapping>
<mapping> <!-- Назначение роли пользователю -->
    <name>Сотрудник assignment</name>
    <authoritative>true</authoritative>
    <expression>
        <assignmentTargetSearch>
            <targetType>c:RoleType</targetType>
            <oid>7bd5a0fe-90eb-40ce-b311-
b8f95132bd3e</oid>
        </assignmentTargetSearch>
    </expression>
    <target>
        <c:path>assignment</c:path>
    </target>
</mapping>
<mapping> <!-- Установка руководителя для
пользователя-->
    <strength>weak</strength>
    <source>
        <name>managerId</name>
```

```
<c:path>$user/managerId</c:path>
</source>
<expression>
  <script>
    <code>
      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.ObjectReferenceType;

      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.UserType;

      import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

      import
ru.solarsecurity.inrights.requests.impl.*;

      import
ru.solarsecurity.inrights.prism.query.ObjectQuery;

      import java.lang.Exception;

      import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;

      import
ru.solarsecurity.inrights.prism.PrismObject;

      import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;

      import
ru.solarsecurity.inrights.prism.query.EqualFilter;

      import
ru.solarsecurity.inrights.prism.PrismContext;

      def modelService =
SpringApplicationContextHolder.getBean (ModelControl
ler.class);
```

```
def prismContext =
SpringApplicationContextHolder.getBean(PrismContext
.class);

def managerOid = null;;

OperationResult result = new
OperationResult("get_object_by_employee_number");
try {
    ObjectQuery query = new
ObjectQuery();

query.setFilter(EqualFilter.createEqual
(UserType.F_EMPLOYEE_NUMBER,
UserType.class, prismContext,
managerId.toString()));

    managerOid =
        modelService.searchObjects(
            UserType.class, query, null,
null, result).get(0).asObjectable().getOid();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    if(managerOid == null) return null;
    ObjectReferenceType managerRefer =
new ObjectReferenceType();
    managerRefer.setOid(managerOid);

managerRefer.setType(UserType.COMPLEX_TYPE);
    managerRefer;
</code>
</script>
</expression>
<target>
    <c:path>$user/managerRef</c:path>
</target>
<condition>
    <script>
```

```
<code>managerId != null</code>
</script>
</condition>
</mapping>
<mapping> <!-- Политика генерации пароля-->
  <source>
    <name>password</name>

<c:path>$user/credentials/password/value</c:path>
    </source>
    <expression>
      <generate>
        <valuePolicyRef oid="00000000-0000-
0000-0000-000000000003"/>
      </generate>
    </expression>
    <target>

<c:path>$user/credentials/password/value</c:path>
    </target>
    <condition>
      <script>
        <code>
          return password == null ||
password.isEmpty();
        </code>
      </script>
    </condition>
  </mapping>
  <mapping> <!-- установка подразделения, при
необходимости,
          создаётся запись о новом
подразделении. Условие, сотрудник не уволен.-->
    <source>
```



```

    <name>orgIdentifier</name>
    <c:path>$user/organizationalUnit</c:path>
</source>
<source>
    <name>hrStatus</name>
    <c:path>$user/hrStatus</c:path>
</source>
<expression>
    <assignmentTargetSearch>
        <targetType>c:OrgType</targetType>
        <filter>
            <q:equal>
                <q:path>c:identifier</q:path>
                <expression>
<q:path>$orgIdentifier</q:path>
                    </expression>
                </q:equal>
            </filter>
            <createOnDemand>true</createOnDemand>
            <populateObject>
                <populateItem>
                    <expression>
<c:path>$orgIdentifier</c:path>
                        </expression>
                    <target>
                        <c:path>identifier</c:path>
                    </target>
                </populateItem>
                <populateItem>
                    <expression>
<c:path>$orgIdentifier</c:path>

```

```
</expression>
<target>
  <c:path>name</c:path>
</target>
</populateItem>
</populateObject>
</assignmentTargetSearch>
</expression>
<target>
  <c:path>assignment</c:path>
</target>
<condition>
  <script>
    <code>
      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.HrStatusType;

      return hrStatus !=
HrStatusType.FIRED && orgIdentifier !=
null;
    </code>
  </script>
</condition>
</mapping>
<mapping> <!-- вычисление имени учётной записи--
>

  <source>
    <name>familyName</name>
    <c:path>$user/familyName</c:path>
  </source>
  <source>
    <name>givenName</name>
    <c:path>$user/givenName</c:path>
  </source>
```

```

<source>
    <name>additionalName</name>
    <c:path>$user/additionalName</c:path>
</source>
<expression>
    <script>
        <code>
            import
ru.solarsecurity.inrights.common.policy.StringPolicyUtils;

            StringBuilder sb = new
StringBuilder();

            int charsToInclude = iteration + 2;
            if (familyName != null) {

sb.append(StringPolicyUtils.transliterateString(familyName.norm));

            }

            str =
StringPolicyUtils.getSuffixFromStrings
                ([givenName !=
                    null?givenName.norm:"",
additionalName!=null?additionalName.norm:""], ".",
iteration);

            if (str.length() > 0)
sb.append(".").append(str);
            if (sb.length() > 20) {
sb.delete(17, sb.length());
            if (sb.charAt(sb.length()-1) == '.')
//в AD имя не должно оканчиваться точкой
                sb.delete(sb.length()-
1,sb.length()-1);

sb.append(String.valueOf(iteration > 0 ?
iteration : ""));

            }

```

```
        sb.toString();
    </code>
</script>
</expression>
<target>
    <c:path>$user/name</c:path>
</target>
<condition>
    <script>
        <code>givenName != null &&
familyName != null && additionalName !=
null</code>
    </script>
</condition>
</mapping>
<mapping> <!-- блокировка пользователя на
основании статуса в кадровой системе-->
    <authoritative>false</authoritative>
    <source>
        <name>hrStatus</name>
        <c:path>$user/hrStatus</c:path>
    </source>
    <expression>
        <script>
            <code>
                import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.ActivationStatusType;

                return
ActivationStatusType.DISABLED;
            </code>
        </script>
    </expression>
</target>
```

```
<c:path>$user/activation/administrativeStatus</c:pa  
th>  
  
    </target>  
    <condition>  
        <script>  
            <code>  
                import  
ru.solarsecurity.inrights.xml.ns._public.common.com  
mon_3.HrStatusType;  
                return hrStatus ==  
HrStatusType.FIRED;;  
            </code>  
        </script>  
    </condition>  
</mapping>  
  
<mapping><!-- Активация учётной записи на  
основании статуса в кадровой системе-->  
    <authoritative>>false</authoritative>  
    <strength>weak</strength>  
    <source>  
        <name>hrStatus</name>  
        <c:path>$user/hrStatus</c:path>  
    </source>  
    <expression>  
        <script>  
            <code>  
                import  
ru.solarsecurity.inrights.xml.ns._public.common.com  
mon_3.ActivationStatusType;  
                return ActivationStatusType.ENABLED;  
            </code>  
        </script>  
    </expression>  
<target>
```

```
<c:path>$user/activation/administrativeStatus</c:pa  
th>  
    </target>  
    <condition>  
        <script>  
            <code>  
                import  
ru.solarsecurity.inrights.xml.ns._public.common.com  
mon_3.HrStatusType;  
                return hrStatus ==  
HrStatusType.ACTIVE;  
            </code>  
        </script>  
    </condition>  
</mapping>  
<mapping>  
    <authoritative>true</authoritative>  
    <source>  
        <name>hrStatus</name>  
        <c:path>$user/hrStatus</c:path>  
    </source>  
    <expression>  
        <assignmentTargetSearch>  
            <targetType>c:OrgType</targetType>  
            <oid>07a55aad-341a-461a-b9f7-  
6a0a274741eb</oid>  
        </assignmentTargetSearch>  
    </expression>  
    <target>  
        <c:path>assignment</c:path>  
    </target>  
    <condition>  
        <script>
```

```
<code>
    import
    ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.HrStatusType;

    return hrStatus ==
HrStatusType.FIRED;
</code>
</script>
</condition>
</mapping>
<mapping>
    <authoritative>true</authoritative>
    <source>
        <name>hrStatus</name>
        <c:path>$user/hrStatus</c:path>
    </source>
    <source>
        <name>title</name>
        <c:path>$user/title</c:path>
    </source>
    <source>
        <name>asmt</name>
        <c:path>$user/assignment</c:path>
    </source>
    <expression>
        <script>
            <code>
                import
                ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.ObjectReferenceType;

                import
                ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.UserType;
```

```
import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

import
ru.solarsecurity.inrights.requests.impl.*;

import
ru.solarsecurity.inrights.prism.query.*;

import java.lang.Exception;

import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;

import
ru.solarsecurity.inrights.prism.PrismObject;

import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;

import
ru.solarsecurity.inrights.prism.query.EqualFilter;

import
ru.solarsecurity.inrights.prism.PrismContext;

import
ru.solarsecurity.inrights.prism.*;

import
ru.solarsecurity.inrights.prism.path.*;

PrismContext prismContext =
SpringApplicationContextHolder.getBean(PrismContext
.class);

ModelController modelService =
SpringApplicationContextHolder.getBean(ModelControl
ler.class);

AssignmentType at = asmt;

if(at != null &&&
at.getTargetRef() !=
```



```

        null &&&;
at.getTargetRef().getType().getLocalPart().equals("
OrgType")) {

        String orgOid = ((AssignmentType)
at).getTargetRef().getOid();

        ObjectFilter filter =
OrgFilter.createOrg(orgOid,
OrgFilter.Scope.ONE_LEVEL);

        OperationResult result = new
OperationResult("get_object_by_employee_number");
        ObjectQuery objectQuery = new
ObjectQuery();

        objectQuery.setFilter(filter);

        List rolesInOrg =
modelService.searchObjects(RoleType.class,
objectQuery, null, null, result);

        for(Object role : rolesInOrg){
            PrismObject prismRole =
(PrismObject)role;

if(prismRole.asObjectable().getDisplayName() !=
null &&&;
prismRole.asObjectable().getDisplayName().toString(
).equals(title.toString())){

                AssignmentType
assignmentType = new AssignmentType();

                ObjectReferenceType
objectReferenceType = new ObjectReferenceType();

                objectReferenceType.setOid(prismRole.getOid());

                objectReferenceType.setType(RoleType.COMPLEX_TYPE);

assignmentType.setTargetRef(objectReferenceType);

```

```

                                prismContext.adopt(

assignmentType.asPrismContainerValue(),
FocusType.COMPLEX_TYPE, new
ItemPath(FocusType.F_ASSIGNMENT));

                                return assignmentType;
                                }
                                }
                                }

</code>
</script>
</expression>
<target>
    <c:path>assignment</c:path>
</target>
<condition>
    <script>
        <code>
            import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.HrStatusType;

            return hrStatus !=
HrStatusType.FIRED;
        </code>
    </script>
</condition>
</mapping>
<mapping>
    <authoritative>true</authoritative>
    <source>
        <name>hrStatus</name>
        <c:path>$user/hrStatus</c:path>

```

```
</source>
<source>
  <name>title</name>
  <c:path>$user/title</c:path>
</source>
<source>
  <name>orgIdentifier</name>
  <c:path>$user/organizationalUnit</c:path>
</source>
<expression>
  <script>
    <code>
      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.ObjectReferenceType;
      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.UserType;
      import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;
      import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
      import
ru.solarsecurity.inrights.requests.impl.*;
      import
ru.solarsecurity.inrights.prism.query.*;
      import java.lang.Exception;
      import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;
      import
ru.solarsecurity.inrights.prism.PrismObject;
      import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;
```

```
import
ru.solarsecurity.inrights.prism.query.EqualFilter;

import
ru.solarsecurity.inrights.prism.PrismContext;

import
ru.solarsecurity.inrights.prism.*;

import
ru.solarsecurity.inrights.prism.path.*;


PrismContext prismContext =
SpringApplicationContextHolder.getBean(PrismContext
.class);

ModelController modelService =
SpringApplicationContextHolder.getBean(ModelControl
ler.class);


ObjectFilter filterOrg =
EqualFilter.createEqual(

OrgType.F_IDENTIFIER, OrgType.class, prismContext,
orgIdentifier.toString())

ObjectQuery objectQueryOrg = new
ObjectQuery();

objectQueryOrg.setFilter(filterOrg);

OperationResult result = new
OperationResult("get_object_by_employee_number");

List rolesInOrg1 =
modelService.searchObjects(OrgType.class,
objectQueryOrg, null, null, result);

if(rolesInOrg1.size() == 0)
return;


ObjectFilter filter =
OrgFilter.createOrg(rolesInOrg1.get(0).getOid(),
OrgFilter.Scope.ONE_LEVEL);
```

```

        ObjectQuery objectQuery = new
ObjectQuery();

        objectQuery.setFilter(filter);

        List rolesInOrg =
modelService.searchObjects(RoleType.class,
objectQuery, null, null, result);

        for(Object role : rolesInOrg){
            PrismObject prismRole =
(PrismObject)role;

            if(prismRole.asObjectable().getDisplayName() !=
                null &&&
prismRole.asObjectable().getDisplayName().toString(
).equals(title.toString())){

                AssignmentType assignmentType
= new AssignmentType();

                ObjectReferenceType
objectReferenceType = new ObjectReferenceType();

                objectReferenceType.setOid(prismRole.getOid());

                objectReferenceType.setType(RoleType.COMPLEX_TYPE);

                assignmentType.setTargetRef(objectReferenceType);

                prismContext.adopt(

assignmentType.asPrismContainerValue(),
FocusType.COMPLEX_TYPE, new
ItemPath(FocusType.F_ASSIGNMENT));

                return assignmentType;
            }
        }
    }
</code>
</script>
</expression>
<target>

```

```

        <c:path>assignment</c:path>
    </target>
    <condition>
        <script>
            <code>
                import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.HrStatusType;

                return hrStatus !=
HrStatusType.FIRED;
            </code>
        </script>
    </condition>
</mapping>
</objectTemplate>

```

4.2 Управление УЗ пользователей в целевых системах

4.2.1 Настройка соответствия полей в ПС полям в подключенной ИС (маппинг атрибутов)

Из систем-источников данных в ПС загружаются сведения о конкретном человеке, которые считаются истинными/эталонными. Настройка остальных ресурсов (управляемых систем) предполагает установку соответствия загруженных в ПС сведений о человеке с его учётными записями в конкретных системах. У объекта, который хранит информацию о человеке в ПС, имеются атрибуты (Имя, Фамилия и т.д.). При этом в каждой ИС для учётной записи также определён свой набор атрибутов.

Правила маппинга атрибутов указывают соответствие между атрибутами объекта в ПС и атрибутами объекта в другой ИС. Правила inbound и outbound указывают направление передачи информации: в ПС (**inbound**) или в стороннюю ИС (**outbound**).

Примечание

Подробнее о процессе взаимодействия см. в разделе 1.3.

Маппинг атрибутов выполняется при создании / редактировании конкретного ресурса (либо в разделе Новая инфомарционная система> Обработка схемы, см. раздел 3.6.1, либо при редактировании xml-кода соответствующего ресурса, выбранного в разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории)**)).

Для маппинга атрибутов в блоке **attribute** необходимо настроить параметры секции outbound: указывается поле объекта в ПС, из которого будет записываться значение в конкретное поле в управляемой информационной системе.

Правила маппинга состоят из следующих частей:

- **Source** — определяет источник данных для маппинга;
- **Expression** — определяет способ преобразования данных, их генерации и передачи "на другую сторону";
- **Target** — определяет дальнейшие действия с результатами маппинга;
- **Condition** — условие применения маппинга. Если условие принимает истинное значение (true), маппинг будет применен. В противном случае, система пропустит маппинг;

Сила маппинга — атрибут маппинга, определяющий его приоритет:

- **Strong** — маппинг, имеющий силу strong, будет применяться всегда, независимо от контекста.
- **Normal** — если в правиле нет более конкретных указаний для применения маппинга.
- **Weak** — используется для установки начальных значений атрибутов и свойств.

Примеры:

1) Пример установки соответствия полей familyName в ПС и подключенной системе:

```
<attribute>
  <c:ref>ri:sn</c:ref>
  <displayName>Surname</displayName>
  <outbound>
    <source>
      <c:path>$user/familyName</c:path>
    </source>
  </outbound>
  <inbound>
    <target>
      <c:path>$user/familyName</c:path>
    </target>
  </inbound>
</attribute>
```

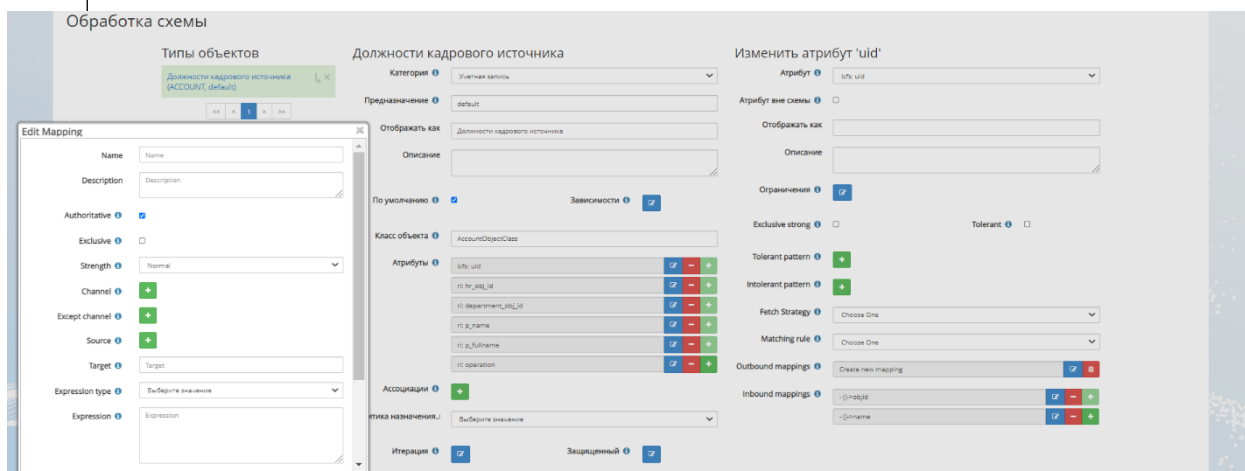


Рис. 73 – Раздел Новая информационная система > Обработка схемы:
маппинг атрибутов

2) Пример настройки DistinguishName:

Примечание

DistinguishName (DN) - это уникальный древовидный идентификатор в AD. Можно указать, по какому правилу связывать учётную запись в ПС с учётной записью в AD.


```

<attribute>
  <c:ref>icfs:name</c:ref>
  <displayName>Distinguished Name</displayName>
  <limitations>
    <minOccurs>0</minOccurs>
  </limitations>
  <outbound>
    <source>
      <c:path>$user/name</c:path>
    </source>
    <source>
      <c:path>$user/extension/orgIdentifier</c:path>
    </source>
    <source>
      <c:path>$user/parentOrgRef</c:path>
    </source>
    <expression>
      <script>
        <code>
          import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

          OrgType getParentOrg(org) {
            if(org == null) return null;
            def parentOrgs =
org.getParentOrgRef();
            if(parentOrgs.isEmpty()) {
              return null;
            }
            OrgType parentOrg =
inrights.getOrgByOid(parentOrgs.find{true}?.getOid(
));

```

```
        return parentOrg;
    }

    OrgType getRootOrg(org) {
        if(getParentOrg(org) == null) {
            return org;
        }
        return
getRootOrg(getParentOrg(org));
    }

    Set orgs =
inrights.getOrgUnits(focus);
    String firstOrgOid =
orgs?.find{true}

        if(firstOrgOid == null) {
            return
'cn='+name+',ou=inrights,dc=isim,dc=local';
        }

        OrgType firstOrg =
inrights.getOrgById(firstOrgOid)
        OrgType rootOrg =
getRootOrg(firstOrg);

        if(rootOrg == null) {
            return
'cn='+name+',ou=inrights,dc=isim,dc=local';
        }

'cn='+name+',ou='+rootOrg.getName()+',ou=inrights,d
c=isim,dc=local'
</code>
```

```

</script>
</expression>
<condition>
...
</condition>
</outbound>
</attribute>

```

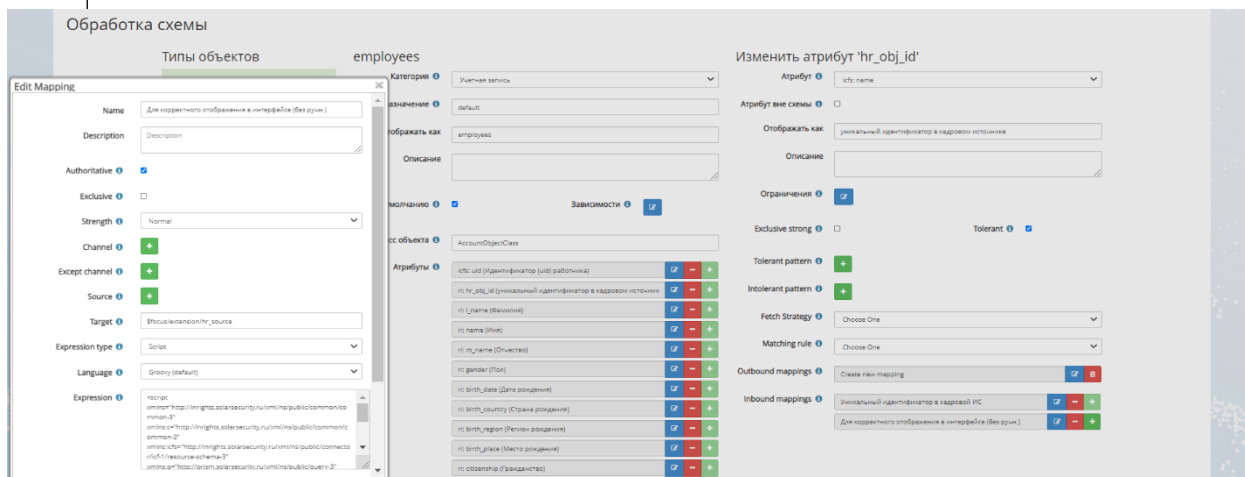


Рис. 74 – Раздел Новая информационная система > Обработка схемы: создание и настройка DistinguishName

3) Пример записи значения, хранящегося в поле ПС, в соответствующее поле ИС:

Примечание

В параметре **ri:RecipientType** определяется, требуется ли при создании учетной записи в AD создавать еще и почтовый ящик в Exchange. В данном примере передается значение, определяющее, что требуется создание почтового ящика в Exchange. То есть в параметр **ri:RecipientType** записывается значение **UserMailbox**.

```

<attribute>
  <c:ref>ri:RecipientType</c:ref>
  <displayName>Тип учетной записи</displayName>
  <limitations>
    <minOccurs>0</minOccurs>
  </limitations>
</attribute>

```

```
<expression>
  <script>
    <code>'UserMailbox'</code>
  </script>
</expression>
</outbound>
</attribute>
```

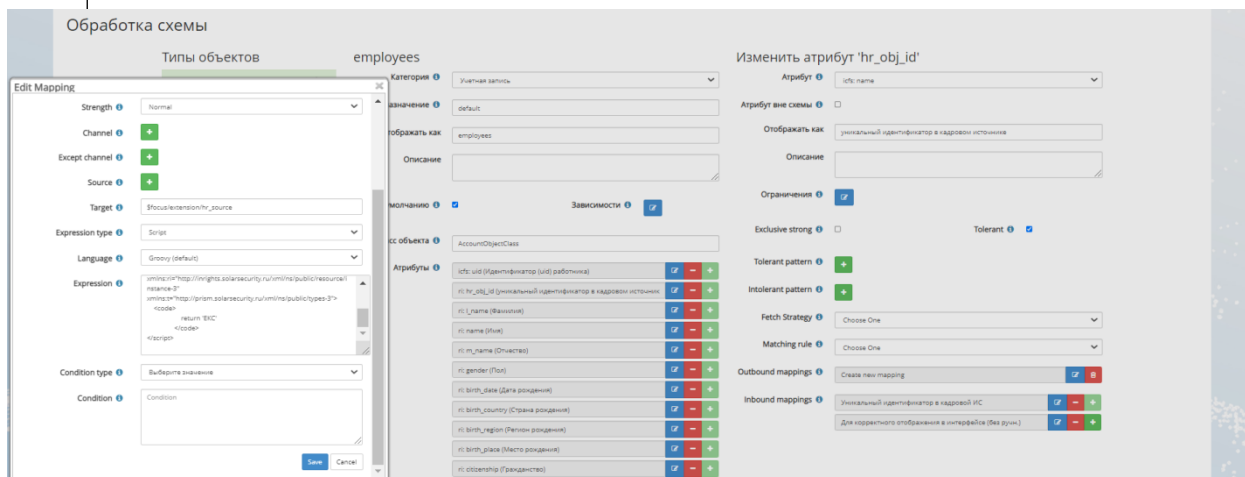


Рис. 75 – Раздел Новая информационная система > Обработка схемы:
маппинг атрибутов

4.2.2 Настройка генерации имен пользователей

Генерация уникальных имен пользователей обеспечивается с помощью специального скрипта, в котором организовано повторение необходимых операций. Эти операции выполняются в цикле до успешного завершения генерации имени пользователя (login) на основе введенных им данных (ФИО).

Для генерации уникальных имен пользователей необходимо в шаблоне профиля пользователя (см. раздел 4.1.3.2(подпункт (**Error! Reference source not found.**))) прописать:

1) Строки

```
<iteration>
  <maxIterations>N</maxIterations>
```

`</iteration>`

где N - количество неуспешных попыток (например, 9999).

2) Код:

```
<expression>
    <script>
        <code>
            import
ru.solarsecurity.inrights.common.policy.StringPolicyUtils;

            StringBuilder sb = new StringBuilder();

            int charsToInclude = iteration + 2;
            if (familyName != null) {

sb.append(StringPolicyUtils.transliterateString(familyName.norm));
            }

            str =
StringPolicyUtils.getSuffixFromStrings(
                [givenName !=
null?givenName.norm:"",
additionalName!=null?additionalName.norm:""], ".",
iteration

            );
            if (str.length() > 0)
sb.append(".").append(str);
            if(sb.length() > 20) {
                sb.delete(17, sb.length());
                if(sb.charAt(sb.length()-1) == '.')
//В АД имя не должно оканчиваться точкой
                sb.delete(sb.length()-1, sb.length()-1);

                sb.append(String.valueOf(iteration
> 0 ? iteration : ""));
```

```
    }  
    sb.toString();  
</code>  
</script>  
</expression>
```

где:

- *StringPolicyUtils.transliterateString* - функция, которая транслитерирует русскую строку в латинскую.
- *StringPolicyUtils.getSuffixFromStrings([givenName != null?givenName.norm:"", additionalName!=null?additionalName.norm:""], ".", iteration)* - функция, которая формирует суффикс (для суффикса используются 2 первые буквы *givenName* и *additionalName* + значение переменной *iteration*). При этом применяется правило:
$$S1[1..i1].S2[1..i2]. \quad i1 + i2 = iteration+2. \quad i1 \geq i2. \\ i1 - i2 \leq 1$$

Примеры работы генератора имен пользователей:

```
iteration = 0;  
givenName = "Иванов"  
additionalName = "Иванович"  
result = "i.i."
```

```
iteration = 1;  
givenName = "Иванов"  
additionalName = "Иванович"  
result = "iv.i."
```

```
iteration = 2;  
givenName = "Иванов"  
additionalName = "Иванович"  
result = "iv.iv."
```

```
iteration = 3;  
givenName = "Иванов"  
additionalName = "Иванович"  
result = "iva.iv."
```

При этом цикл по **iteration** гарантирует уникальность имени пользователя.

4.2.3 Настройка запрета повторного использования имен УЗ пользователей

Для настройки запрета повторного использования имен учетных записей пользователей (к примеру, если нежелательно использовать имена УЗ, которые были удалены) необходимо отредактировать xml-код объекта **ConfigurationObjectType** (Конфигурирование - Объекты в репозитории - Конфигурация):

```
<!--oid = configuration.security name = Security  
Configuration-->
```

```
<security>
```

```
...
```

```
<userNamePolicy>
```

```
    <reuseAfter>P365D</reuseAfter>    <!--
```

*Период, в течение которого нельзя повторно
использовать зарезервированное имя УЗ.*

*Отсчет периода начинается с того времени, когда УЗ
либо становится свободной*

*(свободная УЗ не привязана ни к одному
пользователю), либо удалена.*

*Отсутствие параметра означает, что ограничений на
повторное использование нет -->*

```
</userNamePolicy>
```

```
...
```

```
<enabled>true</enabled>    <!--  
включает/выключает всю конфигурацию целиком, по  
умолчанию false -->  
</security>
```

Формат значения периода аналогичен **java.time.Duration**.

Примеры:

- PT15M - 15 минут;
- PT10H - 10 часов;
- P2D - 2 дня;
- P2DT3H4M - 2 дня, 3 часа и 4 минуты.

4.2.4 Настройка генерации паролей

Для настройки генерации паролей необходимо определить политику генерации паролей, т.е. набор правил, по которым будут генерироваться пароли.

4.2.4.1 Задание политики генерации паролей

Политика генерации паролей задается в XML-формате. Загрузка (импорт) политики генерации паролей в ПС выполняется в AGUI в разделе **Загрузка объектов (Конфигурирование > Загрузить объект, подробнее см. раздел 3.8.1)**.

Пример политики генерации паролей:

```
<?xml version="1.0" encoding="UTF-8"?>  
<object xsi:type="ValuePolicyType" version="0"  
xmlns="http://inrights.solarsecurity.ru/xml/ns/publ  
ic/common/common-3"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance">  
    >  
    <name>Testing Complex Password Policy</name>  
    <description>Testing complex password  
policy</description>
```



```
<lifetime>
  <expiration>999</expiration>

<warnBeforeExpiration>9</warnBeforeExpiration>
  <lockAfterExpiration>0</lockAfterExpiration>
  <minPasswordAge>0</minPasswordAge>

<passwordHistoryLength>0</passwordHistoryLength>
</lifetime>
<stringPolicy>
  <description>Testing string
policy</description>
  <limitations>
    <minLength>5</minLength>
    <maxLength>8</maxLength>
    <minUniqueChars>3</minUniqueChars>

<checkAgainstDictionary>true</checkAgainstDictionary>

  <checkPattern />
  <limit>
    <description>extreme</description>
    <minOccurs>1</minOccurs>
    <maxOccurs>2</maxOccurs>
    <mustBeFirst>true</mustBeFirst>
    <characterClass>
      <value>345678</value>
    </characterClass>
  </limit>
  <limit>
    <description>Alphas</description>
    <minOccurs>1</minOccurs>
    <maxOccurs>2</maxOccurs>
    <mustBeFirst>false</mustBeFirst>
```

```
<characterClass ref="alphas" />
</limit>
<limit>
  <description>Numbers</description>
  <minOccurs>1</minOccurs>
  <maxOccurs>2</maxOccurs>
  <mustBeFirst>false</mustBeFirst>
  <characterClass ref="numbers" />
</limit>
<limit>
  <description>Lowers</description>
  <minOccurs>1</minOccurs>
  <maxOccurs>2</maxOccurs>
  <mustBeFirst>false</mustBeFirst>
  <characterClass ref="lowers" />
</limit>
<limit>
  <description>Specials</description>
  <minOccurs>1</minOccurs>
  <maxOccurs>2</maxOccurs>
  <mustBeFirst>false</mustBeFirst>
  <characterClass ref="special" />
</limit>
<limit>
  <description>Alphanum</description>
  <minOccurs>1</minOccurs>
  <maxOccurs>3</maxOccurs>
  <mustBeFirst>false</mustBeFirst>
  <characterClass ref="alphanum" />
</limit>
</limitations>
<characterClass name="default">
  <characterClass name="alphanum">
```

```
<characterClass name="alphas">
  <characterClass name="lowers">
    <value>abcd</value>
  </characterClass>
  <characterClass name="uppers">
    <value>ABCD</value>
  </characterClass>
</characterClass>
<characterClass name="numbers">
  <value>1234</value>
</characterClass>
<characterClass name="special">
  <value>!@#$%+*</value>
</characterClass>
</characterClass>
</stringPolicy>
</object>
```

Можно задать значения следующих параметров:

- **minLength** - минимальная длина пароля;
- **maxLength** - максимальная длина пароля;
- **minUniqueChars** - минимальное количество уникальных

символов в пароле.

В секции **limit** задаются ограничения для конкретных типов символов.

Кроме того, в соответствии с требованиями безопасности администратор может установить максимальные сроки действия и активации пароля, а также настроить запрет смены пароля при нарушении парольной политики.

4.2.4.1.1 Установка максимального срока действия пароля

Для установки максимального срока действия пароля следует в **Security Configuration** в блоке **passwordPolicy/maxActivationPeriod** добавить код:

```
<configuration>
```

```
  <crt:security>
```

```
    .....
```

```
      <crt:passwordPolicy>
```

```
        <crt:maxValidityPeriod>PT15M</crt:maxValidityPeriod>
```

```
      </crt:passwordPolicy>
```

```
    .....
```

```
  </crt:security>
```

```
</configuration>
```

Формат значения срока действия аналогичен **java.time.Duration**.

Примеры:

- PT20.345S - 20.345 секунд;
- PT15M - 15 минут;
- PT10H - 10 часов;
- P2D - 2 дня;
- P2DT3H4M - 2 дня, 3 часа и 4 минуты.

Кроме того, необходимо задать значение параметра системной конфигурации **expiringNotification** (в блоке **notificationConfiguration**). Этот параметр отвечает за период, с которого начинается отправка уведомлений пользователю, у которого истекает срок действия пароля. То есть если до момента, когда пароль считается просроченным, осталось меньше, чем **expiringNotification** дней, то пользователю будут отправляться уведомления.

4.2.4.1.2 Установка максимального срока активации пароля

Для установки максимального срока активации пароля следует в **Security Configuration** в блоке **passwordPolicy/maxActivationPeriod** добавить код:

```
<configuration>
  <cert:security>
    .....
    <cert:passwordPolicy>

<cert:maxActivationPeriod>PT15M</cert:maxActivationPe
riod>

    </cert:passwordPolicy>
    .....
  </cert:security>
</configuration>
```

Формат значения срока активации аналогичен **java.time.Duration**.

Примеры:

- PT20.345S - 20.345 секунд;
- PT15M - 15 минут;
- PT10H - 10 часов;
- P2D - 2 дня;
- P2DT3H4M - 2 дня, 3 часа и 4 минуты.

4.2.4.1.3 Запрет смены пароля при нарушении парольной политики

Парольная политика включает:

- запрет изменения пароля, если новый пароль совпадает с одним из n1 предыдущих;
- запрет изменения пароля, если новый пароль отличается от старого менее чем на n2 символов.

Настройка производится в **Default Password Policy (admin-gui -> Объекты в репозитории -> Value Policy)** в блоке **extension**:

```
<extension>
```

```

        <passwordHistoryDepth
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="xsd:int">10</passwordHistoryDepth>
        <passwordDiffPolicy
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="xsd:int">3</passwordDiffPolicy>
</extension>

```

Указать необходимость генерации пароля в соответствии с парольной политикой для объекта в маппингах можно следующим образом:

```

<expression>
    <generatePassword/>
</expression>

```

4.2.4.2 Использование политики генерации паролей

Политика генерации паролей может быть использована, например, в шаблоне профиля пользователя:

```

<objectTemplate
xmlns="http://inrights.solarsecurity.ru/xml/ns/publ
ic/common/common-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/resource-schema-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/publi
c/types-3"

```

```
xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"
```

```
xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"
```

```
xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"
```

```
oid="246f7b2e-20cd-4377-8848-a1ebb5c0bcc6"
```

```
version="21">
```

```
<name>Стандартный пользователь</name>
```

```
<description>Формирует имя для входа на основе  
ФИО пользователя. Генерирует пароль (при импорте из  
HR) .
```

```
Заполняет полное имя. Привязывает к  
подразделению (если заполнен organizationalUnit).
```

```
Assignment роли Сотрудник
```

```
</description>
```

```
<metadata>
```

```
<createTimestamp>2015-05-  
28T14:21:21.856+03:00</createTimestamp>
```

```
<creatorRef oid="00000000-0000-0000-0000-  
000000000002" type="c:UserType"><!-- administrator  
--></creatorRef>
```

```
<createChannel>http://inrights.solarsecurity.ru/xml  
/ns/public/model/channels-  
3#objectImport</createChannel>
```

```
</metadata>
```

```
<iteration>
```

```
<maxIterations>9999</maxIterations>
```

```
</iteration>
```

```
<mapping>
```

```
<source>
```

```
<name>password</name>
```

```
<c:path>$user/credentials/password/value</c:path>
  </source>
  <expression>
    <generate>
      <valuePolicyRef oid="00000000-0000-
0000-0000-000000000003"/>
    </generate>
  </expression>
  <target>

<c:path>$user/credentials/password/value</c:path>
  </target>
  <condition>
    <script>
      <code>
        return password == null ||
password.isEmpty();
      </code>
    </script>
  </condition>
</mapping>
</objectTemplate>
```

В шаблоне в параметре **valuePolicyRef** задается идентификатор соответствующего объекта **oid**. Получить сведения об **oid** требуемой политики генерации паролей можно, выбрав соответствующий объект в разделе **Объекты из хранилища** (Конфигурирование > Объекты в репозитории, см. Рис. 76)

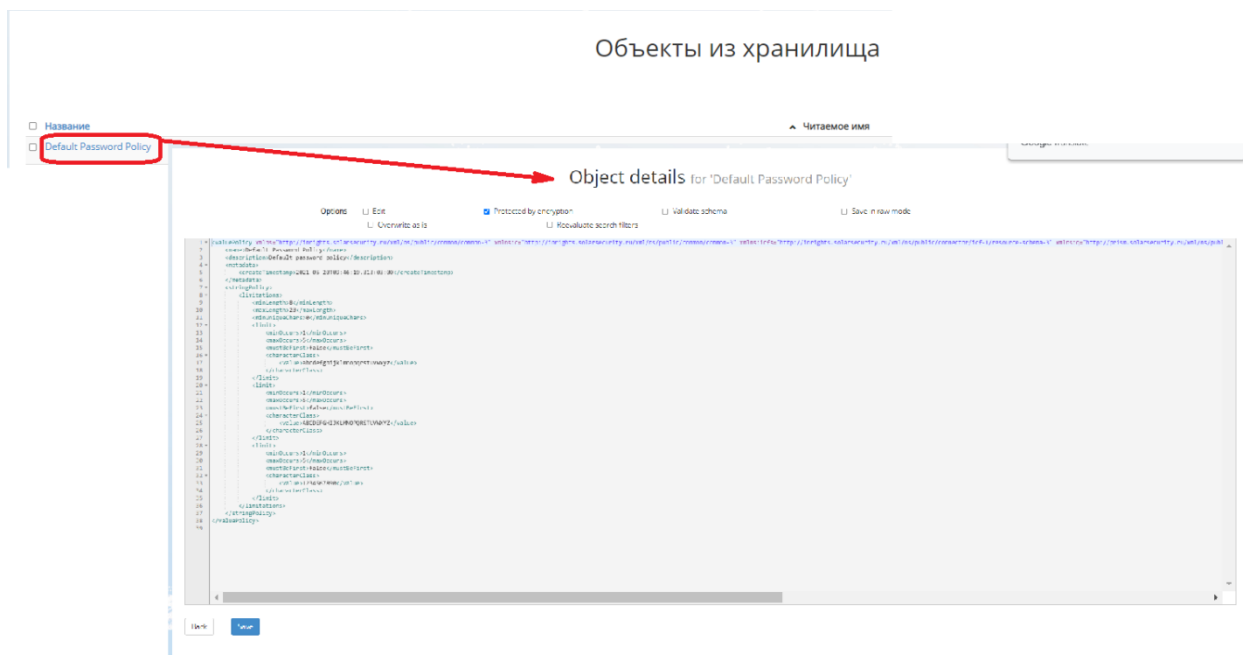


Рис. 76 – Получение сведений об oid

4.2.5 Настройка отображения карточки пользователя

В начале XML-описания объекта **SystemConfiguration** необходима ссылка на стандартную форму отображения/редактирования пользователя, например:

```
<systemConfiguration oid="00000000-0000-0000-0000-000000000001" version="0"

xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3">

...

<defaultUserFormRef oid="00000000-0000-0000-0000-0000000000011" type="c:FormType"/>

</systemConfiguration>
```

Для формирования карточки необходимо импортировать данные
объекта **FormType**:

```
<form
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

    oid="00000000-0000-0000-0000-000000000011">
    <name>User form</name>
    <metadata>
        <createTimestamp>2016-07-
20T17:29:24.898+03:00</createTimestamp>
        <createChannel>
http://inrights.solarsecurity.ru/xml/ns/public/gui/channels-3#init
        </createChannel>
    </metadata>
    <model>
        <attribute context="userType"
            displayName="Фамилия"
            name="lastName"
            ref="familyName"
```

```
        required="true"
        readonly="false"
        type="attribute">
        <input placeholder="Обязательное поле"
width="224"/>
    </attribute>
    <attribute context="userType"
        displayName="Имя"
        name="firstName"
        ref="givenName"
        readonly="false"
        required="true"
        type="attribute">
        <input placeholder="Обязательное поле"
width="224"/>
    </attribute>
    <attribute context="userType"
        displayName="Отчество"
        name="additionalName"
        ref="additionalName"
        readonly="false"
        type="attribute">
        <input width="224"/>
    </attribute>
    <attribute context="userType"
        displayName="Эл. почта"
        name="emailAddress"
        ref="emailAddress"
        readonly="true"
        required="true"
        type="attribute">
        <input placeholder="Обязательное поле"
width="224"/>
    </attribute>
```

```
<attribute context="orgType"
    displayName="Подразделение"
    name="parentOrgs"
    ref="org"
    readonly="false"
    type="orgAssignment">
    <orgPicker/>
</attribute>
<attribute context="userType"
    displayName="Должность"
    name="title"
    readonly="false"
    ref="title"
    type="attribute">
    <input width="224"/>
</attribute>
<attribute context="userType"
    displayName="Принят на работу"
    name="dateFrom"
    ref="workstart"
    readonly="false"
    required="true"
    type="attribute">
    <datePicker placeholder="Обязательное
поле"/>
</attribute>
<attribute context="userType"
    displayName="Уволен"
    name="workend"
    readonly="false"
    ref="workend"
    type="attribute">
    <datePicker/>
```

```
</attribute>
<attribute context="userType"
            displayName="Кадровый статус"
            name="hrStatus"
            ref="hrStatus"
            readonly="false"
            required="true"
            type="attribute">
    <select defaultValue="active" width="214">
        <value displayName="Работает"
val="active"/>
        <value displayName="Уволен"
val="fired"/>
        <value displayName="В отпуске"
val="vacation"/>
        <value displayName="В декретном
отпуске" val="maternityleave"/>
    </select>
</attribute>
<attribute context="userType"
            displayName="Руководитель"
            name="manager"
            readonly="false"
            ref="managerRef"
            type="manager">
    <userPicker/>
</attribute>
</model>
</form>
```

Параметры:

- **context** - указывает объект, который является источником данных: **userType** - пользователь, **orgType** - подразделение.

- **ref** - ссылка на конвертер, который будет получать данные объекта, указанного в **context**.

Примечание

Конвертер - это класс, преобразующий данные, хранимые в системе, в данные для просмотра в карточке пользователя. Список конвертеров, используемых в ПС по умолчанию, приведен в таблице ниже (см. Таблица 9 –).

- **displayName** - атрибут, отвечающий за отображение названия поля в карточке пользователя;
- **name** - атрибут, который отвечает за идентификатор поля в карточке пользователя;
- **readonly** - атрибут, указывающий, что данные только для просмотра;
- **required** - не обязательный атрибут, указывающий, что при изменении пользователя поле должно быть не пустым.

Также указывается тип компонента, который отображается в GUI:

- **input** - однострочное текстовое поле;
- **datePicker** - компонент для выбора даты;
- **orgPicker** - компонент для выбора подразделения;
- **userPicker** - компонент для выбора пользователя;
- **select** - выпадающий список возможных вариантов выбора.

Таблица 9 – Список конвертеров, используемых в ПС по умолчанию

№	Значение поля ref	Тип конвертера	Описание
Конвертеры для userType			
1.	fullName	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
2.	givenName	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое

			хранится в поле типа PolyString
3.	familyName	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
4.	additionalName	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
5.	nickName	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
6.	honorificPrefix	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
7.	honorificSuffix	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
8.	organization	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
9.	organizationalUnit	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString
10.	title	PolyStringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа PolyString

11.	preferredLanguage	StringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа String
12.	locale	StringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа String
13.	timezone	StringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа String
14.	emailAddress	StringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа String
15.	telephoneNumber	StringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа String
16.	employeeNumber	StringDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа String
17.	hrStatus	HrStatusDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа HrStatus
18.	managerRef	UserReferenceDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа UserReferenceDataConverter
19.	workstart	XMLGregorianCalendarDataConverter	Преобразует соответствующее поле из объекта userType, которое хранится в поле типа XMLGregorianCalendarDataConverter
20.	workend	XMLGregorianCalendarDataConverter	Преобразует соответствующее поле из объекта userType, которое

			хранится в поле типа XMLGregorianCalendarDataConverter
Конвертеры для OrgType			
21.	org	AssignmentDataConverter	Читает из репозитория объект типа AssignmentObjectType с AssignmentKindType.ORG и возвращает название для отображения + OID. При сохранении редактирует ссылки на массив объектов AssignmentObjectType

Примечание

Для расширения списка конвертеров для **UserType** необходимо вызвать метод **UserFormCardProcessor.addConverter(String key, AbstractDataConverter converter)**, где key — это значение, которое надо использовать в поле **ref**.

Для расширения списка конвертеров для **OrgType** необходимо вызвать метод **OrgFormCardProcessor.addConverter(String key, AbstractDataConverter converter)**.

4.2.6 Настройка временной блокировки УЗ пользователя после неуспешных попыток входа

Для настройки временной блокировки учетной записи пользователя после неуспешных попыток входа необходимо отредактировать xml-код объекта **ConfigurationObjectType** (Конфигурирование - Объекты в репозитории - Конфигурация):

```

<!--oid = configuration.security name = Security
Configuration-->
<security>
    ...
    <authenticationPolicy>
        <temporaryLockout>

```

```
<enabled>true</enabled> <!-- Включена
ли данная настройка, по умолчанию false -->

<maxFailedAttempts>5</maxFailedAttempts> <!--
Максимальное количество попыток, после которого
пользователь временно заблокируется, по умолчанию 3
-->

    <within>PT10M</within> <!-- Период
времени, за который нужно превысить количество
попыток, чтобы заблокировался пользователь, по
умолчанию PT5M (5 мин) -->

<lockoutDuration>PT30M</lockoutDuration> <!--
Период, на который осуществляется временная
блокировка, по умолчанию PT15M (15 мин) -->
    </temporaryLockout>
</authenticationPolicy>
...
    <enabled>true</enabled> <!-- Включает/выключает
всю конфигурацию целиком, по умолчанию false -->
</security>
```

Формат значения периода аналогичен **java.time.Duration**.

Примеры:

- PT15M - 15 минут;
- PT10H - 10 часов;
- P2D - 2 дня;
- P2DT3H4M - 2 дня, 3 часа и 4 минуты.

4.2.7 Ручная разблокировка УЗ пользователя, заблокированного после неуспешных попыток входа

При временной блокировке УЗ пользователя, которая произошла после неуспешных попыток входа, в xml-код объекта (конкретного пользователя) записывается атрибут **extension/tempLockout**, указывающий, что вход для пользователя временно заблокирован, а также момент времени, до которого блокировка активна:

```
<user>
  <extension
xmlns:crt="http://inrights.solarsecurity.ru/plugins
/certification">

    <crt:tempLockout
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="xsd:dateTime">2017-11-
10T17:19:49.624+03:00</crt:tempLockout>

  </extension>
</user>
```

Администратор может разблокировать УЗ вручную, не дожидаясь срока окончания блокировки. Для этого необходимо отредактировать xml-код соответствующего объекта, удалив атрибут **extension/tempLockout**. В результате история неудачных попыток входа пользователя очищается, и он разблокируется.

4.2.8 Настройка блокировки неактивных УЗ пользователей

Для настройки автоматической блокировки неактивных (неиспользуемых) учетных записей пользователей необходимо отредактировать xml-код объекта **ConfigurationObjectType** (Конфигурирование - Объекты в репозитории - Конфигурация):

```
<!--oid = configuration.security name = Security
Configuration-->
<security>

  ...

  <userPolicy>

    <inactivityPeriod>P90D</inactivityPeriod>
<!-- Период, после которого блокируются неактивные
пользователи. По умолчанию период равен 90 дней-->
```

```

        <protectedUser oid="admin-oid"
type="c:UserType" />    <!-- Список пользователей,
которые не будут блокироваться (даже если превышен
период
их неактивности) -->
    </userPolicy>

    ...

    <enabled>true</enabled>    <!--
включает/выключает всю конфигурацию целиком, по
умолчанию false -->
</security>

```

Формат значения периода аналогичен **java.time.Duration**.

Примеры:

- PT15M - 15 минут;
- PT10H - 10 часов;
- P2D - 2 дня;
- P2DT3H4M - 2 дня, 3 часа и 4 минуты.

4.2.9 Задание автоматической блокировки временных УЗ пользователей по окончании периода их действия

Для автоматической блокировки временных учетных записей по окончании периода их действия необходимо отредактировать xml-код объекта **Стандартный сотрудник** (oid = 00000000-0000-0000-0000-0000000000012), добавив следующие строки:

```

    <mapping>
        <name>Запись даты начала работы</name>
        <source>
            <name>validFrom</name>
            <c:path>$focus/workstart</c:path>
        </source>
        <target>

<c:path>$focus/activation/validFrom</c:path>

```

```
        </target>
    </mapping>
    <mapping>
        <name>Запись даты окончания работы</name>
        <source>
            <name>validTo</name>
            <c:path>$focus/workend</c:path>
        </source>
        <target>

<c:path>$focus/activation/validTo</c:path>
        </target>
    </mapping>
```

4.3 Настройка прав доступа (ролей)

4.3.1 Управление доступом к объектам и операциям в ПС

Для управления доступом к объектам и операциям в ПС используется ролевая модель. То есть чтобы иметь возможность выполнять ту или иную операцию над каким-либо объектом, необходима роль, которая явно разрешает такой доступ.

Для управления разрешениями внутри роли используются авторизации (**authorization**). Они позволяют явно указать множество объектов и операции над ними, к которым конкретная авторизация предоставляет доступ. Авторизации перечисляются в роли в виде обычного списка и активируются для конкретного пользователя при его входе в систему. Таким образом, чтобы изменения в настройках управления доступом (изменения авторизаций внутри роли, назначение/отзыв ролей) вступили в силу для конкретного пользователя во время активного сеанса работы с веб-интерфейсом, ему необходимо выйти и повторно войти в систему.

ПС имеет гибкую модель авторизации, которая контролирует доступ пользователей к конкретным страницам пользовательского и административного интерфейсов, а также действия (чтение, редактирование) над конкретными объектами и атрибутами. По умолчанию в ПС применяется политика Запрещено все. Модель авторизации позволяет указать разрешенное: страницы, действия, объекты и атрибуты.

Для задания прав доступа к конкретному разделу интерфейса и/или на определенные действия, выполняемые конкретной ролью в ПС, необходимо в разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории)** выбрать требуемый объект типа **roles** (Абстрактная роль, Роль) и добавить в XML-описание роли соответствующий набор блоков `authorization` (описание авторизаций приведено в разделах 4.3.1.1 и 4.3.1.2). Например:

```
<authorization id="1"> <!-- Доступ на web-страницы
в inrights-ui -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-3#home</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-
3#newRequest</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-
3#newRequestForSelf</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-
3#newRequestForUser</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-
3#incomingRequests</action>
```

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#outgoingRequests</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#users</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#userRoles</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user.rolesWidget</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user.accountsWidget</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user.substituteWidget</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#request</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#error</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#error401</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#error404</action>

```
</authorization>

<authorization id="2"> <!-- Доступ на все web-
страницы в admin-gui -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-3#guiAll</action>

</authorization>

<authorization id="3"> <!-- Просмотр всех
пользователей с атрибутами кроме назначений и
учетных записей -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

  <object>
    <type>UserType</type>
  </object>
  <item>name</item>
  <item>fullName</item>
  <item>givenName</item>
  <item>familyName</item>
  <item>additionalName</item>
  <item>nickName</item>
  <item>title</item>
  <item>honorificPrefix</item>
  <item>honorificSuffix</item>
  <item>preferredLanguage</item>
  <item>locale</item>
  <item>timezone</item>
  <item>emailAddress</item>
  <item>telephoneNumber</item>
  <item>employeeNumber</item>
  <item>employeeType</item>
  <item>costCenter</item>
  <item>organization</item>
  <item>organizationalUnit</item>
```



```
<item>locality</item>
<item>hrStatus</item>
<item>jpegPhoto</item>
<item>result</item>
<item>activation</item>
<item>metadata</item>
<item>parentOrgRef</item>
<item>parentOrg</item>
<item>managerRef</item>

<item>credentials/password/lastSuccessfulLogin</item>
m>
    <item>workstart</item>
    <item>workend</item>
</authorization>
<authorization id="4"> <!-- Просмотр себя с
учетными записями -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>
    <object>
        <special>self</special>
    </object>
    <item>association</item>
<item>assignment</item>
    <item>linkRef</item>
    <item>technologicLinkRef</item>
</authorization>
<authorization id="5"> <!-- Просмотр всех ролей -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>
    <object>
        <type>RoleType</type>
    </object>
```

```
</authorization>

<authorization id="6"> <!-- Просмотр ресурсного
объекта для самого себя -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

  <object>
    <type>ShadowType</type>
    <owner>
      <special>self</special>
    </owner>
  </object>
</authorization>

<authorization id="7"> <!-- просмотр ресурсных
объектов групп -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

  <object>
    <type>ShadowType</type>
    <filter>
      <q:equal>
        <q:path>kind</q:path>
        <q:value>entitlement</q:value>
      </q:equal>
    </filter>
  </object>
</authorization>

<authorization id="8"> <!-- Просмотр всех
подразделений -->

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

  <object>
    <type>OrgType</type>
```

```
</object>
</authorization>
<authorization id="9"> <!-- Просмотр всех ИС -->

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
  <object>
    <type>ResourceType</type>
  </object>
</authorization>
<authorization id="10"> <!-- Любые действия над заявками -->

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#add</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#modify</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#delete</action>
  <object>
    <type>RequestType</type>
  </object>
</authorization>
<authorization id="11"> <!-- Любые действия над заявками по назначениям-->

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#add</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#modify</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#delete</action>
```

```
<object>
```

```
<type>AssignmentsRequestType</type>
```

```
</object>
```

```
</authorization>
```

```
<authorization id="12"> <!-- Права назначать и  
отзывать права-->
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#unassign</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#assign</action>
```

```
<object>
```

```
<type>UserType</type>
```

```
</object>
```

```
<target>
```

```
<type>RoleType</type>
```

```
</target>
```

```
</authorization>
```

```
<authorization id="13"> <!-- Право редактировать  
всех пользователей -->
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#modify</action>
```

```
<phase>execution</phase>
```

```
<object>
```

```
<type>UserType</type>
```

```
</object>
```

```
</authorization>
```

```
<authorization id="14"> <!-- Все права для  
манипуляцией над ресурсными объектами -->
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/pub  
lic/security/authorization-model-3#add</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/pub  
lic/security/authorization-model-3#modify</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/pub  
lic/security/authorization-model-3#delete</action>
```

```
<phase>execution</phase>
```

```
<object>
```

```
<type>ShadowType</type>
```

```
</object>
```

```
</authorization>
```

```
<authorization id="15"> <!-- Разрешение чтения  
объектов, типа Трудоустройство -->
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/pub  
lic/security/authorization-model-3#read</action>
```

```
<object>
```

```
<type>EmploymentType</type>
```

```
</object>
```

```
</authorization>
```

```
<authorization id="16"> <!-- Разрешение чтения  
объектов, типа Должность -->
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/pub  
lic/security/authorization-model-3#read</action>
```

```
<object>
```

```
<type>PositionType</type>
```

```
</object>
```

```
</authorization>
```

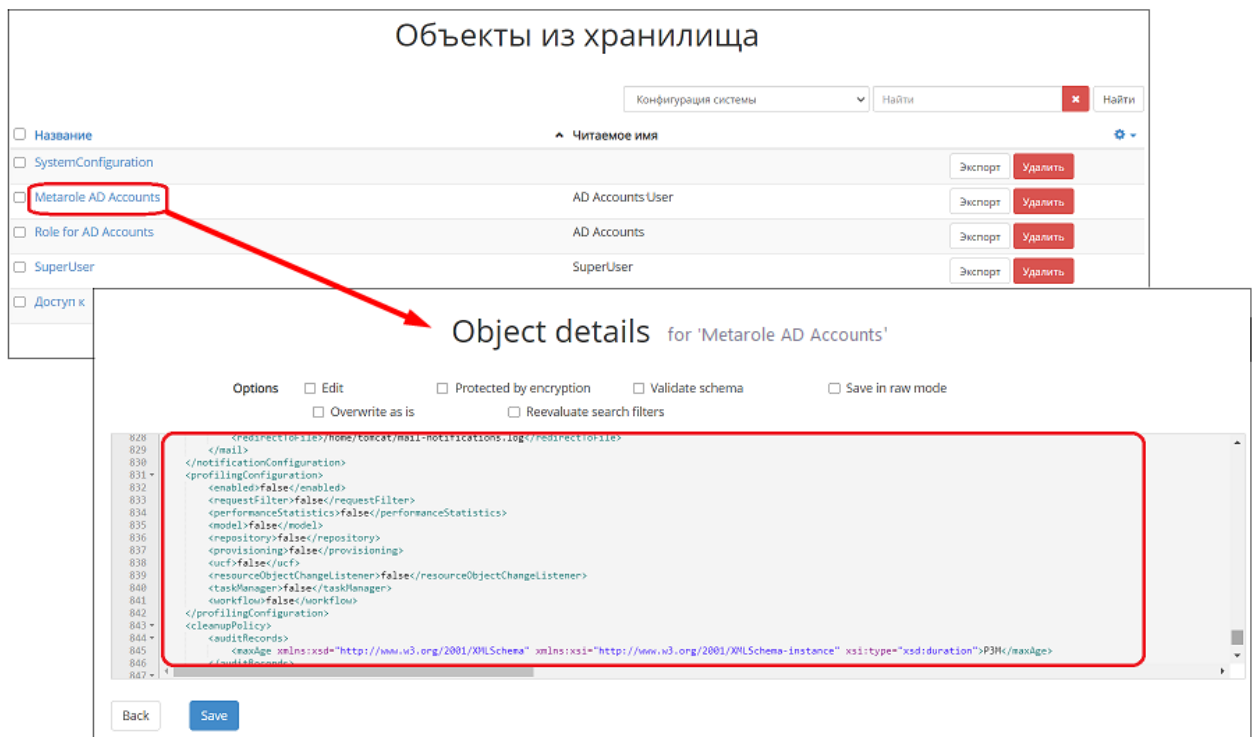


Рис. 77 – Редактирование роли: задание прав доступа

4.3.1.1 Задание доступа к страницам пользовательского интерфейса ПС

В ПС можно разделять доступ к конкретным страницам пользовательского интерфейса, а также к предоставляемым пользователю функциям на этих страницах.

В таблице ниже (см. Таблица 10 –) приведены строки, которые необходимо добавить в XML-описание требуемой роли для организации доступа к страницам пользовательского интерфейса.

Таблица 10 – Задание доступа к страницам пользовательского интерфейса (GUI)

Раздел интерфейса	Описание прав доступа	Авторизация
Все страницы GUI ПС	Доступ ко всей интерфейсной части системы (пользователь может увидеть все страницы разделов ПС)	http://inrights/xml/ns/public/security/authorization-3#guiAll

Все страницы GUI ПС + функции	Доступ к интерфейсной и функциональной части системы (пользователь может не только видеть все страницы разделов, но и выполнять действия в системе)	http://inrights/xml/ns/public/security/authorization-3#all
Главная страница GUI ПС	Доступ к главной странице системы	http://inrights/xml/ns/public/security/authorization-ui-new-3#incomingRequests
Заявки Входящие	- Доступ на просмотр списка входящих заявок	http://inrights/xml/ns/public/security/authorization-ui-new-3#outgoingRequests
Заявки Отправленные	- Доступ на просмотр списка отправленных заявок	http://inrights/xml/ns/public/security/authorization-ui-new-3#newRequest
Заявки Отправить заявку	- Доступна кнопка Отправить заявку (создание новой заявки)	http://inrights/xml/ns/public/security/authorization-ui-new-3#newRequest
Заявки Запросить полномочия для себя	- Доступна кнопка Запросить полномочия для себя	http://inrights/xml/ns/public/security/authorization-ui-new-3#newRequestForSelf
Заявки Запросить полномочия для других	- Доступна кнопка Запросить полномочия для других	http://inrights/xml/ns/public/security/authorization-ui-new-3#newRequestForUser
Заявки Просмотр заявки	- Доступ на просмотр существующей в системе заявки	http://inrights/xml/ns/public/security/authorization-ui-new-3#request
Пользователи Список	- Доступ на просмотр списка пользователей	http://inrights/xml/ns/public/security/authorization-ui-new-3#users
Карточка пользователя*	Доступ на просмотр вкладки	http://inrights/xml/ns/public/security/authorization-ui-new-3#user

	Информация о работнике в карточке пользователя	
Карточка пользователя* Фотография пользователя	- Доступ к функции загрузки фотографии пользователя (необходимо прописать в xml-коде роли, предоставляющей право доступа на редактирование данных пользователя)	http://inrights/xml/ns/public/security/authorization-ui-new-3#userPhotoEdit
Пользователи Добавить пользователя	- Доступна кнопка Добавить пользователя	http://inrights/xml/ns/public/security/authorization-ui-new-3#createUser
Карточка пользователя* Роли	- Доступ на просмотр вкладки Роли в карточке пользователя	http://inrights/xml/ns/public/security/authorization-ui-new-3#user.rolesWidget
Карточка пользователя* Учетные записи	- Доступ на просмотр вкладки Учетные записи в карточке пользователя	http://inrights/xml/ns/public/security/authorization-ui-new-3#user.accountsWidget
Карточка пользователя* Заместители	- Доступ на просмотр вкладки Заместители в карточке пользователя	http://inrights/xml/ns/public/security/authorization-ui-new-3#user.substituteWidget
Карточка пользователя* Трудоустройства	- Доступ на просмотр вкладки Трудоустройства в карточке пользователя	http://inrights/xml/ns/public/security/authorization-ui-new-3#userEmploymentAccess
Карточка пользователя* Трудоустройства - Редактировать	- В карточке пользователя на вкладке Трудоустройства доступны кнопки Редактировать и	http://inrights/xml/ns/public/security/authorization-ui-new-3#userEmploymentEdit

	Добавить трудоустройство	
Роли - Список	Доступ на просмотр линейного списка ролей системы	http://inrights/xml/ns/public/security/authorization-ui-new-3#roles
Карточка роли	Доступ на просмотр карточки роли (список работников с данной ролью)	http://inrights/xml/ns/public/security/authorization-ui-new-3#role
Информационные системы - Список	Доступ на просмотр линейного списка ИС	http://inrights/xml/ns/public/security/authorization-ui-new-3#resources
Версия приложения	Доступ на просмотр информации о версии приложения	http://inrights/xml/ns/public/security/authorization-ui-new-3#settings_version
Ошибка	Отображение ошибки	http://inrights/xml/ns/public/security/authorization-ui-new-3#error
Ошибка 401	Отображение ошибки 401	http://inrights/xml/ns/public/security/authorization-ui-new-3#error401
Ошибка 404	Отображение ошибки 404	http://inrights/xml/ns/public/security/authorization-ui-new-3#error404
Заявка - кнопка Назначить	Доступна кнопка Назначить (для ручного назначения согласующего)	http://inrights/xml/ns/public/security/authorization-request-3#requestApproverManualAssignment
Заявка - кнопка Пересчитать	Доступна кнопка Пересчитать	http://inrights/xml/ns/public/security/authorization-request-3#recomputeApprovalStep
Роли - Каталог ролей	Редактирование каталогов: создание, переименование, удаление каталога	http://inrights/xml/ns/public/security/authorization-ui-new-3#roleCatalogEdit

Карточка роли - кнопка Редактировать	Редактирование карточки роли	http://inrights/xml/ns/public/security/authorization-ui-new-3#roleCardEdit
Отчет - Аудит событий входа	Просмотр и формирование отчета "Аудит событий входа"	http://inrights/xml/ns/public/security/authorization-ui-new-3#reports.authEvents
Отчет - История изменений кадровых данных	Просмотр и формирование отчета "История изменений кадровых данных"	http://inrights/xml/ns/public/security/authorization-ui-new-3#reportsUserHistory
Подключения информационных систем	Доступ к списку информационных систем и их учетным записям (UserGUI)	http://inrights/xml/ns/public/security/authorization-ui-3#resources
Администрирован ие > Подключения ИС	Доступ к пункту меню Администрирован ие > Подключения ИС (UserGUI)	http://inrights/xml/ns/public/security/authorization-ui-new-3#resourcesConnections
Технические страницы - Системные логи	Право на просмотр системных журналов	http://inrights/xml/ns/public/security/authorization-rest-3#technicalLogs

ВАЖНО! (*) Для всех работ с карточкой пользователя также необходим доступ к объекту **FormType**:

```
<authorization>
```

```
<action>http://inrights.ru/xml/ns/public/security/authorization-model-3#read</action>
```

```
<object>
```

```
<type>FormType</type>
```

```
</object>
```

```
</authorization>
```

4.3.1.2 Управление правами доступа на действия в ПС

В ПС можно ограничивать права на действия (чтение, редактирование) с конкретными объектами и атрибутами.

В таблице ниже (см. Таблица 11 –) приведены строки, которые необходимо добавить в XML-описание требуемой роли для организации доступа к действиям над конкретными объектами.

Таблица 11 – Задание доступа к действиям над объектами в ПС

Просмотр		Добавление		Редактирование		Удаление	
Подразделение, отдел (Роль типа Org)							
	<pre><authorization> <action> http://inrights/ xml/ns/public/ security/ authorization- model-3#read </action> <object> <type>OrgType</type> </object> </authorization></pre>		<pre><authorization> <action> http://inrights/ xml/ns/public/ security/ authorization- model-3#add </action> <object> <type>OrgType</type> </object> </authorization></pre>		<pre><authorization> <action> http://inrights/ xml/ns/public/ security/ authorization- model-3#modify </action> <object> <type>OrgType</type> </object> </authorization></pre>		<pre><authorization> <action> http://inrights/ xml/ns/public/ security/ authorization- model-3#delete </action> <object> <type>OrgType</type> </object> </authorization></pre>
Информационная система							
	<pre><authorization> <action></pre>						

	<pre> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#read </action > <object> <type>Re sourceTy pe</type > </object > </author ization> </pre>					
Заявка						
	<pre> <authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#read </action > <object> <type>Re questTyp e</type> </object > </pre>		<pre> <authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#add </action > <object> <type>Re questTyp e</type> </object > </pre>		<pre> <authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#modify </action > <object> <type>Re questTyp e</type> </object > </pre>	<pre> <authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#delete </action > <object> <type>Re questTyp e</type> </object > </pre>

	</author ization>		</author ization>		</author ization>		</author ization>
Пользователь							
	<authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#read </action > <object> <type>Us erType</ type> </object > </author ization>		<authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#add </action > <object> <type>Us erType</ type> </object > </author ization>		<authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#modify </action > <object> <type>Us erType</ type> </object > </author ization>		<authori zation> <action> http://i nrights/ xml/ns/p ublic/se curity/ authoriz ation- model- 3#delete </action > <object> <type>Us erType</ type> </object > </author ization>

4.3.1.3 Управление правами доступа на действия с отчетами и их шаблонами

В ПС можно разделять доступ на действия с отчетами и шаблонами отчетов. При этом, отчеты формируются с учетом прав доступа пользователя, который их запрашивает. Соответственно, в отчет попадает только та информация, к которой у пользователя есть доступ. Эти права не обязательно выдаются вместе с правами на вызов отчета, они могут быть получены посредством назначения других ролей.

4.3.1.3.1 Задание прав доступа на просмотр шаблонов отчетов и формирование отчетов в HTML-формате

В таблице ниже (см. Таблица 12 –) приведены строки, которые необходимо добавить в XML-описание требуемой роли для доступа к встроенным шаблонам отчетов (дают права на просмотр встроенных шаблонов отчетов и формирование на их основе отчетов в HTML-формате):

Таблица 12 – Задание доступа к встроенным шаблонам отчетов

Название шаблона отчетов	Авторизация
Бесхозные учетные записи	<code>http://inrights/xml/ns/public/security/authorization-ui-new-3#unlinkAccounts</code>
Текущий доступ пользователей	<code>http://inrights/xml/ns/public/security/authorization-ui-new-3#userAccess</code>
Доступ к информационным системам	<code>http://inrights/xml/ns/public/security/authorization-ui-new-3#resourceAccess</code>
История изменений ролей работника	<code>http://inrights/xml/ns/public/security/authorization-ui-new-3#rolesHistory</code>
Аудит событий входа	<code>http://inrights/xml/ns/public/security/authorization-ui-new-3#reportsAuthEvents</code>
История изменений кадровых данных	<code>http://inrights/xml/ns/public/security/authorization-ui-new-3#reportsUserHistory</code>

При этом авторизации должны быть прописаны в одной секции `authorization`, например:

```
<authorization id="1">

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#home</action>

    <!-- пропущена часть стандартных
авторизаций -->

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#rolesModel</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#unlinkAccounts</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#userAccess</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#resourceAccess</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#rolesHistory</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#reportsAuthEvents</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#reportsUserHistory</action>
```

```
</authorization>
```

Для доступа к обычным (не встроенным) шаблонам отчетов необходимо в XML-описании требуемой роли в одной секции authorization прописать доступ к соответствующим объектам типа **ReportType**, например:

```
<authorization id="17">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
```

```
<object>
```

```
<type>ReportType</type>
```

```
<filter>
```

```
<q:or>
```

```
<q:equal>
```

```

        <q:path>name</q:path>
        <q:value>История нарушения
политик</q:value>
    </q:equal>
    <q:equal>
        <q:path>name</q:path>
        <q:value>Демо отчет</q:value>
    </q:equal>
    <q:equal>
        <q:path>name</q:path>
        <q:value>Реестр заявок</q:value>
    </q:equal>
    <q:equal>
        <q:path>name</q:path>
        <q:value>Назначения ролей сверх
ролевой модели</q:value>
    </q:equal>
</q:or>
</filter>
</object>
</authorization>

```

4.3.1.3.2 Задание прав доступа для формирования и загрузки построенных отчетов

Для задания прав доступа на формирование и загрузку построенных отчетов по определенной ссылке необходимо добавить в XML-описание требуемой роли строки:

```

<!-- Авторизация для перехода по url для скачивания
отчета -->
<action>

http://inrights.solarsecurity.ru/xml/ns/public/secu
rity/authorization-ui-new-3#report
</action>

```



```
...
<!-- Авторизация для работы с объектом
ReportOutputType -->
<authorization id="3">
    <action>

http://inrights.solarsecurity.ru/xml/ns/public/secu
rity/authorization-model-3#read

    </action>
    <object>
        <type>ReportOutputType</type>
    </object>
</authorization>
```

4.3.1.4 Примеры организации доступа

В разделе приведены примеры организации доступа:

1) Пример конфигурации, которая позволит пользователю авторизоваться в административном интерфейсе ПС и иметь доступ к **главной странице, списку пользователей и страницам пользователей**, а также просматривать атрибуты **name, fullName, familyName** и **title** объектов типа **UserType**:

```
<authorization id="1">

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-3#home</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-3#users</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-ui-new-3#user</action>
</authorization>
<authorization id="3">
    <action>
```

```
http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read
  </action>
  <object>
    <type>UserType</type>
  </object>
  <c:item>name</c:item>
  <c:item>fullName</c:item>
  <c:item>familyName</c:item>
  <c:item>title</c:item>
</authorization>
```

2) Пример конфигурации, которая позволит пользователю ПС подавать заявки на предоставление полномочий:

```
<authorization id="1">

  <action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#home</action>

  <action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#users</action>

  <action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user</action>

  <action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#newRequest</action>

  <action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#newRequestForSelf</action>

  <action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#incomingRequests</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#outgoingRequests</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user.rolesWidget</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#user.accountsWidget</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#request</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#error</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#error401</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#error404</action>
```

```
</authorization>
```

```
<authorization id="2">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
```

```
  <object>
```

```
    <type>UserType</type>
```

```
  </object>
```

```
  <c:item>name</c:item>
```

```
  <c:item>fullName</c:item>
```

```
  <c:item>familyName</c:item>
```

```
  <c:item>title</c:item>
```

```
</authorization>
<authorization id="3">

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

  <object>
    <type>RoleType</type>
  </object>
</authorization>
<authorization id="4">

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#add</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#modify</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#delete</action>

  <object>
    <type>RequestType</type>
  </object>
</authorization>
<authorization id="5">

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#add</action>

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#modify</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#delete</action>
  <object>
    <type>AssignmentsRequestType</type>
  </object>
</authorization>
```

3) Пример конфигурации, которая позволит пользователю ПС создавать и просматривать заявки на себя и на другое лицо:

```
<authorization id="7">

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
  <object>
    <type>RoleType</type>
  </object>
</authorization>

<authorization id="8">

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#add</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#modify</action>

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#delete</action>
  <object>
    <type>RequestType</type>
  </object>
</authorization>

<authorization id="9">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#add</action>
```

```
<object>
```

```
<type>RequestType</type>
```

```
</object>
```

```
</authorization>
```

```
<authorization id="10">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#add</action>
```

```
<object>
```

```
<type>AssignmentsRequestType</type>
```

```
</object>
```

```
</authorization>
```

```
<authorization id="11">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#assign</action>
```

```
<object>
```

```
<type>UserType</type>
```

```
</object>
```

```
<target>
```

```
<type>RoleType</type>
```

```
</target>
```

```
</authorization>
```

```
<authorization id="12">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#modify</action>
```

```
<phase>execution</phase>
<object>
  <type>UserType</type>
</object>
</authorization>
```

4) Пример конфигурации, которая позволит пользователю ПС поменять свой пароль:

```
<authorization id="4">

<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#modify</action>

  <object>
    <special>self</special>
  </object>
  <item>credentials/password</item>
</authorization>
```

5. Пример роли, предоставляющей доступ к шаблонам отчетов:

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"
```

```
    oid="5fa68618-def7-4d42-9f95-9276927ee669"
    version="16">
<name>Отчетная</name>
<metadata>
    <createTimestamp>2016-11-
15T16:41:39.436+03:00</createTimestamp>
    <creatorRef oid="00000000-0000-0000-0000-
000000000002" type="c:UserType"><!-- administrator
--></creatorRef>

<createChannel>http://inrights.solarsecurity.ru/xml
/ns/public/gui/channels-3#user</createChannel>

    <modifyTimestamp>2016-11-
15T16:41:39.569+03:00</modifyTimestamp>

    <modifierRef
xmlns:tns="http://inrights.solarsecurity.ru/xml/ns/
public/common/common-3"
        oid="00000000-0000-0000-0000-
000000000002"
        type="tns:UserType"><!--
administrator --></modifierRef>

<modifyChannel>http://inrights.solarsecurity.ru/xml
/ns/public/gui/channels-3#user</modifyChannel>
    </metadata>
    <activation>

<administrativeStatus>enabled</administrativeStatus
>
    <effectiveStatus>enabled</effectiveStatus>
    <enableTimestamp>2016-11-
15T16:41:39.475+03:00</enableTimestamp>
    </activation>
    <iteration>0</iteration>
    <iterationToken/>
    <displayName>Отчетная</displayName>
    <authorization id="1">
```



```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#rolesModel</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#unlinkAccounts</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#userAccess</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#resourceAccess</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#rolesHistory</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#reportsAuthEvents</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#reportsUserHistory</action>
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-ui-new-3#report</action>
```

```
</authorization>
```

```
<authorization id="16">
```

```
<action>http://inrights.solarsecurity.ru/xml/ns/public/security/authorization-model-3#read</action>
```

```
<object>
```

```
<type>ReportType</type>
```

```
<filter>
```

```
<q:or>
```

```
<q:equal>
    <q:path>name</q:path>
    <q:value>История нарушения
политик</q:value>
</q:equal>
<q:equal>
    <q:path>name</q:path>
    <q:value>Анализ ролевой
модели</q:value>
</q:equal>
<q:equal>
    <q:path>name</q:path>
    <q:value>Текущий доступ
пользователей</q:value>
</q:equal>
<q:equal>
    <q:path>name</q:path>
    <q:value>Доступ к информационным
системам</q:value>
</q:equal>
<q:equal>
    <q:path>name</q:path>
    <q:value>Бесхозные учетные
записи</q:value>
</q:equal>
<q:equal>
    <q:path>name</q:path>
    <q:value>История изменений ролей
сотрудника</q:value>
</q:equal>
<q:equal>
    <q:path>name</q:path>
    <q:value>Аудит событий
входа</q:value>
</q:equal>
```

```

        <q:equal>
            <q:path>name</q:path>
            <q:value>История изменения
кадровых данных</q:value>
        </q:equal>
        <q:equal>
            <q:path>name</q:path>
            <q:value>Демо отчет</q:value>
        </q:equal>
        <q:equal>
            <q:path>name</q:path>
            <q:value>Реестр заявок</q:value>
        </q:equal>
        <q:equal>
            <q:path>name</q:path>
            <q:value>Назначения ролей сверх
ролевой модели</q:value>
        </q:equal>
    </q:or>
</filter>
</object>
</authorization>
<authorization id="3">

<action>http://inrights.solarsecurity.ru/xml/ns/pub
lic/security/authorization-model-3#read</action>
    <object>
        <type>ReportOutputType</type>
    </object>
</authorization>
<requestable>true</requestable>
</role>

```

4.3.1.5 Ограничение видимости объектов на основании организационно штатной структуры

В случае наличия в организации потенциальных нарушителей, работающих в разных подразделениях организации и желающих скрытно обмениваться информацией с использованием ПС, объекты Подразделение и Пользователь могут использоваться как скрытый канал передачи данных. Также для этих целей могут использоваться комментарии к заявкам.

Для нейтрализации данного скрытого канала необходимо настраивать доступ к объектам ПС таким образом, чтобы пользователи не могли видеть карточки пользователей, подразделений и заявок других департаментов. Особенно это касается департаментов, работники которых работают со сведениями, составляющими коммерческую тайну, и другими видами информации ограниченного доступа.

Для предотвращения доступа к карточкам работников других подразделений необходимо создать роли, предоставляющие доступ к объектам типа **UserType**, **OrgType**, **RequestType** для каждого подразделения определенного уровня (решение до какого уровня подразделений необходимо создавать роли принимается на этапе внедрения, исходя из требований политики ИБ организации). Данные роли должны быть назначены как базовые на соответствующие подразделения (см. раздел 4.3.5).

4.3.2 Настройка маршрута согласования заявок на предоставление доступа

4.3.2.1 Задание маршрута согласования заявок на предоставление доступа

Для задания маршрута согласования заявок на предоставление доступа необходимо:

1) В разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории)** выбрать требуемый объект типа **roles** (Абстрактная роль, Роль).

2) Отредактировать XML-код, добавив блок approvalSchema.
Например:

```
<approvalSchema>
  <level>
    <name>Согласование у руководителя</name>

    <approverDefinition>manager</approverDefinition>
  </level>
  <level>
    <name>Согласование у владельца роли</name>

    <approverDefinition>roleOwner</approverDefinition>
  </level>
  <level>
    <name>Согласование у офицера
безопасности</name>
    <approverDefinition>role[oid=7bd5a0fe-90eb-
40ce-b311-b8f95132bd5a]</approverDefinition>
  </level>
  <level>
    <name>Согласование у пользователя</name>
    <approverRef
xmlns:c="http://midpoint.evolveum.com/xml/ns/public
/common/common-3"
      oid="75f2806d-e31b-40c9-8133-
85ed4d9e6252"
      type="c:UserType">
      <description>Big boss #1
approval</description>
    </approverRef>
  </level>
```

</approvalSchema>

В примере настроен следующий маршрут:

- Согласование у руководителя пользователя, для которого запрашиваются права доступа.
- Согласование у владельца роли, на которую запрашиваются права.
- Согласование у одного из пользователей, которым назначена роль с oid="75f2806d-e31b-40c9-8133-85ed4d9e6252".
- Согласование у конкретного пользователя.

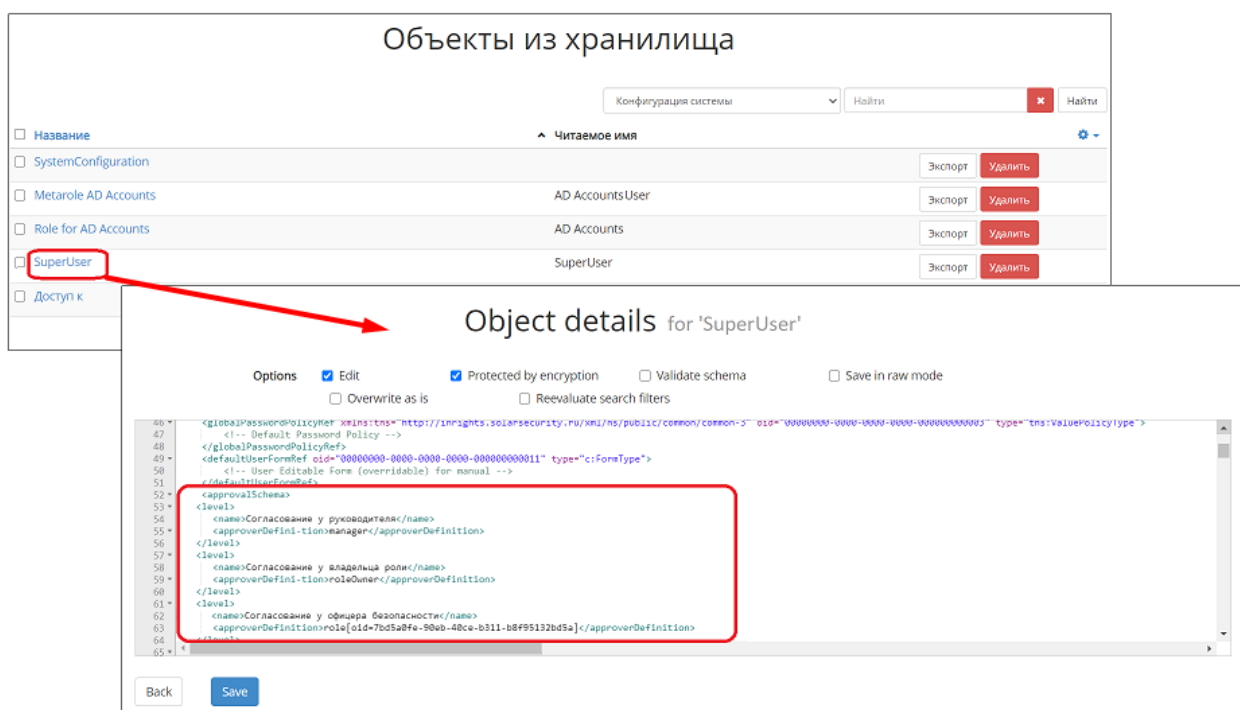


Рис. 78 – Редактирование роли: задание маршрута согласования заявки на предоставление прав доступа

ВАЖНО! Маршрут согласования можно задать / настроить только с помощью редактирования XML-кода соответствующего объекта.

Система позволяет настроить маршрут согласования в зависимости от уровня риска, категории пользователя или наличия SOD-конфликта.

Например:

```
<approvalSchemaObject>
  <name>Запрос доступов</name>
  <description/>
  <type>approval</type>
  <level>
    <name>Согласование руководителя</name>
    <approverExpression>
      <description>Руководитель
бенефициара</description>
      <script>
        <code>
          import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

          import
ru.solarsecurity.inrights.ra.util.request.enums.Sel
ectMode

          return new ApproverBuilder()
            getFinder(approvalStepContext)
            searchFunctionalManager()
            searchLineManager()
            searchUpperLineManager()
            find(SelectMode.FirstOf)
        </code>
      </script>
    </approverExpression>
  </level>
  <level>
    <name>Согласование Представителем СБ</name>
    <approverExpression>
      <description>Представитель СБ
организации</description>
```

```
<script>
    <code>
        import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

        import
ru.solarsecurity.inrights.ra.util.request.enums.Sel
ectMode

        return new ApproverBuilder()
        getFinder(approvalStepContext)
        searchUserWithRole("-id-role-
org-security")

        find(SelectMode.FirstOf)
    </code>
</script>
</approverExpression>
<condition>
    <description>Категория пользователя Б /
B</description>
    <script>
        <code>
            import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

            import
ru.solarsecurity.inrights.ra.util.request.enums.Con
ditionMode

            return new ApproverBuilder()

getFinder(approvalProcessContext)
                isCorrectUserCategory(["Б",
"B"])

computeConditions(ConditionMode.Or)
```



```
</code>
</script>
</condition>
</level>
<level>
  <name>Согласование работником ДЗГТИ</name>
  <approverExpression>
    <description>Работник ДЗГТИ
организации</description>
    <script>
      <code>
        import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

        import
ru.solarsecurity.inrights.ra.util.request.enums.Sel
ectMode

        return new ApproverBuilder()
        getFinder(approvalStepContext)

        .searchUserWithRole("-ID-Role-Employee-DZGTI")
        find(SelectMode.FirstOf)
      </code>
    </script>
  </approverExpression>
  <condition>
    <description>Бенефициар - иностранный
гражданин</description>
    <script>
      <code>
        import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder
```

```
import
ru.solarsecurity.inrights.ra.util.request.enums.ConditionMode

return new ApproverBuilder()

getFinder(approvalProcessContext)
isForeignCitizen()

computeConditions(ConditionMode.Or)
</code>
</script>
</condition>
</level>
<level>
<name>Согласование владельцем роли</name>
<approverExpression>
<description>Владелец
роли</description>
<script>
<code>
import
ru.solarsecurity.inrights.ra.util.request.ApproverBuilder

import
ru.solarsecurity.inrights.ra.util.request.enums.SelectMode

return new ApproverBuilder()
getFinder(approvalStepContext)

.searchRoleOwner()

find(SelectMode.FirstOf)
</code>
</script>
</approverExpression>
```

```
<condition>
    <description>У роли есть
владелец</description>
    <script>
        <code>
            import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

            import
ru.solarsecurity.inrights.ra.util.request.enums.Con
ditionMode

            return new ApproverBuilder()

getFinder(approvalProcessContext)
            isRoleRequested()

computeConditions (ConditionMode.Or)
        </code>
    </script>
</condition>
</level>
<level>
    <name>Согласование владельцем бизнес-
роли</name>
    <approverExpression>
        <description>Владелец бизнес-
роли</description>
        <script>
            <code>
                import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

                import
ru.solarsecurity.inrights.ra.util.request.enums.Sel
ectMode
```

```
        return new ApproverBuilder()
            getFinder(approvalStepContext)
            searchBusinessRoleOwner()
            find(SelectMode.FirstOf)
    </code>
</script>
</approverExpression>
<condition>
    <description>У бизнес-роли есть
владелец</description>
    <script>
        <code>
            import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

            import
ru.solarsecurity.inrights.ra.util.request.enums.Con
ditionMode

            return new ApproverBuilder()

getFinder(approvalProcessContext)

            isBusinessRoleHasOwner(true)

computeConditions(ConditionMode.Or)
        </code>
    </script>
</condition>
</level>
<level>
    <name>Согласование владельцем ИТ-
ресурса</name>
    <approverExpression>
```

<description>Владелец ИТ-ресурса</description>

<script>

<code>

*import
ru.solarsecurity.inrights.ra.util.request.ApproverBuilder*

*import
ru.solarsecurity.inrights.ra.util.request.enums.SelectMode*

*return new ApproverBuilder()
getFinder(approvalStepContext)
searchResourceOwner()
find(SelectMode.FirstOf)*

</code>

</script>

</approverExpression>

<condition>

*<description>У логической системы есть
владелец</description>*

<script>

<code>

*import
ru.solarsecurity.inrights.ra.util.request.ApproverBuilder*

*import
ru.solarsecurity.inrights.ra.util.request.enums.ConditionMode*

return new ApproverBuilder()

getFinder(approvalProcessContext)

isBusinessRoleHasOwner(false)

computeConditions(ConditionMode.Or)

```
</code>
</script>
</condition>
</level>
<level>
  <name>Согласование администратором ИБ ИТ-
ресурса</name>
  <approverExpression>
    <description>Администратор ИБ ИТ-
ресурса</description>
    <script>
      <code>
        import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder

        import
ru.solarsecurity.inrights.ra.util.request.enums.Sel
ectMode

        return new ApproverBuilder()
        getFinder(approvalStepContext)
        searchUserWithRole("-ID-Role-
Administrator-IS")

        find(SelectMode.FirstOf)
      </code>
    </script>
  </approverExpression>
  <condition>
    <description>В заявке есть SoD-конфликт
И/ИЛИ в заявке есть роли с риском больше
0</description>
    <script>
      <code>
        import
ru.solarsecurity.inrights.ra.util.request.ApproverB
uilder
```

```

import
ru.solarsecurity.inrights.ra.util.request.enums.ConditionMode

return new ApproverBuilder()

getFinder(approvalProcessContext)
isRoleHasSoD()
isRoleHasRisk()

computeConditions(ConditionMode.Or)
</code>
</script>
</condition>
</level>
</approvalSchemaObject>

```

4.3.2.2 Настройка тайм-аута в маршруте согласования

Для указания тайм-аута шага согласования необходимо внутри элемента `<level></level>` задать:

```

<timeout>
    <value>%d%h%m</value>
</timeout>

```

где % - целые неотрицательные числа. Например, 3d2h1m - 3 дня 2 часа и 1 минута.

Параметры можно опускать, но нельзя менять местами (2d30m - корректно, 5h1d – нет).

Можно указать множество элементов `<timeout></timeout>`, которые будут применяться один за другим (временной интервал каждого следующего тайм-аута будет применяться ко времени истечения предыдущего).

4.3.2.3 Настройка эскалации в маршруте согласования

Чтобы по истечению тайм-аута произошла эскалация шага согласования другому пользователю, необходимо внутри тега **<timeout/>** добавить тег **<action/>**, внутри которого указать пользователя, которому будет эскалирован следующий шаг согласования.

Допустимые элементы идентичны тем, что используются для указания согласующего на шаге согласования (например, **<approverRef></approverRef>**, **<approverDefinition></approverDefinition>**).

Пример:

```
<timeout>
    ...
    <action
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="c:EscalationTimeoutActionType">

    </action>
    ...
</timeout>
```

4.3.2.4 Задание маршрута согласования: использование groovy-скриптов

При задании маршрута согласования в целях настройки поведения системы при определении пользователя, согласующего маршрут, возможно использовать groovy-скрипты, аналогичные скриптам конфигурации, внедряемым в xml-документы конфигурации.

В качестве исходных данных в скрипте рекомендуется использовать объекты, доступные из контекста шага согласования (доступен через переменную **approvalStepContext** (ru.inrights.requests.impl.contexts.ApprovalStepContext)).

В состав скрипта можно включать классы и методы из публичного API ПС, который релизован и доступен через переменную ПС - набор утилитарных функций (**ru.inrights.model.api.expr.InrightsFunctions**).

В результате выполнения скрипт должен вернуть информацию об идентификаторе (oid) согласующего.

При определении согласующего на шаге его динамического вычисления в маршруте вместо тега `<approverDefinition>` необходимо использовать тег **`<approverExpression>`**. Внутри тега **`approverExpression`** необходимо задать groovy-скрипт, вычисляющий oid согласующего.

Пример роли, используемой в дескрипторах маршрутов:

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

    oid="4aee52ad-bdea-4c99-a27b-e9f950c7c163"
    version="1">
    <name>Роль кастомная: эскалация</name>
    <description>Роль кастомная:
эскалация</description>
    <metadata>
```

```
<createTimestamp>2016-10-
03T16:36:27.016+03:00</createTimestamp>
    <creatorRef oid="00000000-0000-0000-0000-
000000000002" type="c:UserType"><!-- administrator
--></creatorRef>

<createChannel>http://inrights.solarsecurity.ru/xml
/ns/public/gui/channels-3#user</createChannel>
    <modifyTimestamp>2016-10-
03T16:36:27.297+03:00</modifyTimestamp>
    <modifierRef
xmlns:tns="http://inrights.solarsecurity.ru/xml/ns/
public/common/common-3"
        oid="00000000-0000-0000-0000-
000000000002"
        type="tns:UserType"><!--
administrator --></modifierRef>

<modifyChannel>http://inrights.solarsecurity.ru/xml
/ns/public/gui/channels-3#user</modifyChannel>
    </metadata>
    <activation>

<administrativeStatus>enabled</administrativeStatus
>
    <effectiveStatus>enabled</effectiveStatus>
    <validFrom>2016-09-
30T23:00:00.000+03:00</validFrom>
    <validityStatus>in</validityStatus>
    <enableTimestamp>2016-10-
03T16:36:27.064+03:00</enableTimestamp>
    <validityChangeTimestamp>2016-10-
03T16:36:26.815+03:00</validityChangeTimestamp>
    </activation>
    <iteration>0</iteration>
    <iterationToken/>
    <displayName>Роль кастомная
эскалация</displayName>
```

```
<requestable>true</requestable>
<ownerRef type="c:UserType">
  <filter>
    <q:equal>
      <q:path>c:name</q:path>
      <q:value>muzalevskii.v.a</q:value>
    </q:equal>
  </filter>
</ownerRef>
</role>
```

Пример скрипта, определяющего руководителя для руководителя
пользователя:

```
<approvalSchemaObject

xmlns="http://inrights.solarsecurity.ru/xml/ns/publ
ic/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/publi
c/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/pu
blic/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/publi
c/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/p
ublic/resource/instance-3"

                                oid="4d656214-fb2e-4763-94e6-
0b9b2d649034"

                                version="35">
```

```

    <name>Маршрут согласования руководителем
руководителя (скрипт вычисляющий руководителя
руководителя) </name>
    <level>
        <name>Согласование у руководителя</name>

<approverDefinition>manager</approverDefinition>
    </level>
    <level>
        <name>Согласование руководителем
руководителя</name>
        <approverExpression>
            <description>Скрипт вычисляющий
руководителя руководителя</description>
            <script>
                <code>
                    import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

                    import
ru.solarsecurity.inrights.requests.impl.AssignmentR
equestHelper;

                    import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.UserType;

                    import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.RequestItemType;

                    RequestItemType requestItem =
approvalStepContext.getParentContext().getRequestIt
em();

                    AssignmentRequestHelper
assignmentRequestHelper =
SpringApplicationContextHolder.getUniqueBeanForClas
s(AssignmentRequestHelper.class);

```

```
String requesteeOid =  
assignmentRequestHelper.getRequesteeOid(requestItem  
);  
  
String managerOid =  
inrights.getUserByOid(requesteeOid)?.getManagerRef(  
)?.getOid();  
  
String managerManagerOid =  
inrights.getUserByOid(managerOid)?.getManagerRef()?.  
getOid();
```

```
</code>  
</script>  
</approverExpression>  
</level>  
<selector>  
  <roleRef oid="4aee52ad-bdea-4c99-a27b-  
e9f950c7c163" type="c:RoleType"></roleRef>  
</selector>  
</approvalSchemaObject>
```

Пример скрипта, вычисляющего путь эскалации:

```
<approvalSchemaObject
```

```
xmlns="http://inrights.solarsecurity.ru/xml/ns/publ  
ic/common/common-3"
```

```
xmlns:q="http://prism.solarsecurity.ru/xml/ns/publi  
c/query-3"
```

```
xmlns:c="http://inrights.solarsecurity.ru/xml/ns/pu  
blic/common/common-3"
```

```
xmlns:t="http://prism.solarsecurity.ru/xml/ns/publi  
c/types-3"
```

```
xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns  
/public/connector/icf-1/resource-schema-3"
```

```
xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/p  
ublic/resource/instance-3"
```

oid="4d656214-fb2e-4763-94e6-0b9b2d649034"

version="35">

<name>Маршрут согласования с кастомной
эскалацией на суперадминистратора</name>

<level>

<name>Согласование у руководителя</name>

<approverDefinition>manager</approverDefinition>

</level>

<level>

<name>Кастомная эскалация на
суперадминистратора</name>

<approverRef oid="eb4a80fa-7679-45d2-9964-1d14589b4874" type="c:UserType">

<!-- muzalevskii.v.a -->

<description>Approval by timeout script
escalation</description>

</approverRef>

<timeout>

<value>1m</value>

<action

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="c:EscalationTimeoutActionType">

<approverExpression>

<description>Get first user
with "Superuser role"</description>

<script>

<code>

import
ru.solarsecurity.inrights.prism.query.RefFilter;

import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.AssignmentObjectType;

```
import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

import
ru.solarsecurity.inrights.prism.PrismContext;

import
ru.solarsecurity.inrights.task.api.TaskManager;

import
ru.solarsecurity.inrights.repo.sql.SqlRepositorySer
viceImpl;

import
ru.solarsecurity.inrights.prism.query.ObjectQuery;

import
ru.solarsecurity.inrights.prism.PrismObject;

def prismContext =
SpringApplicationContextHolder.getUniqueBeanForClas
s(PrismContext.class);

def taskManager =
SpringApplicationContextHolder.getUniqueBeanForClas
s(TaskManager.class);

def repositoryService =
SpringApplicationContextHolder.getUniqueBeanForClas
s
(SqlRepositoryServiceImpl.class);

RefFilter
superUserFilter = RefFilter.createReferenceEqual
(AssignmentObjectType.F_OBJECT_REF,
AssignmentObjectType.class, prismContext,
"00000000-0000-0000-0000-000000000004");

PrismObject
superUserAssignmentPrism =
repositoryService.searchObjects
(AssignmentObjectType.class,
ObjectQuery.createObjectQuery(superUserFilter),
null,

taskManager.createTaskInstance().getResult())?.get(
0);
```

```

return
superUserAssignmentPrism?.asObjectable()?.getSubjectRef()?.getOid();
    </code>
    </script>
    </approverExpression>
    </action>
    </timeout>
    </level>
    <selector>
        <roleRef oid="4aee52ad-bdea-4c99-a27b-e9f950c7c163" type="c:RoleType"></roleRef>
    </selector>
</approvalSchemaObject>

```

Пример вычисления эскалации на руководителя руководителя
после эскалации от владельца роли:

```

<approvalSchemaObject
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

                                oid="4d656214-fb2e-4763-94e6-0b9b2d649035"

                                version="36">

```



```
<name>Маршрут согласования владельцем роли с
кастомной эскалацией на руководителя
руководителя</name>
<level>
  <name>Согласование у руководителя</name>

<approverDefinition>manager</approverDefinition>
</level>
<level>
  <name>Кастомная эскалация на руководителя
руководителя</name>

<approverDefinition>roleOwner</approverDefinition>
  <timeout>
    <value>1m</value>
    <action
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="c:EscalationTimeoutActionType">
      <approverExpression>
        <description>Получаем
руководителя-руководителя</description>
        <script>
          <code>
            import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

            import
ru.solarsecurity.inrights.requests.impl.AssignmentR
equestHelper;

            import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.UserType;

            import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.RequestItemType;
```

```

RequestItemType
requestItem =
approvalStepContext.getParentContext().getRequestItem();

AssignmentRequestHelper
assignmentRequestHelper =

SpringApplicationContextHolder.getUniqueBeanForClass(
AssignmentRequestHelper.class);

String requesteeOid =
assignmentRequestHelper.getRequesteeOid(requestItem);

String managerOid =
inrights.getUserByOid(requesteeOid)?.getManagerRef()
?.getOid();

String
managerManagerOid =
inrights.getUserByOid(managerOid)?.getManagerRef()
?.getOid();

return
managerManagerOid;
</code>
</script>
</approverExpression>
</action>
</timeout>
</level>
<selector>
    <roleRef oid="4aee52ad-bdea-4c99-a27b-
e9f950c7c163" type="c:RoleType"></roleRef>
</selector>
</approvalSchemaObject>

```

4.3.2.5 Настройка селектора ролей в маршруте согласования

Селектор ролей представляет собой правило, в котором задается, к каким ролям будет применяться конкретный маршрут согласования. Параметры селектора ролей указываются в XML-

описании маршрута согласования (approvalSchemaObject) внутри секции <selector>. Могут быть указаны следующие параметры:

- **roleRef** – ссылка на конкретную роль, к которой будет применяться маршрут согласования. Можно указывать несколько **roleRef**, например:

```
<selector>
  <roleRef oid="oid-role-boss"
type="c:RoleType"></roleRef>
  <roleRef oid="oid-role-manager"
type="c:RoleType"></roleRef>
  <roleRef oid="oid-role-employee"
type="c:RoleType"></roleRef>
</selector>
```

- **requestItemActionUri** – URI типа заявок, к которым будет применяться маршрут согласования. Можно указывать несколько **requestItemActionUri**. Пример селектора с **requestItemActionUri**:

```
<selector>
  <requestItemActionUri>

http://inrights.solarsecurity.ru/xml/ns/public/requ
est/actions#createRoleAssignment

  </requestItemActionUri>
</selector>
```

- **condition** – условие (обычно скриптовое выражение) применимости маршрута согласования к конкретной роли. В контексте выражения доступны следующие переменные:

- о **actionUri** – URI типа заявки. Для заявок на назначение ролей **actionUri** = http://inrights/xml/ns/public/request/actions#createRoleAssignment

- о **args** – аргументы согласования. Аргументы имеют свойства **object** (ссылка на объект согласования, например, на роль), **subject** (ссылка на субъект согласования, например, на пользователя).

Пример задания условия на роль, OID которой = **oid-role-boss**:

```
<selector>
  <condition>
    <script>
      <code>
        import javax.xml.namespace.*;
        import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

        return args.getValue(new
QName(args.uri, "object")).oid.equals("oid-role-
boss");
      </code>
    </script>
  </condition>
</selector>
```

4.3.3 Назначение контекстных ролей

Контекстная роль – это роль, которая будет назначена пользователю автоматически при назначении ему другой роли. Например, при назначении роли **Superuser** пользователю автоматически назначаются роли **Admin** и **User**. При этом отмена назначения роли **Superuser** не отменяет назначение контекстных ролей **Admin** и **User**.

Для назначения контекстных ролей необходимо:

1) 1. В разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории)** выбрать требуемый объект типа **roles** (Абстрактная роль, Роль).

2) Отредактировать XML-код объекта.

Пример назначения контекстной роли:

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"
    oid="044ba6a8-d4fa-428d-86a5-c3e9765f5323"
    version="2">
    <name>10106</name>
    ....
    <displayName>Менеджер по продажам</displayName>
    <contextAssignments>
        <assignment>
            <targetRef oid="00797cd7-4b45-4a08-9113-c997d33e65a5" type="c:RoleType"/>
        </assignment>
    </contextAssignments>
</role>
```

В данном примере пользователю назначаются:

- роль – должность менеджера по продажам;
- контекстная роль, идентификатор (**oid**) которой **00797cd7-**

4b45-4a08-9113-c997d33e65a5.

Получить сведения об **oid** требуемой роли можно, выбрав соответствующий объект в разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории)**.

4.3.4 Назначение прав доступа в контексте должностей (работа с объектом Трудоустройство)

При назначении прав доступа пользователю (работнику) необходимо учитывать его должность. Каждое назначение пользователя на какую-либо позицию в ОШС приводит к созданию нового связанного с ним объекта - трудоустройства.

В общем случае объект типа **Трудоустройство (Employment)** представляет собой связь **Подразделение - Должность**. У каждого такого объекта имеется набор атрибутов для хранения кадровой информации (дата приема/увольнения, руководитель, подразделение, должность и т.п.). На **Трудоустройство** можно назначать организации, роли и должности. При этом в описании любого объекта типа **Трудоустройство** должна быть ссылка на пользователя-владельца УЗ (в параметре **parentRef**).

Пример связи объекта **Трудоустройство** с объектом **Пользователь** (пример заполнения поля **parentRef**):

```
<mapping>
    <source>
        <name>personUniqueId</name>

    <c:path>$focus/extension/person_unq_id</c:path>
    </source>
    <expression>
        <script>
            <code>

import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;
```

```
import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
import ru.solarsecurity.inrights.prism.*;
import ru.solarsecurity.inrights.prism.delta.*;
import
ru.solarsecurity.prism.xml.ns._public.types_3.*;
import ru.solarsecurity.inrights.schema.result.*;
import ru.solarsecurity.inrights.model.api.*;
import ru.solarsecurity.inrights.repo.api.*;
import ru.solarsecurity.inrights.repo.sql.*;
import ru.solarsecurity.inrights.task.api.*;
import ru.solarsecurity.inrights.requests.api.*;
import
ru.solarsecurity.inrights.prism.PrismContext;
import
ru.solarsecurity.inrights.prism.path.ItemPath;
import
ru.solarsecurity.inrights.repo.api.RepositoryServic
e;
import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;
import ru.solarsecurity.inrights.util.exception.*;
import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
import ru.solarsecurity.inrights.prism.query.*;
import javax.xml.namespace.QName;

def prismContext =
SpringApplicationContextHolder.getUniqueBeanForClas
s(PrismContext.class);
def repositoryService =
SpringApplicationContextHolder.getUniqueBeanForClas
s(SqlRepositoryServiceImpl.class);
def filter = EqualFilter.createEqual
```

```

        (new ItemPath(UserType.F_EXTENSION, new
QName("person_unq_id")), UserType.class,
prismContext, personUniqueId);
def paging = ObjectPaging.createPaging(0, 2);
def searchResult = repositoryService.searchObjects
        (UserType.class,
ObjectQuery.createObjectQuery(filter, paging),
null, new OperationResult("1"));
if(searchResult.size() == 0) {
    return null;
} else if(searchResult.size() == 1) {
    def foundUser = searchResult.get(0);
    ObjectReferenceType parentRef = new
ObjectReferenceType();
    parentRef.setOid(foundUser.getOid());
    parentRef.setType(UserType.COMPLEX_TYPE);
    return parentRef;
} else {
    //нашли более одного пользователя.
Исключительная ситуация
    throw new RuntimeException("found more than one
user to be a parent of employment: " +
personUniqueId);
}

</code>
</script>
</expression>
<target>
    <c:path>$focus/parentRef</c:path>
</target>
</mapping>

```

ВАЖНО! Владелец учетных записей всегда является пользователем. Исключение составляет только случай, когда

Трудоустройство является владельцем собственной учетной записи (например, аккаунт трудоустройства из HR).

Пример:

В системе имеются следующие сущности:

Роль1, предоставляющая права доступа **Полномочие1** в Active Directory, и **Роль2**, предоставляющая права доступа **Полномочие2** в Active Directory.

Пользователь и два связанных с ним трудоустройства: **Трудоустройство1** и **Трудоустройство2**.

Если объекту **Трудоустройство1** назначить **Роль1**, то у Пользователя появится учетная запись в Active Directory, имеющая права доступа **Полномочие1**.

Если объекту **Трудоустройство2** назначить **Роль2**, то учетная запись в Active Directory, владельцем которой является **Пользователь**, получает права доступа **Полномочие2**.

При работе с разными объектами в маппингах информационных систем в одних и тех же параметрах должны быть указаны разные значения:

Таблица 13 – Параметры при работе с разными объектами

Параметр (свойство)	Значение при работе с трудоустройствами (система сконфигурирована на работу с трудоустройствами)	Значение при работе с пользователями
\$focus	Трудоустройство	Пользователь
\$user	Трудоустройство	Пользователь
\$root	Пользователь - владелец трудоустройства	Пользователь
\$owner	Объект, который непосредственно связан с учётной записью	

Соответственно, при настройке ресурса Active Directory в разделе **schemaHandling** для обращения к атрибутам пользователя необходимо указывать объект **root**.

Пример маппинга, составляющего имя учетной записи в Active Directory на основе имени и фамилии пользователя-владельца.

```
<attribute>
  <c:ref
xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/resource-schema-3">
    icfs:name
  </c:ref>
  <outbound>
    <source>
      <c:path>$root/givenName</c:path>
    </source>
    <source>
      <c:path>$root/familyName</c:path>
    </source>
    <expression>
      <script>
        <code>'cn=' + givenName + ' ' +
familyName + iterationToken + ', OU=Test, DC=test,
DC=ru'</code>
      </script>
    </expression>
  </outbound>
</attribute>
```

4.3.5 Настройка базовых ролей

Базовая роль – это роль, назначенная на узел ОШС (подразделение или должность). При назначении (приеме/перевode) работника на должность ему присваиваются соответствующие базовые

роли (относящиеся как к указанной должности, так и к подразделению, с которым связана эта должность).

Например, в компании имеется **Отдел разработки ПО**, в котором, в свою очередь, имеются **Отдел анализа** и **Отдел тестирования**. **Отдел разработки ПО** связан с базовой ролью **Безлимитный Интернет**, предоставляющей всем работникам этого отдела доступ в Интернет.

Отдел анализа связан с базовой ролью **Аналитический раздел**, предоставляющей всем работникам этого отдела доступ к каталогу с аналитическими материалами.

Отдел тестирования связан с базовой ролью **Раздел для автотестов**, предоставляющей всем работникам этого отдела доступ к каталогу, в котором хранятся специальные программы для тестирования ПО.

В отделе тестирования имеются должности **Начальник отдела анализа** и **Аналитик**, в отделе тестирования – должности **Начальник отдела тестирования** и **Тестировщик**. **Начальник отдела тестирования** и **Начальник отдела анализа** связаны с базовой ролью **Руководство**, предоставляющей доступ к планам и другим управленческим материалам.

В результате соответствующие работники отдела разработки ПО будут иметь следующие права доступа:

Таблица 14 – Права доступа

Отдел / должность	Базовая роль	Права доступа
Отдел разработки ПО	Безлимитный интернет	Доступ в Интернет
Отдел анализа	Аналитический раздел	Доступ в Интернет, доступ к каталогу с аналитическими материалами

Начальник анализа	отдела	Руководство	Доступ в Интернет, доступ к каталогу с аналитическими материалами, доступ к планам и другим управленческим материалам
Аналитик			Доступ в Интернет, доступ к каталогу с аналитическими материалами
Отдел тестирования		Раздел для автотестов	Доступ в Интернет, доступ к каталогу с программами для тестирования ПО
Начальник тестирования	отдела	Руководство	Доступ в Интернет, доступ к каталогу с программами для тестирования ПО, доступ к планам и другим управленческим материалам
Тестировщик			Доступ в Интернет, доступ к каталогу с программами для тестирования ПО

4.3.5.1 Назначение базовых ролей

Для назначения объекту базовых ролей в AGUI необходимо в XML-описание соответствующего объекта добавить строки:

```
<contextAssignments>
  <assignment>
    <targetRef oid="54e5c6ec-4e8d-4450-ab80-
d7a977ad32e1" type="c:RoleType"/>
    <request>false</request>
  </assignment>
</contextAssignments>
```

где *oid* – идентификатор требуемой роли.

Примечание

Получить сведения об **oid** роли можно, выбрав соответствующий объект в разделе **Объекты из хранилища (Конфигурирование > Объекты в репозитории)**.

4.3.5.2 Отзыв базовых ролей при переводе на другую должность / в другое подразделение

После перевода работника на другую должность и/или в другое подразделение в дополнение к имеющимся ролям назначаются базовые роли, соответствующие новой должности или подразделению. То есть роли, соответствующие прежней должности или подразделению, автоматически не отзываются, поскольку возможны различные варианты: отозвать все роли, не отзывать ролей или отозвать старые роли и присвоить новые.

В случае необходимости отзыва старых и назначения новых базовых ролей можно воспользоваться скриптом. Для этого в AGUI в разделе **Конфигурирование > Объекты в репозитории > Конфигурация системы** в XML-описании объекта **SystemConfiguration** в секции **modelHooks/change/hook/** необходимо ввести код требуемого скрипта.

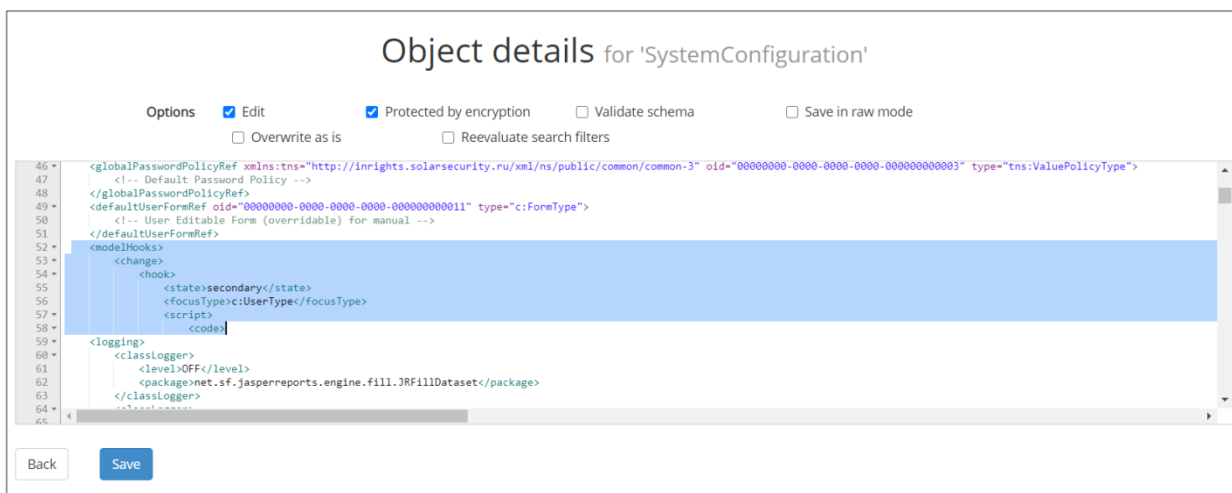


Рис. 79 – XML-описание объекта SystemConfiguration: секция для ввода кода скрипта

Пример скрипта, используемого для отзыва старых и назначения новых ролей при переводе на другую должность:

```
//скрипт для отзыва старых и назначения новых ролей
при переводе на другую должность
<modelHooks>
    <change>
        <hook>
            <state>secondary</state>
            <focusType>c:UserType</focusType>
            <script>
                <code>

import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
import ru.solarsecurity.inrights.prism.*;
import ru.solarsecurity.inrights.prism.delta.*;
import
ru.solarsecurity.prism.xml.ns._public.types_3.*;
import ru.solarsecurity.inrights.schema.result.*;
import ru.solarsecurity.inrights.model.api.*;
import ru.solarsecurity.inrights.repo.api.*;
import ru.solarsecurity.inrights.repo.sql.*;
import ru.solarsecurity.inrights.task.api.*;
import ru.solarsecurity.inrights.requests.api.*;
import ru.solarsecurity.inrights.model.impl.lens.*;
import ru.solarsecurity.inrights.schema.util.*;
import
ru.solarsecurity.inrights.model.impl.assignments.*;
import
ru.solarsecurity.inrights.model.api.context.*;
import ru.solarsecurity.inrights.prism.query.*;
import
ru.solarsecurity.inrights.model.api.context.ModelCo
ntext;
```

```
import
ru.solarsecurity.inrights.prism.PrismContext;
import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;
import ru.solarsecurity.inrights.util.exception.*;
import ru.solarsecurity.inrights.schema.*;
import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;
import static
ru.solarsecurity.inrights.common.SpringApplicationC
ontextHolder.*;
import javax.xml.namespace.QName;

def repositoryService =
getBean(RepositoryService.class);
def prismContext = getBean(PrismContext.class);

def assignmentsToRevoke = new ArrayList();

def findAssignment(def focus, def targetOid) {
    for(def assignment :
focus.asObjectable().getAssignment()) {
        if(assignment.getTargetRef() != null
& amp; & amp;
targetOid.equals(assignment.getTargetRef().getOid()
))
            return assignment;
    }
    return null;
}

def getOldAndNewPositionsOids(def focusContext) {
    def objectOld = focusContext.getObjectOld();
    def objectNew = focusContext.getObjectNew();
```

```

    def oldAssignments = objectOld != null ?
objectOld.asObjectable().getAssignment() : [];

    def newAssignments = objectNew != null ?
objectNew.asObjectable().getAssignment() : [];


    def oldPositionAssignments =
oldAssignments.findAll({
it.getTargetRef().getType().getLocalPart().contains
('PositionType') });

    def newPositionAssignments =
newAssignments.findAll({
it.getTargetRef().getType().getLocalPart().contains
('PositionType') });

    def oldPositionOids =
oldPositionAssignments.collect({
it.getTargetRef().getOid() });

    def newPositionOids =
newPositionAssignments.collect({
it.getTargetRef().getOid() });

    return [
        'old': oldPositionOids,
        'new': newPositionOids
    ]
}

def getForDeletingPositionsOids(def focusContext) {
    def oldAndNew =
getOldAndNewPositionsOids(focusContext);

    def result = oldAndNew['old'].findAll({
!oldAndNew['new'].contains(it) });

    return result;
}

```



```

}

def getForAddingPositionsOids(def focusContext) {
    def oldAndNew =
getOldAndNewPositionsOids(focusContext);
    def result = oldAndNew['new'].findAll({
!oldAndNew['old'].contains(it) });
    return result;
}

def getByOid(def oid, def clazz) {
    try {
        repositoryService =
getBean(RepositoryService.class);
        return repositoryService.getObject(clazz,
oid, null, new
OperationResult('1')).asObjectable();
    } catch(Exception ex) {
        println ex;
        return null;
    }
}

def calcRelatedObjectsOids(def positionsOids) {
    def result = [];
    for(oid in positionsOids) {
        def position = getByOid(oid,
ObjectType.class);
        if(position == null) continue;
        def contextAssignments =
position.getContextAssignments();
        if(contextAssignments != null)

result.addAll(contextAssignments.getAssignment().collect({ it.getTargetRef().getOid() }));
    }
}

```

```
        def parentOrgOids =
position.getParentOrgRef().collect({ it.getOid()
});

        result.addAll(parentOrgOids);

result.addAll(calcRelatedObjectsOids(parentOrgOids)
);

    }
    return result;
}

if(modelContext == null) return;
if(modelContext.getFocusContext() == null) return;
if(modelContext.getFocusContext().getDelta() ==
null) return;
if(modelContext.getFocusContext().getObjectAny() ==
null) return;

def newObject =
modelContext.getFocusContext().getObjectNew();
if(newObject == null) return;

def positionsOidsForDeleting =
getForDeletingPositionsOids(modelContext.getFocusCo
ntext());

def oidsToDelete =
calcRelatedObjectsOids(positionsOidsForDeleting);

def positionsOidsForAdding =
getForAddingPositionsOids(modelContext.getFocusCont
ext());

def oidsToAdd =
calcRelatedObjectsOids(positionsOidsForAdding);

def oidsToReallyDelete = oidsToDelete.findAll({
!oidsToAdd.contains(it) });
```

```

for(oid in oidsToReallyDelete) {
    def realAssignment = findAssignment(newObject,
oid);
    if(realAssignment != null)
        assignmentsToRevoke.add(realAssignment);
}

if (assignmentsToRevoke.size() > 0) {
    def focusClone = focus.clone();

    focusClone.assignment.removeAll(assignmentsToRevoke
);
    delta =
focus.asPrismObject().diff(focusClone.asPrismObject
());
    for(def itemDelta : delta.getModifications()) {
        def source = new
CustomItemDeltaSourceType();
        source.setDescription("Перевод по
должности");
        itemDelta.setSource(source);
    }

    modelContext.focusContext.addPrimaryDelta(delta);
    modelContext.setFresh(false);
}

</code>
</script>
</hook>
</change>
</modelHooks>

```

Пример скрипта, используемого для отзыва старых и назначения новых ролей при переводе в другое подразделение:

*//скрипт для отзыва старых и назначения новых ролей
при переводе в другое подразделение*

<modelHooks>

<change>

<hook>

<state>secondary</state>

<focusType>c:UserType</focusType>

<script>

<code>

import

*ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;*

import ru.solarsecurity.inrights.prism.;*

import ru.solarsecurity.inrights.prism.delta.;*

import

ru.solarsecurity.prism.xml.ns._public.types_3.;*

import ru.solarsecurity.inrights.schema.result.;*

import ru.solarsecurity.inrights.model.api.;*

import ru.solarsecurity.inrights.repo.api.;*

import ru.solarsecurity.inrights.repo.sql.;*

import ru.solarsecurity.inrights.task.api.;*

import ru.solarsecurity.inrights.requests.api.;*

import ru.solarsecurity.inrights.model.impl.lens.;*

import ru.solarsecurity.inrights.schema.util.;*

import

ru.solarsecurity.inrights.model.impl.assignments.;*

import

ru.solarsecurity.inrights.model.api.context.;*

import ru.solarsecurity.inrights.prism.query.;*

import

*ru.solarsecurity.inrights.model.api.context.ModelCo
ntext;*

import

ru.solarsecurity.inrights.prism.PrismContext;

```
import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;
import ru.solarsecurity.inrights.util.exception.*;
import ru.solarsecurity.inrights.schema.*;
import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;
import static
ru.solarsecurity.inrights.common.SpringApplicationC
ontextHolder.*;
import javax.xml.namespace.QName;

def repositoryService =
getBean(RepositoryService.class);
def prismContext = getBean(PrismContext.class);

def assignmentsToRevoke = new ArrayList();

def findAssignment(def focus, def targetOid) {
    for(def assignment :
focus.asObjectable().getAssignment()) {
        if(assignment.getTargetRef() != null
&amp;&amp;
targetOid.equals(assignment.getTargetRef().getOid()
))
            return assignment;
    }
    return null;
}

def getOldAndNewPositionsOids(def focusContext) {
    def objectOld = focusContext.getObjectOld();
    def objectNew = focusContext.getObjectNew();
```

```

    def oldAssignments = objectOld != null ?
objectOld.asObjectable().getAssignment() : [];

    def newAssignments = objectNew != null ?
objectNew.asObjectable().getAssignment() : [];


    def oldPositionAssignments =
oldAssignments.findAll({
it.getTargetRef().getType().getLocalPart().contains
('OrgType') });

    def newPositionAssignments =
newAssignments.findAll({
it.getTargetRef().getType().getLocalPart().contains
('OrgType') });


    def oldPositionOids =
oldPositionAssignments.collect({
it.getTargetRef().getOid() });

    def newPositionOids =
newPositionAssignments.collect({
it.getTargetRef().getOid() });


    return [
        'old': oldPositionOids,
        'new': newPositionOids
    ]
}

def getForDeletingPositionsOids(def focusContext) {
    def oldAndNew =
getOldAndNewPositionsOids(focusContext);

    def result = oldAndNew['old'].findAll({
!oldAndNew['new'].contains(it) });

    return result;
}

def getForAddingPositionsOids(def focusContext) {

```

```

    def oldAndNew =
getOldAndNewPositionsOids(focusContext);
    def result = oldAndNew['new'].findAll({
!oldAndNew['old'].contains(it) });
    return result;
}

def getByOid(def oid, def clazz) {
    try {
        repositoryService =
getBean(RepositoryService.class);
        return repositoryService.getObject(clazz,
oid, null, new
OperationResult('1')).asObjectable();
    } catch(Exception ex) {
        println ex;
        return null;
    }
}

def calcRelatedObjectsOids(def positionsOids) {
    def result = [];
    for(oid in positionsOids) {
        def position = getByOid(oid,
ObjectType.class);
        if(position == null) continue;
        def contextAssignments =
position.getContextAssignments();
        if(contextAssignments != null)

result.addAll(contextAssignments.getAssignment().co
llect({ it.getTargetRef().getOid() }));

        def parentOrgOids =
position.getParentOrgRef().collect({ it.getOid()
});
    }
}

```

```
        result.addAll (parentOrgOids) ;

result.addAll (calcRelatedObjectsOids (parentOrgOids)
) ;
    }
    return result;
}

if(modelContext == null) return;
if(modelContext.getFocusContext() == null) return;
if(modelContext.getFocusContext().getDelta() ==
null) return;
if(modelContext.getFocusContext().getObjectAny() ==
null) return;

def newObject =
modelContext.getFocusContext().getObjectNew();
if(newObject == null) return;

def positionsOidsForDeleting =
getForDeletingPositionsOids (modelContext.getFocusCo
ntext());
def oidsToDelete =
calcRelatedObjectsOids (positionsOidsForDeleting);

def positionsOidsForAdding =
getForAddingPositionsOids (modelContext.getFocusCont
ext());
def oidsToAdd =
calcRelatedObjectsOids (positionsOidsForAdding);

def oidsToReallyDelete = oidsToDelete.findAll({
!oidsToAdd.contains(it) });

for(oid in oidsToReallyDelete) {
```



```

    def realAssignment = findAssignment(newObject,
oid);
    if(realAssignment != null)
        assignmentsToRevoke.add(realAssignment);
}

if (assignmentsToRevoke.size() > 0) {
    def focusClone = focus.clone();

    focusClone.assignment.removeAll(assignmentsToRevoke
);

    delta =
focus.asPrismObject().diff(focusClone.asPrismObject
());
    for(def itemDelta : delta.getModifications()) {
        def source = new
CustomItemDeltaSourceType();
        source.setDescription("Перевод по
должности");
        itemDelta.setSource(source);
    }

    modelContext.focusContext.addPrimaryDelta(delta);
    modelContext.setFresh(false);
}

```

</code>

</script>

</hook>

</change>

</modelHooks><modelHooks>

<change>

<hook>

<state>secondary</state>

<focusType>c:UserType</focusType>

<script>

<code>

```
import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
import ru.solarsecurity.inrights.prism.*;
import ru.solarsecurity.inrights.prism.delta.*;
import
ru.solarsecurity.prism.xml.ns._public.types_3.*;
import ru.solarsecurity.inrights.schema.result.*;
import ru.solarsecurity.inrights.model.api.*;
import ru.solarsecurity.inrights.repo.api.*;
import ru.solarsecurity.inrights.repo.sql.*;
import ru.solarsecurity.inrights.task.api.*;
import ru.solarsecurity.inrights.requests.api.*;
import ru.solarsecurity.inrights.model.impl.lens.*;
import ru.solarsecurity.inrights.schema.util.*;
import
ru.solarsecurity.inrights.model.impl.assignments.*;
import
ru.solarsecurity.inrights.model.api.context.*;
import ru.solarsecurity.inrights.prism.query.*;
import
ru.solarsecurity.inrights.model.api.context.ModelCo
ntext;
import
ru.solarsecurity.inrights.prism.PrismContext;
import
ru.solarsecurity.inrights.schema.result.OperationRe
sult;
import ru.solarsecurity.inrights.util.exception.*;
import ru.solarsecurity.inrights.schema.*;
import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;
import static
ru.solarsecurity.inrights.common.SpringApplicationC
ontextHolder.*;
```

```

import javax.xml.namespace.QName;

def repositoryService =
  getBean(RepositoryService.class);
def prismContext = getBean(PrismContext.class);

def assignmentsToRevoke = new ArrayList();

def findAssignment(def focus, def targetOid) {
  for(def assignment :
    focus.asObjectable().getAssignment()) {
    if(assignment.getTargetRef() != null
    & &
    targetOid.equals(assignment.getTargetRef().getOid()
    ))
      return assignment;
  }
  return null;
}

def getOldAndNewPositionsOids(def focusContext) {
  def objectOld = focusContext.getObjectOld();
  def objectNew = focusContext.getObjectNew();

  def oldAssignments = objectOld != null ?
    objectOld.asObjectable().getAssignment() : [];
  def newAssignments = objectNew != null ?
    objectNew.asObjectable().getAssignment() : [];

  def oldPositionAssignments =
    oldAssignments.findAll({
    it.getTargetRef().getType().getLocalPart().contains
    ('OrgType') });

```

```

    def newPositionAssignments =
newAssignments.findAll({
it.getTargetRef().getType().getLocalPart().contains
('OrgType') });

    def oldPositionOids =
oldPositionAssignments.collect({
it.getTargetRef().getOid() });

    def newPositionOids =
newPositionAssignments.collect({
it.getTargetRef().getOid() });

    return [
        'old': oldPositionOids,
        'new': newPositionOids
    ]
}

def getForDeletingPositionsOids(def focusContext) {
    def oldAndNew =
getOldAndNewPositionsOids(focusContext);
    def result = oldAndNew['old'].findAll({
!oldAndNew['new'].contains(it) });
    return result;
}

def getForAddingPositionsOids(def focusContext) {
    def oldAndNew =
getOldAndNewPositionsOids(focusContext);
    def result = oldAndNew['new'].findAll({
!oldAndNew['old'].contains(it) });
    return result;
}

```

```
def getByOid(def oid, def clazz) {
    try {
        repositoryService =
getBean(RepositoryService.class);
        return repositoryService.getObject(clazz,
oid, null, new
OperationResult('1')).asObjectable();
    } catch(Exception ex) {
        println ex;
        return null;
    }
}

def calcRelatedObjectsOids(def positionsOids) {
    def result = [];
    for(oid in positionsOids) {
        def position = getByOid(oid,
ObjectType.class);
        if(position == null) continue;
        def contextAssignments =
position.getContextAssignments();
        if(contextAssignments != null)

result.addAll(contextAssignments.getAssignment().co
llect({ it.getTargetRef().getOid() }));

        def parentOrgOids =
position.getParentOrgRef().collect({ it.getOid()
});
        result.addAll(parentOrgOids);

result.addAll(calcRelatedObjectsOids(parentOrgOids)
);
    }
    return result;
}
```

```

}

if(modelContext == null) return;
if(modelContext.getFocusContext() == null) return;
if(modelContext.getFocusContext().getDelta() ==
null) return;
if(modelContext.getFocusContext().getObjectAny() ==
null) return;

def newObject =
modelContext.getFocusContext().getObjectNew();
if(newObject == null) return;

def positionsOidsForDeleting =
getForDeletingPositionsOids(modelContext.getFocusCo
ntext());
def oidsToDelete =
calcRelatedObjectsOids(positionsOidsForDeleting);

def positionsOidsForAdding =
getForAddingPositionsOids(modelContext.getFocusCont
ext());
def oidsToAdd =
calcRelatedObjectsOids(positionsOidsForAdding);

def oidsToReallyDelete = oidsToDelete.findAll({
!oidsToAdd.contains(it) });

for(oid in oidsToReallyDelete) {
    def realAssignment = findAssignment(newObject,
oid);
    if(realAssignment != null)
        assignmentsToRevoke.add(realAssignment);
}

if (assignmentsToRevoke.size() > 0) {

```

```

def focusClone = focus.clone();

focusClone.assignment.removeAll(assignmentsToRevoke
);

delta =
focus.asPrismObject().diff(focusClone.asPrismObject
());

for(def itemDelta : delta.getModifications()) {
    def source = new
CustomItemDeltaSourceType();
    source.setDescription("Перевод по
должности");
    itemDelta.setSource(source);
}

modelContext.focusContext.addPrimaryDelta(delta);
modelContext.setFresh(false);
}

</code>
</script>
</hook>
</change>
</modelHooks>

```

4.3.6 Настройка наследования ролей

Наследование ролей позволяет описать часто повторяющиеся свойства ролей в одной родительской роли. Чтобы применить эти свойства в конкретной роли, необходимо связать требуемую роль с родительской. Для этого следует отредактировать XML-код родительского объекта, где в тегах **assignment** задать ссылки на соответствующие роли.

Пример наследования ролей:

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

    oid="827c98f7-6e2b-4756-805c-06debd046e92"
    version="18">
    <name>Интернет</name>
    <displayName>Интернет</displayName>
    <assignment id="1">
        <targetRef oid="e1b56b20-9d09-472a-9ede-8fa19b0c112a" type="c:RoleType"><!-- Metarole for groups --></targetRef>
    </assignment>
    <assignment id="2">
        <targetRef oid="e1b56b20-9d09-472a-9ede-8fa19b0c112b"
                                type="c:RoleType"><!-- Metarole for account --></targetRef>
    </assignment>
    ...
</role>
```

4.3.7 Управление полномочиями в подключенных системах

Управление полномочиями в подключенных системах выполняется посредством назначения на роли специальных функциональных ролей – метаролей.

Метароль – это роль, назначение которой фокусному объекту приводит к созданию для него учетной записи или присвоению полномочий в целевой (подключенной) системе.

4.3.7.1 Создание метаролей

Для создания метароли следует ввести ее XML-описание в разделе **Import objects to** (Конфигурирование > Загрузить объект, см. раздел 3.8.1) или отредактировать необходимую роль в разделе **Object details** (подробнее см. раздел 3.8.2.4) таким образом, чтобы в XML-описании содержались строки (теги):

```
<inducement>
  <construction>
    ...
  </construction>
</inducement>
```

Загрузка объектов

Параметры

- ☒ Protected by encryption
- ☐ Fetch resource schema
- ☐ Keep OID
- ☐ Overwrite existing object
- ☐ Referential integrity
- ☒ Summarize errors
- ☒ Summarize successes
- ☐ Validate dynamic schema
- ☐ Validate static schema

Остановка после превышения количества ошибок

Количество

Загрузить объекты из

- ☐ File
- ☒ Embedded editor

```
3 * <simpleUserNotifier><!-- Генерирует оповеще-ние о создании УЗ -->
4   <operation>add</operation><!-- Операция добавления -->
5   <status>success</status><!-- Успешна -->
6 *   <recipientExpression><!-- Получатель (Ад-министратор) -->
7 *   <script>
8   <assignment>
9     <targetRef oid="идентификатор метароли" type="c:RoleType"></targetRef>
10 *   <activation>
11     ...
12   </activation>
13 </assignment>
14 <notificationConfiguration>
15 *   <handler>
16 *     <script><!-- Генерирует оповеще-ние о создании УЗ -->
```

Добавить объект

Рис. 80 – Создание метароли путем загрузки XML-описания

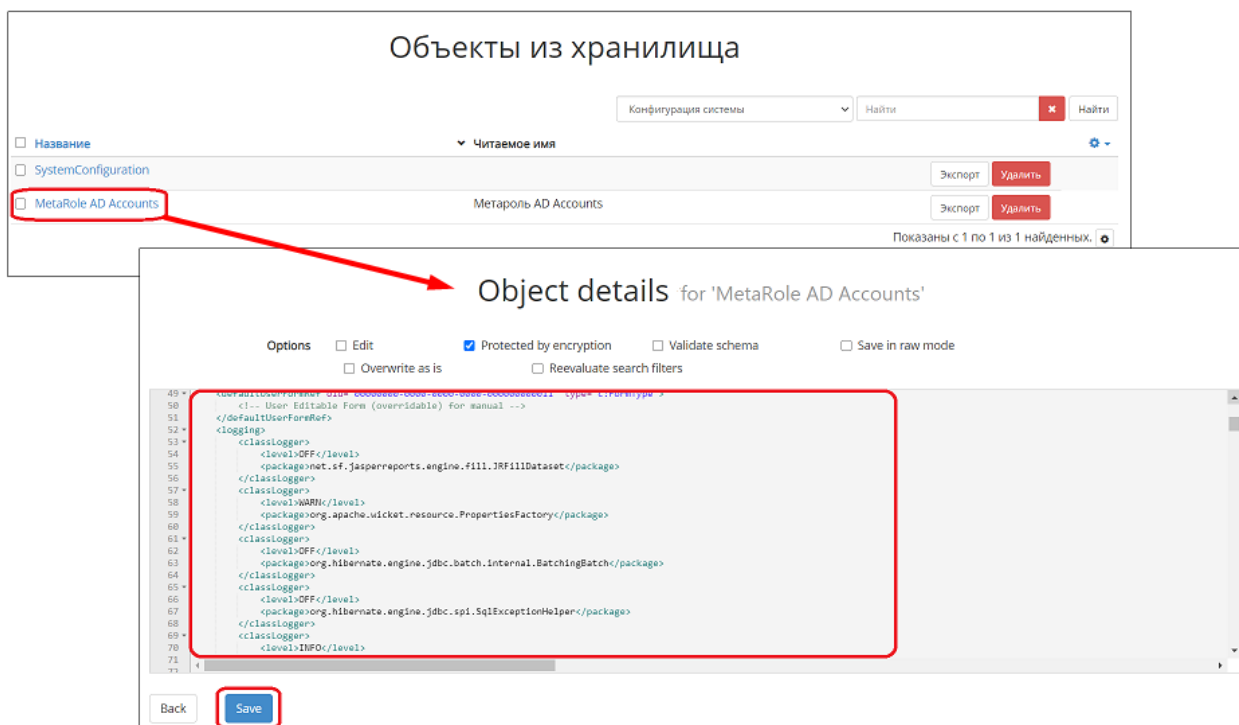


Рис. 81 – Создание метароли путем редактирования роли

В теге **construction** должны указываться идентификатор подключенной системы (ресурса), тип объекта и подкласс типа объекта. Например, для создания учетной записи необходимо прописать следующие строки:

```
<inducement>
  <construction>
    <resourceRef oid="..." type "c:ResourceType">
      <!-- ссылка на информационную систему -->
    </resourceRef>
    <kind>account</kind> <!-- тип объекта:
аккаунт/назначение/общий -->
    <intent>default</intent> <!-- подкласс типа
объекта -->
  </construction>
</inducement>
```

Примечание

Чтобы узнать идентификатор ИС (ресурса), необходимо просмотреть его XML-описание. Для этого в разделе **Информационные системы, подключенные к ПС (Информационные системы > Список**

информационных систем) следует в меню действий с конкретным ресурсом выбрать пункт Изменить XML (см. Рис. 82).

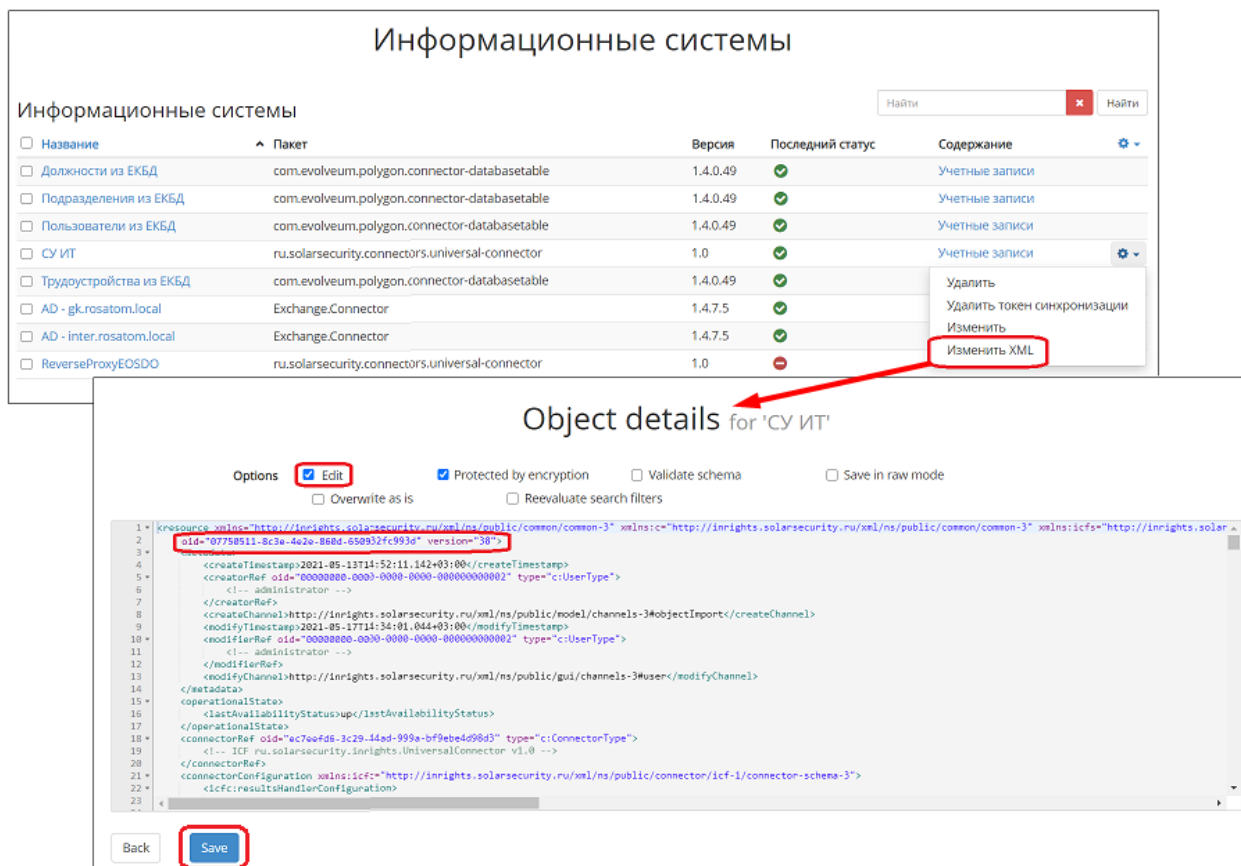


Рис. 82 – Просмотр XML-описания ресурса

Кроме того, в теге construction можно задать присвоение полномочий, например:

```
<inducement>
  <construction>
    <resourceRef oid="..." type="c:ResourceType">
      <!-- ссылка на информационную систему -->
    </resourceRef>
    <association> <!--тег ассоциации
с/присвоения полномочия -->
      <c:ref>group</c:ref> <!-- наименование
ассоциации в настройках информационной системы-->
    </association>
  </construction>
</inducement>
```

```
<kind>entitlement</kind> <!-- тип объекта:
назначение/общий -->

<intent>default</intent> <!-- подкласс типа объекта
-->

                                </projectionDiscriminator>
                                </associationFromLink>
                                <expression>
                                </outbound>
                                </association>
                                </construction>
</inducement>
```

4.3.7.2Примеры метаролей

В данном разделе приведены примеры метаролей, применяемых в Active Directory:

1. Метароль для аккаунта

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/publ
ic/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/publi
c/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/pu
blic/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/publi
c/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns
/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/p
ublic/resource/instance-3"

oid="e1b56b20-9d09-472a-9ede-8fa19b0c112b"
```

[illegible]

```
<intent>default</intent>
</projectionDiscriminator>
</associationFromLink>
</expression>
</outbound>
</association>
</construction>
<order>2</order>
</inducement>
<requestable>>false</requestable>
</role>
```

2. Метароль для группы

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

    oid="e1b56b20-9d09-472a-9ede-8fa19b0c112a"
    version="15">
    <name>Metarole for groups</name>
    <description>Системная роль</description>
    <metadata>
```

```
<createTimestamp>2016-06-
03T16:46:01.591+03:00</createTimestamp>

<creatorRef oid="00000000-0000-0000-0000-
000000000002" type="c:UserType"><!-- administrator
--></creatorRef>

<createChannel>http://inrights.solarsecurity.ru/xml
/ns/public/model/channels-
3#objectImport</createChannel>

</metadata>

<activation>

  <effectiveStatus>disabled</effectiveStatus>

  <disableTimestamp>2016-02-
02T13:40:41.375+03:00</disableTimestamp>

</activation>

<iteration>0</iteration>

<iterationToken/>

<inducement id="1">

  <construction>

    <resourceRef oid="8790e490-326a-46e9-ba35-
9e0c1dcbb41d" type="c:ResourceType"><!--Active
Directory&Exchange--></resourceRef>

    <kind>entitlement</kind>

    <intent>default</intent>

  </construction>

</inducement>

<requestable>false</requestable>
</role>
```

3. Метароль для подразделения

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/publi
c/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/publi
c/query-3"
```

```
xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"
```

```
xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"
```

```
xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"
```

```
xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"
```

```
oid="cd195cbf-e260-4809-9207-d6f2ddee04ab"  
version="2">
```

```
<name>Metarole for orgUnit</name>
```

```
<description>Системная роль</description>
```

```
<metadata>
```

```
<createTimestamp>2016-06-  
15T14:07:29.199+03:00</createTimestamp>
```

```
<creatorRef oid="00000000-0000-0000-0000-  
000000000002" type="c:UserType"><!-- administrator  
--></creatorRef>
```

```
<createChannel>http://inrights.solarsecurity.ru/xml/  
/ns/public/model/channels-  
3#objectImport</createChannel>
```

```
</metadata>
```

```
<activation>
```

```
<effectiveStatus>disabled</effectiveStatus>
```

```
<disableTimestamp>2016-02-  
02T13:40:41.375+03:00</disableTimestamp>
```

```
</activation>
```

```
<iteration>0</iteration>
```

```
<iterationToken/>
```

```
<inducement id="2">
```

```
<construction>
```



```

        <resourceRef oid="8790e490-326a-46e9-ba35-
9e0c1dcbb41d" type="c:ResourceType"><!-- Active
Directory & Exchange --></resourceRef>

        <kind>generic</kind>

        <intent>default</intent>

    </construction>

</inducement>

    <requestable>false</requestable>

</role>

```

4.3.7.3 Назначение метаролей

Для назначения роли метароли следует:

- 1) Перейти в раздел **Список ролей (Роли > Список ролей)**;
- 2) В списке ролей выбрать требуемую роль (см. Рисунок 88);
- 3) В разделе **Изменить роль <Наименование роли>** в меню действий **Назначения** выбрать пункт **Назначить роль** (см. Рис. 83);
- 4) В окне **Select object(s)** выбрать метароль(-и) и нажать кнопку **Assign**;
- 5) В полях, относящихся к метароли, указать необходимые значения свойств этой метароли (см. Рис. 84):

Таблица 15 – Перечень совместимых ОС

Параметр	Описание
Description	Подробное описание метароли
Tenant	Указывает, является ли метароль корневой (верхнего уровня)
Organization Unit/Project	Подразделение / Проект
Activation	
Административный статус	Административный статус метароли: • Разрешено – метароль активна (можно назначать); • Запрещено – метароль неактивна;

	• Архивировано – метароль неактивна, находится в архиве ролей
Начало действия	Действует с...
Окончание действия	Действует по...

6) Нажать кнопку **Сохранить** (см. Рис. 84).

Список ролей

Всё Найти

Наименование	Отображать как	Идентификатор	Описание
☐ Администратор ИБ ИТ-ресурса	Администратор ИБ ИТ-ресурса		
☐ Администратор ИС	Администратор ИС		Роль расширяет полномочия пользователя до администратора информационной системы
☐ Владелец ИР	Владелец ИР		Роль расширяет полномочия пользователя до владельца информационного ресурса
☐ Владелец ИС	Владелец ИС		Роль расширяет полномочия пользователя до владельца информационной системы
☐ Владелец роли			
☐ Доступ к пользовательским ролям (чтение)			
☐ Линейный руководитель			

Изменить роль 'Владелец ИС'

Основные

Название: Владелец ИС

Отображать как: Владелец ИС

Описание: Роль расширяет полномочия пользователя до владельца информационной системы

Тип: Тип

Состав роли

Назначения

Назначить учетную запись
Назначить роль
Назначить структурное подразделение
Удалить назначение

Select object(s)

Name	Description
☐ Администратор ИБ ИТ-ресурса	
☐ Администратор ИС	Роль расширяет полномочия пользователя до администратора информационной системы
☐ Владелец ИР	Роль расширяет полномочия пользователя до владельца информационного ресурса
☐ Владелец роли	Роль расширяет полномочия пользователя до владельца роли
☐ Доступ к пользовательским ролям (чтение)	
☐ Линейный руководитель	Роль расширяет полномочия пользователя до линейного руководителя
☐ Назначение делегатов для других пользователей	Техническая роль позволяет назначить делегата для любого пользователя.
☐ Пользователь	Роль позволяет пользователю выполнять стандартные операции в Системе (создание заявки, просмотр своей карточки пользователя, изменение пароля для своей УЗ).
☒ Metarole for accounts	Системная роль
☒ Metarole for groups	Системная роль

Показаны с 1 по 10 из 34 найденных.

Assign

Рис. 83 – Назначение метаролей: шаги 1-4

Изменить роль 'Владелец ИС'

Основные

Название: Владелец ИС

Отображать как: Владелец ИС

Описание: Роль расширяет полномочия пользователя до владельца информационной системы

Тип: Тип

Можно запрашивать: ☐

Идентификатор: Идентификатор

Уровень риска: Уровень риска

Владелец: Изменить Укажите владельца

Согласующий: Изменить Укажите согласующего

Минимально назначено: +

Максимально назначено: +

☐ Состав роли

☐ Назначения

Metarole for accounts

Description:

Tenant: None ✎ ✖

Organization Unit/Project: None ✎ ✖

Activation

Административный статус: Undefined

Начало действия: : :

Окончание действия: : :

Target

Metarole for accounts

Metarole for groups

Description:

vacEndDate: :

vacStartDate: :

vipUser: Не определено

workplaceNumber:

workplacePresence:

Варианты ☐ Принудительно ☐ Выполнить реконсильяцию ☐ Выполнить реконсильяцию затронутых объектов ☐ Запустить после всех согласований ☐ Показать результаты

Назад Сохранить

Рис. 84 – Назначение метаролей: шаги 5-6

Кроме того, назначить метароль определенной роли можно, отредактировав эту роль в разделе **Object details** (подробнее см. раздел 3.8.2.4) таким образом, чтобы в XML-описании содержались строки (теги):

```
<assignment>
  <targetRef oid="<идентификатор метароли>"
type="c:RoleType"></targetRef>
  <activation>
    ...
  </activation>
</assignment>
```

где **oid** - идентификатор назначаемой метароли.

Чтобы узнать идентификатор метароли, необходимо просмотреть ее XML-описание (см. раздел 3.8.2.4).

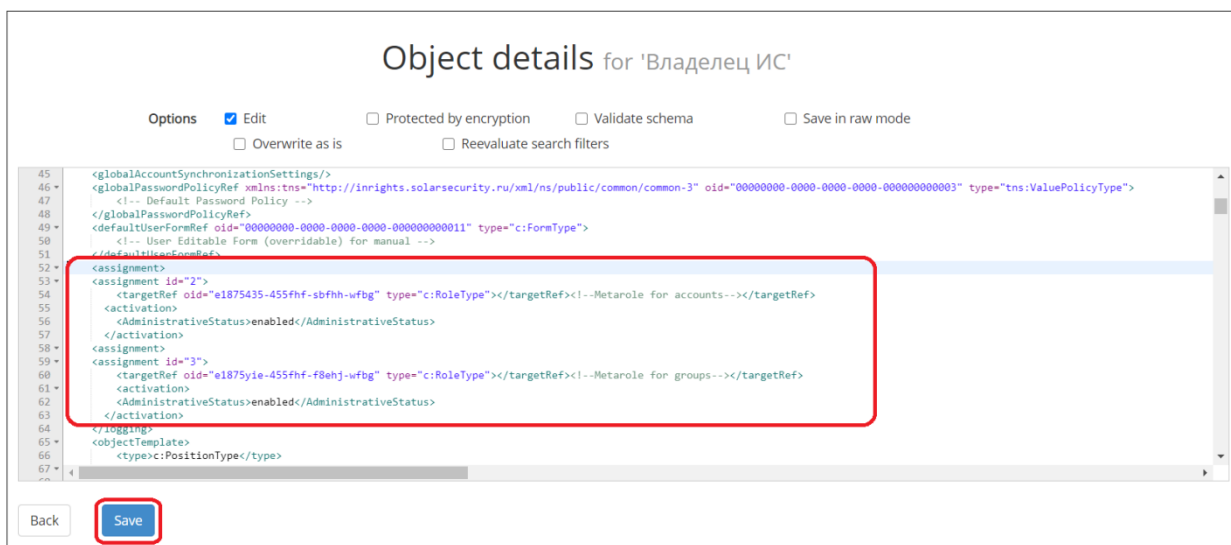


Рис. 85 – Назначение метаролей: редактирование XML-описания роли, которой необходимо назначить метароль

4.4 Настройка уведомлений

4.4.1 Общие сведения

Уведомления позволяют сообщать пользователям ПС о каких-либо системных событиях. Вид конкретного уведомления соответствует определенному шаблону. В ПС имеются следующие типы шаблонов уведомлений:

- 1) **User notifications** – уведомления о создании, изменении или удалении пользовательских данных.
- 2) **Resource object notifications** – уведомления о создании, изменении или удалении ресурсных объектов (аккаунтов, групп и т.п.).
- 3) **Workflow notifications** – уведомления о старте или завершении рабочих процессов.
- 4) **Account notifications** – уведомления о действиях.

Уведомления генерируются специальными функциями (уведомителями), использующими конкретный шаблон (см. Таблица 16 –).

Таблица 16 – Стратегии развертывания ПС

Наименование	Шаблон	Описание
simpleUserNotifier	User notification	Генерирует оповещение о действиях с пользовательскими данными
simpleResourceObjectNotifier	Resource notification object	Генерирует оповещение о действиях с объектами ресурса (аккаунты, группы)
userPasswordNotifier	User notification	Генерирует оповещение о действиях с паролем пользователя
accountPasswordNotifier	Account notification	Генерирует оповещение о действиях с паролем аккаунта
simpleWorkflowNotifier	Workflow notification	Генерирует оповещение о старте/завершении рабочих процессов (задач)
generalNotifier		Общий уведомитель, который запускается с помощью выражений, преобразующих событие в уведомлении

Настройка уведомлений заключается в задании параметров подключения к почтовому серверу (см. раздел 4.4.2), с помощью которого отправляются уведомления, и настройке уведомителей (см. раздел 4.4.3).

4.4.2 Настройка подключения к почтовому серверу

Для настройки подключения к почтовому серверу следует:

- 1) Перейти в раздел **Configuration for (Конфигурирование > Основные)**.
- 2) В списке **Mail Server** выбрать требуемый сервер или с помощью соответствующей кнопки добавить данные нового сервера.
- 3) В отобразившихся полях указать необходимые значения:
 - **Host** и **Port** – адрес и порт почтового сервера;
 - **Username** и **Password** – имя пользователя и пароль для подключения к почтовому серверу;
 - **Transport security** – протокол для обеспечения защиты данных (например, SSL).
- 4) Нажать кнопку **Save**.

The screenshot shows the 'Конфигурирование' (Configuration) window with two tabs: 'Основное' (Basic) and 'Протоколирование' (Logging). The 'Basic' tab is active. On the left, there are sections for 'Global password policy', 'Default user template', 'Object Policies', 'Global account synchronization settings', and 'Cleanup Policy'. On the right, there is a 'Notifications' section. A red box highlights the 'Mail Server' configuration area, which includes a dropdown menu for 'Mail Server' (set to 'mail.rosatom.ru'), input fields for 'Host' ('mail.rosatom.ru'), 'Port' ('25'), 'Username' ('mail@rosatom.ru'), and 'Password' ('Password is set'), and a dropdown for 'Transport security' ('StartTLS enabled').

Рис. 86 – Настройка подключения к почтовому серверу

Кроме того, можно отредактировать файл **systemConfiguration.xml**, задав требуемые значения вышеуказанных параметров:

```
<systemConfiguration
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"
```

```
xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"
```

```
xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"
```

```
xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"
```

```
xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"
```

```
xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"
```

```
oid="00000000-0000-0000-0000-000000000001"  
version="289">
```

```
...
```

```
<notificationConfiguration>
```

```
<handler>
```

```
...
```

```
</handler>
```

```
<mail>
```

```
<server>
```

```
<host>localhost</host>
```

```
<port>25</port>
```

```
<username/>
```

```
<password/>
```

```
<transportSecurity>none</transportSecurity>
```

```
</server>
```

```
<debug>>false</debug>
```

```
</mail>
```

```
</notificationConfiguration>
```

```
...
```

```
</systemConfiguration>
```

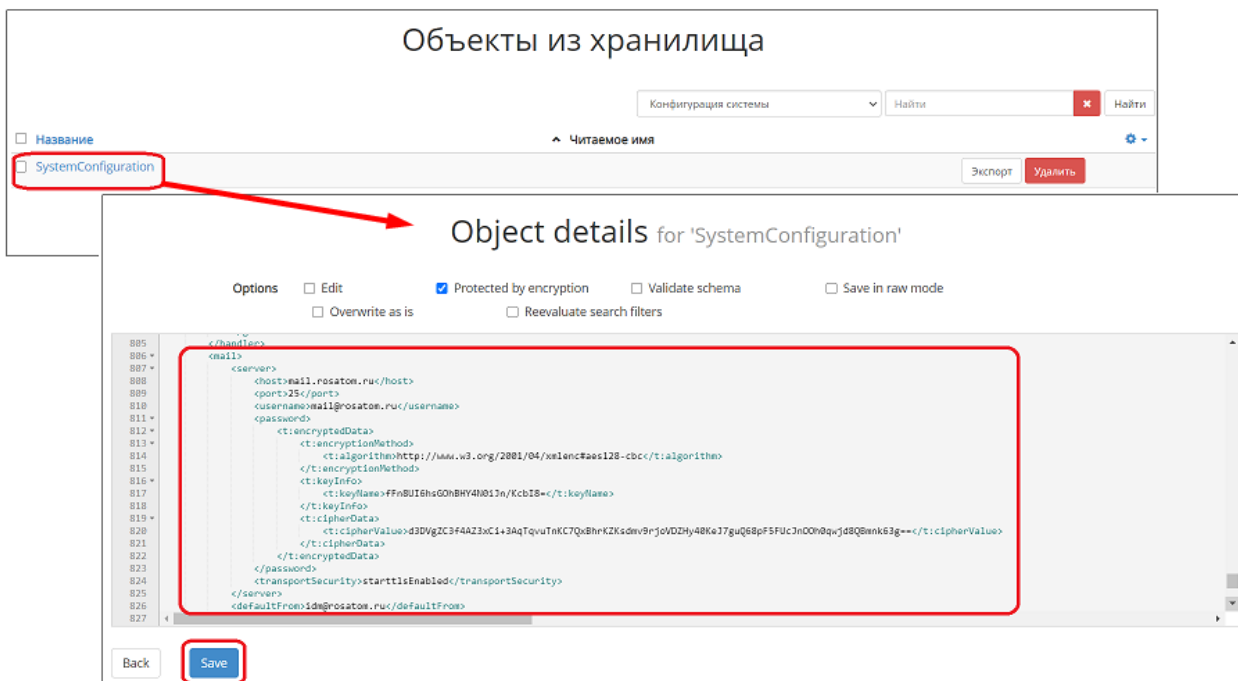


Рис. 87 – Настройка подключения к почтовому серверу:
редактирование файла systemConfiguration.xml

4.4.3 Настройка уведомителей

Настройка уведомителей выполняется путем задания соответствующих параметров объекта **SystemConfiguration** (Конфигурирование -> Объекты репозитория -> **SystemConfiguration**). То есть в XML-описание объекта **SystemConfiguration** необходимо добавить блок:

```
<notificationConfiguration>
  <handler> <!--обработчик оповещений-->
    <simpleUserNotifier>...</simpleUserNotifier>

    <simpleResourceObjectNotifier>...</simpleResourceObjectNotifier>

    <simpleRequestNotifier>...</simpleRequestNotifier>
  </handler>
  ...
</notificationConfiguration>
```


В этом блоке для каждого уведомителя (внутри тегов **<simpleUserNotifier>...</simpleUserNotifier>**, **<simpleResourceObjectNotifier>...</simpleResourceObjectNotifier>** и т.д.) следует задать:

- операцию (**<operation></operation>**),
- статус (**<status> </status>**),
- выражение для указания получателей уведомления (**<recipientExpression></recipientExpression>**),
- тему (**<subjectExpression></subjectExpression>**),
- тело (**<bodyExpression> </bodyExpression>**).

Далее приведен пример настройки уведомлений о создании учетной записи, об отзыве полномочий и о новой заявке:

```
<notificationConfiguration>
  <handler>
    <simpleUserNotifier><!-- Генерирует
оповещение о создании УЗ -->
      <operation>add</operation><!-- Операция
добавления -->
      <status>success</status><!-- Успешна -->
      <recipientExpression><!-- Получатель
(Администратор) -->
      <script>
        <code>
          import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;

          import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

          import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

          import
ru.solarsecurity.inrights.task.api.Task;
```

```
import
ru.solarsecurity.inrights.task.api.TaskManager;

TaskManager taskManager =
SpringApplicationContextHolder.getBean(TaskManager.
class);

Task task =
taskManager.createTaskInstance();

parentResult = task.getResult();

ModelController modelController =
SpringApplicationContextHolder.getBean(ModelControl
ler.class)

admin =
modelController.getObject(UserType.class,
"00000000-0000-0000-0000-000000000002",
null, task,
parentResult);

admin.asObjectable().getEmailAddress();

</code>
</script>
</recipientExpression>
<subjectExpression><!-- Тема письма -->
    <value>[СЦУДП] Уведомление о создании
учётной записи</value>
</subjectExpression>
<bodyExpression><!-- Тело письма -->
    <script>
        <code>

import
ru.solarsecurity.inrights.wf.impl.processes.common.
*;

import
ru.solarsecurity.inrights.prism.crypto.*;
```

```
import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;

import
ru.solarsecurity.inrights.task.api.Task;

import
ru.solarsecurity.inrights.task.api.TaskManager;

def protector =
SpringApplicationContextHolder.getBean(Protector.cl
ass);

body = "Уведомляем Вас о том,
что работнику:\n";

body += "ФИО: " +
requestee.fullName + "\n";

if (!(requestee.parentOrgRef
== null || requestee.parentOrgRef.isEmpty())) {

    TaskManager taskManager =
SpringApplicationContextHolder.getBean(TaskManager.
class);

    Task task =
taskManager.createTaskInstance();

    parentResult =
task.getResult();

    ModelController
modelController =

SpringApplicationContextHolder.getBean(ModelControl
ler.class)

    body += "Подразделение: ";
    for(def org :
requestee.parentOrgRef) {
```

```
body +=  
modelController.getObject  
        (OrgType.class, org.getOid(),  
null, task, parentResult).asObjectable().getName()  
+ ", ";  
        }  
        body = body.substring(0,  
body.length() - 2);  
        body += "\n";  
    }  
    if(requestee.title){  
        body += "Должность: " +  
requestee.title + "\n";  
    }  
  
        body += "\nБыла создана  
учётная запись:\n";  
        body += "Система: СЦУДП\n";  
        body += "Имя учётной записи: "  
+ requestee.name + "\n";  
        body += "Пароль: " +  
protector.decryptString(requestee.credentials.passw  
ord.value) + "\n";
```

```
</code>  
</script>  
</bodyExpression>  
<transport>mail</transport>  
</simpleUserNotifier>  
<simpleUserNotifier>  
    <operation>modify</operation><!--  
Изменение-->  
    <expressionFilter>  
        <script>  
            <code>
```

```
import
ru.solarsecurity.inrights.prism.path.ItemPath;

import
ru.solarsecurity.inrights.prism.delta.ContainerDelta;

import
ru.solarsecurity.inrights.prism.delta.ObjectDelta;

import
ru.solarsecurity.inrights.notifications.api.events.
ModelEvent;

import
ru.solarsecurity.inrights.prism.path.NameItemPathSegment;

import
ru.solarsecurity.inrights.schema.SchemaConstantsGenerated;

import
ru.solarsecurity.inrights.xml.ns._public.common.common_3.*;

boolean acceptable = false;

def rolesToDeleteCount = 0;
if (event instanceof ModelEvent) {
    ModelEvent modelEvent =
((ModelEvent) event);

    List segments = new
ArrayList();

    segments.add(new
NameItemPathSegment (SchemaConstantsGenerated.C_ASSIGNMENT));

    for (ObjectDelta objectDelta
: modelEvent.getFocusDeltas()) {
        ContainerDelta
containerDelta =
(ContainerDelta)
objectDelta.findItemDelta(new ItemPath(segments));
    }
}
```

```

                                if ( (containerDelta !=
null) &&&
(containerDelta.getValuesToDelete() != null) )
                                {
                                    for (Object reference
: containerDelta.getValuesToDelete()) {
                                        //проверяем, что в дельте есть хотя
бы одно удаление assignment с targetRef типа
RoleType,
                                        //только в таком случае нужно
посылать уведомление об отзыве ролей
                                        Class targetClass =
reference.value.targetRef.asReferenceValue().target
TypeCompileTimeClass;

                                        if(targetClass.isAssignableFrom(RoleType.class)) {

rolesToDeleteCount++;

                                        }

                                        }

                                        }

                                        acceptable = acceptable
|| rolesToDeleteCount > 0
                                        }

                                        }

                                acceptable
                                </code>
                                </script>
                                </expressionFilter>
                                <subjectExpression><!-- Тема письма -->
                                    <value>[СЦУДП] Уведомление об отзыве
ролей</value>
                                </subjectExpression>
                                <bodyExpression><!-- Тело письма -->
                                    <script>

```

```
<code>
import
ru.solarsecurity.inrights.prism.path.ItemPath;

import
ru.solarsecurity.inrights.prism.delta.ContainerDelta;

import
ru.solarsecurity.inrights.prism.delta.ObjectDelta;

import
ru.solarsecurity.inrights.notifications.api.events.
ModelEvent;

import
ru.solarsecurity.inrights.prism.path.NameItemPathSegment;

import
ru.solarsecurity.inrights.schema.SchemaConstantsGenerated;

import
ru.solarsecurity.inrights.schema.result.OperationResult;

import
ru.solarsecurity.inrights.model.impl.controller.ModelController;

import
ru.solarsecurity.inrights.task.api.Task;

import
ru.solarsecurity.inrights.task.api.TaskManager;

import
ru.solarsecurity.inrights.xml.ns._public.common.common_3.*;

import
ru.solarsecurity.inrights.prism.PrismObject;

import
ru.solarsecurity.inrights.wf.impl.processes.common.SpringApplicationContextHolder;

import
java.text.SimpleDateFormat;

import java.util.Calendar;
```

```
Calendar calendar =
Calendar.getInstance()

def dateFormat = new
SimpleDateFormat('dd MMMMM yyyy в HH:mm')
def now =
dateFormat.format(calendar.getTimeInMillis())

body = "Сообщаем, что в рамках
системы управления полномочиями у Вас " + now +
" были отозваны следующие
роли:";

ModelEvent modelEvent =
((ModelEvent)event);

List segments = new ArrayList();
segments.add(new
NameItemPathSegment(SchemaConstantsGenerated.C_ASSI
GNMENT));

TaskManager taskManager =
SpringApplicationContextHolder.getBean(TaskManager.
class);

ModelController modelController =
SpringApplicationContextHolder.getBean(ModelControl
ler.class)

for (ObjectDelta objectDelta :
modelEvent.getFocusDeltas()) {
    ContainerDelta containerDelta
=
(ContainerDelta)
objectDelta.findItemDelta(new ItemPath(segments));
    if ( (containerDelta != null)
&& (containerDelta.getValuesToDelete() !=
null) ) {
        for (Object reference :
containerDelta.getValuesToDelete()) {
            String oid =
reference.value.targetRef.oid;
```



```

Class targetClass =

reference.value.targetRef.asReferenceValue().target
TypeCompileTimeClass;

//В уведомлении
отображать только отзвы ролей (RoleType)

if(targetClass.isAssignableFrom(RoleType.class)) {
    Task task =
taskManager.createTaskInstance();
    OperationResult
parentResult = task.getResult();

    PrismObject obj =

modelController.getObject(targetClass, oid, null,
task, parentResult);

    body += "\n - " +
obj.asObjectable().getName();
}
}
}
}
body += "";
</code>
</script>
</bodyExpression>
<transport>mail</transport>
</simpleUserNotifier>
<simpleResourceObjectNotifier>
    <operation>add</operation><!-- Добавление
-->

    <status>success</status><!-- Успешно -->
    <objectKind>account</objectKind>
    <expressionFilter>

```

```
<script>
    <code>
        import
ru.solarsecurity.inrights.notifications.api.events.
ResourceObjectEvent;

        import
ru.solarsecurity.inrights.schema.constants.*;

        event instanceof ResourceObjectEvent &&&
event.getAccountOperationDescription().getSourceCha
nnel() !=

SchemaConstants.CHANGE_CHANNEL_LIVE_SYNC.toString()
    </code>
</script>
</expressionFilter>
<recipientExpression><!-- Получатель -->
    <script>
        <code>
            import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;

            import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

            import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

            import
ru.solarsecurity.inrights.task.api.Task;

            import
ru.solarsecurity.inrights.task.api.TaskManager;

            TaskManager taskManager =
SpringApplicationContextHolder.getBean(TaskManager.
class);
```

```
Task task =
taskManager.createTaskInstance();

parentResult = task.getResult();

ModelController modelController =
SpringApplicationContextHolder.getBean(ModelControl
ler.class)

admin = modelController.getObject
(UserType.class,
"000000000-0000-0000-0000-000000000002", null, task,
parentResult);

admin.asObjectable().getEmailAddress();
</code>
</script>
</recipientExpression>
<subjectExpression><!-- Тема письма -->
    <value>[СЦУДП] Уведомление о создании
учётной записи</value>
</subjectExpression>
<bodyExpression><!-- Тело письма -->
    <script>
        <code>
            import
ru.solarsecurity.inrights.prism.path.ItemPath;

            import
ru.solarsecurity.inrights.wf.impl.processes.common.
*;

            import
ru.solarsecurity.inrights.prism.crypto.*;

            import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.UserType;

            import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
```

```
import
ru.solarsecurity.inrights.model.impl.controller.Mod
elController;

import
ru.solarsecurity.inrights.task.api.Task;

import
ru.solarsecurity.inrights.task.api.TaskManager;

import
ru.solarsecurity.inrights.notifications.impl.format
ters.TextFormatter;


def protector =
SpringApplicationContextHolder.getBean(Protector.cl
ass);

resourceObjectEvent = event;
rod =
resourceObjectEvent.getAccountOperationDescription(
);

body = "Уведомляем Вас о том,
что работнику:\n";

body += "ФИО: " +
requestee.fullName + "\n";


TaskManager taskManager =
SpringApplicationContextHolder.getBean(TaskManager.
class);

Task task =
taskManager.createTaskInstance();

parentResult =
task.getResult();

ModelController
modelController =
SpringApplicationContextHolder.getBean(ModelControl
ler.class);

ShadowType shadow =
modelController.getObject(ShadowType.class,

rod.getObjectDelta().getObjectToAdd().asObjectable(
```

```

).getOid(), null, task,
parentResult).asObjectable();

        String login =
shadow.getName().toString();

        if (!(requestee.parentOrgRef
== null || requestee.parentOrgRef.isEmpty())) {

            body += "Подразделение:
";

            for(def org :
requestee.parentOrgRef) {

                body +=
modelController.getObject(
OrgType.class, org.getOid(),
null, task, parentResult).asObjectable().getName()
+ ", ";

            }

            body = body.substring(0,
body.length() - 2);

            body += "\n";

        }

        if(requestee.title){

            body += "Должность: " +
requestee.title + "\n";

        }

        body += "\nБыла создана
учётная запись:\n";

        body += "Система: " +
rod.getResource().asObjectable().getName() + "\n";

        body += "Имя учётной записи: "
+ login + "\n";

        if
(rod.getResource().asObjectable().getOid() ==
'8790e490-326a-46e9-ba35-9e0c1dcbb41d') {//
EXCHANGE

            def mailAttr =

```

```
shadow.getAttributes().asPrismContainerValue().find
(new ItemPath("PrimarySmtAddress"));

        body += "Адрес электронной
почты: " + ((mailAttr!=
null)?mailAttr.getValue().getValue():
        "в процессе создания") +
"\n";

    }

    body += "Пароль: " +
protector.decryptString(

rod.getObjectDelta().getObjectToAdd().asObjectable(
).getCredentials().getPassword().getValue()) +
"\n";

        </code>
        </script>
    </bodyExpression>
    <transport>mail</transport>

<needDecodePassword>false</needDecodePassword>
    </simpleResourceObjectNotifier>
    <simpleRequestNotifier><!-- Оповещение о
новой заявке на согласование -->
        <recipientExpression><!-- Получатель -->
            <script>
                <code>

                    import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

                    import
ru.solarsecurity.inrights.requests.impl.*;

                    import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
```

```

SpringApplicationContextHolder.getBean(AssignmentRe
questHelper.class).getByOid(
                                UserType.class,
event.approverOid).emailAddress;
                                </code>
                                </script>
                                </recipientExpression>
                                <subjectExpression><!-- Тема -->
                                <script>
                                <code>
                                body = '[inRights] Новая заявка
на предоставление ролей'
                                </code>
                                </script>
                                </subjectExpression>
                                <bodyExpression><!-- Тело письма -->
                                <script>
                                <code>

                                import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

                                import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

                                import
ru.solarsecurity.inrights.requests.impl.*;

                                import
java.text.SimpleDateFormat;

                                def format = new
SimpleDateFormat('dd MMM yyyy HH:mm')

                                def helper =
SpringApplicationContextHolder.getBean(AssignmentRe
questHelper.class);

```

```
body = "Сообщаем, что  
новая заявка на предоставление ролей ожидает Вашего  
согласования.\n\n"
```

```
body += "Номер заявки:  
${event.request.requestId}\n\n"
```

```
body += "Создана:  
${format.format(  
event.request.metadata.createTimestamp.toGregorianCalendar().time)}\n\n"
```

```
def requester =  
helper.getByOid(UserType.class,  
event.request.requesterRef.oid)
```

```
body += "Инициатор:  
${requester.fullName}\n\n"
```

```
def printInfo(helper,  
requestItemsIds) {  
    def body = "";  
    body += "Для перечисленных  
работников запрошены роли:\n"
```

```
    def main = [:]  
    def users = [:]  
    def roles = [:]  
    def orgs = [:]  
    def approvals =  
helper.getApprovals(event.request,  
requestItemsIds);  
    for (String userOid :  
approvals.keySet()) {  
        if(!users[userOid]){
```



```

                                users[userOid] =
helper.getByOid(UserType.class, userOid);
                                main[users[userOid]] = [];
                                }
                                for (String roleOid :
approvals.get(userOid)) {
                                    if(!roles[roleOid]){
                                        roles[roleOid] =
helper.getByOid(RoleType.class, roleOid);
                                    }

main[users[userOid]].add(roles[roleOid])
                                }
                                }
                                main =
main.sort{it.key.fullName.toString()}
                                def index = 1;
                                for(def requestee :
main.keySet()) {
                                    def groups =
main.get(requestee)

if(main.get(requestee).isEmpty()){
                                        continue
                                    }

                                    def name =
requestee.fullName.toString();
                                    body += "        ${index}.
${name}"

                                    for(def orgref :
requestee.getParentOrgRef()) {
                                        if(!orgs[orgref.oid]){
                                            orgs[orgref.oid] =
helper.getByOid(OrgType.class, orgref.oid);
                                        }

```

```
body += ",  
${orgs[orgref.oid].name}"  
}  
if(requestee.title){  
body += "  
${requestee.title}"  
}  
body += '\n';  
  
groups.sort{it.name.toString()}  
for (def role : groups) {  
body += "      -  
${role.name}\n"  
}  
index++;  
}  
return body;  
}  
body += printInfo(helper,  
event.requestItemIds)  
  
body += "Вы можете перейти  
к СОГЛАСОВАНИЮ ЗАЯВКИ ПО ССЫЛКЕ  
  
http://. /inrights/request/${event.  
request.requestId}"  
body=body.toString()  
</code>  
</script>  
</bodyExpression>  
<transport>mail</transport>  
  
<requestEventOperation>newApprovalAssignment</requestEventOperation>  
  
<requestType>addAssignments</requestType>  
</simpleRequestNotifier>
```

```
<simpleRequestNotifier><!-- Оповещение о
результате запроса полномочий -->
    <recipientExpression><!-- Получатель -->
        <script>
            <code>
                import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

                import
ru.solarsecurity.inrights.requests.impl.*;

                import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

SpringApplicationContextHolder.getBean(AssignmentRe
questHelper.class).getByOid(
                    UserType.class,
event.request.requesterRef.oid).emailAddress;
            </code>
        </script>
    </recipientExpression>
    <subjectExpression><!-- Тема письма -->
        <script>
            <code>
                import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.ApprovalStatusType;

                import
java.text.SimpleDateFormat;

                def format = new
SimpleDateFormat('dd MMM yyyy HH:mm')

                def status =
event.request.approvalStatus ==
ApprovalStatusType.REJECTED ? "отклонена" :
```

```
event.request.approvalStatus ==
ApprovalStatusType.PARTIALLY_APPROVED ? "частично
выполнена" :

                                "выполнена"

                                body="[inRights] Заявка №
${event.request.requestId} от ${format.format(

event.request.metadata.createTimestamp.toGregorianC
alendar().time)} ${status}"

                                body=body.toString()
                                </code>
                                </script>
                                </subjectExpression>
                                <bodyExpression><!-- Тело письма -->
                                <script>
                                <code>

                                import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;

                                import
ru.solarsecurity.inrights.wf.impl.processes.common.
SpringApplicationContextHolder;

                                import
ru.solarsecurity.inrights.requests.impl.*;

                                import
java.text.SimpleDateFormat;

                                def format = new
SimpleDateFormat('dd MMM yyyy HH:mm')

                                def helper =
SpringApplicationContextHolder.getBean(AssignmentRe
questHelper.class);

                                def status = " успешно
выполнена"

                                if(event.request.approvalStatus ==
ApprovalStatusType.REJECTED) {
```

```
        status = " была отклонена"
    }

    if(event.request.approvalStatus ==
ApprovalStatusType.PARTIALLY_APPROVED) {
        status = " была выполнена
только частично"
    }

    body = "Сообщаем, что Ваша
заявка на предоставление ролей ${status}.\n\n"

    body += "Номер заявки:
${event.request.requestId}\n\n"

    body += "Создана:
${format.format(

event.request.metadata.createTimestamp.toGregorianC
alendar().time)}\n\n"

    body += "Выполнена:
${format.format(new Date())}\n\n"

    def printUser(index,
requestee, orgs, helper){
        def body = ""
        body += "    ${index}.
${requestee.fullName}"
        for(def orgref :
requestee.getParentOrgRef()){
            if(!orgs[orgref.oid]){
                orgs[orgref.oid] =
helper.getByOid(OrgType.class, orgref.oid);
            }
            body += ",
${orgs[orgref.oid].name}"
        }
    }
}
```

```

    }
    if(requestee.title){
body += ",
${requestee.title}"
    }
    return body + '\n';
}

def printPartiallyInfo(orgs,
helper, map){

    def index = 1;
    def body = ""
    map =
map.sort{it.key.fullName.toString()}
    for(def requestee :
map.keySet()){

        if(map.get(requestee).isEmpty()){
            continue
        }
        body += printUser(index,
requestee, orgs, helper)

        map[requestee] =
map[requestee].sort{it.name.toString()}
        for (def role :
map.get(requestee)) {
            body += "          -
${role.name}\n"
        }
        index++;
    }
    return body
}

```

```

if(event.request.approvalStatus ==
ApprovalStatusType.PARTIALLY_APPROVED) {
    def approved = [:]
    def rejected = [:]
    def users = [:]
    def roles = [:]
    def orgs = [:]
    for(def requestItem :
event.request.requestItem) {
        def userOid =
helper.getUserOid(requestItem)
        if(!users[userOid]){
            users[userOid] =
helper.getByOid(UserType.class, userOid);
            approved[users[userOid]] =
[];
            rejected[users[userOid]] =
[];
        }
        def roleOid =
helper.getRoleOid(requestItem)
        if(!roles[roleOid]){
            roles[roleOid] =
helper.getByOid(RoleType.class, roleOid);
        }

        if(requestItem.approvalResult.result ==
ApprovalStatusType.APPROVED) {

approved[users[userOid]].add(roles[roleOid])
        }

        if(requestItem.approvalResult.result ==
ApprovalStatusType.REJECTED) {

rejected[users[userOid]].add(roles[roleOid])

```

```
    }  
    }  
  
    body += "Для перечисленных  
работников были предоставлены роли:\n"  
    body +=  
    printPartiallyInfo(orgs, helper, approved)  
  
    body += "Для перечисленных  
работников роли были отклонены:\n"  
    body +=  
    printPartiallyInfo(orgs, helper, rejected)  
    }else{  
        def users = [];  
        for (def ref :  
event.request.userRef) {  
  
users.add(helper.getByOid(UserType.class,  
ref.oid));  
  
        }  
        users =  
users.sort{it.fullName.toString()}  
        def roles = [];  
        for (def a :  
event.request.assignment) {  
  
roles.add(helper.getByOid(RoleType.class,  
a.targetRef.oid));  
  
        }  
        roles =  
roles.sort{it.name.toString()}  
        def orgs = [:];  
  
if(event.request.approvalStatus ==  
ApprovalStatusType.APPROVED) {  
  
        body += "По Вашей заявке для  
перечисленных работников:\n"
```



```
}

if(event.request.approvalStatus ==
ApprovalStatusType.REJECTED) {

    body += "Напоминаем, что в
Вашей заявке для работников:\n"

    }
    def index = 1;
    for(def requestee : users){
        body += printUser(index,
requestee, orgs, helper)
        index++;
    }

if(event.request.approvalStatus ==
ApprovalStatusType.APPROVED) {

    body += "были предоставлены
роли:\n"

    }

if(event.request.approvalStatus ==
ApprovalStatusType.REJECTED) {

    body += "были запрошены
роли:\n"

    }

    index = 1;
    for(def role : roles){
        body += "    ${index}.
${role.name}\n"

        index++;
    }
    }

    body += "Для просмотра
детальной информации по заявке Вы можете перейти
по ссылке
http://10.199.28.11:8081/inrights/request/${event.r
equest.requestId}"
```

```
        body=body.toString()
    </code>
</script>
</bodyExpression>
<transport>mail</transport>

<requestEventOperation>requestCompleted</requestEventOperation>
    <requestType>addAssignments</requestType>
</simpleRequestNotifier>
<accountPasswordNotifier>
    <transport>mail</transport>
</accountPasswordNotifier>
</handler>
<mail>
    <server>
        <host>localhost</host>
        <port>25</port>
        <username/>
        <password/>

<transportSecurity>none</transportSecurity>
    </server>
    <debug>false</debug>
</mail>
</notificationConfiguration>
```

4.5 Настройка взаимодействия ПС и Active Directory

4.5.1 Минимальные полномочия УЗ для настройки коннектора AD

Для настройки коннектора AD рекомендуется в AD создать учетную запись пользователя с минимальными административными правами. Для этого следует:

- 1) Запустить **Active Directory** - пользователи и компьютеры;
- 2) Развернуть требуемый домен (например, domain.example.com) и нажать правой кнопкой мыши на контейнере **Users**;
- 3) В отобразившемся контекстном меню выбрать пункт **Создать > Пользователь**;
- 4) Ввести всю необходимую информацию о пользователе, в том числе указать **Имя пользователя** и нажать кнопку **Далее**;
- 5) Указать пароль пользователя и нажать ОК.

Примечание

Рекомендуется снять флажок **Требовать смены пароля** при следующем входе в систему и установить флажок Срок действия пароля не ограничен.

- 6) Нажать правой кнопкой мыши на контейнере **Users**;
- 7) В отобразившемся контекстном меню выбрать пункт **Делегирование управления** и нажать кнопку **Далее**;
- 8) Нажать кнопку **Добавить**, ввести имя нужного пользователя и последовательно нажать кнопки **ОК** и **Далее**;
- 9) В окне **Мастер делегирования управления** в блоке Делегировать следующие обычные задачи установить флажки (см. Рис. 88):
 - Создание, удаление и управление учетными записями пользователей (Create, delete, and manage user accounts);
 - Переустановить пароли пользователей и установить изменение паролей при следующей перезагрузке (Reset user passwords and force password change at next logon);
 - Чтение информации обо всех пользователях (Read all user information);
 - Создание, удаление и управление группами (Create, delete and manage groups);

- Изменение членства в группе (Modify the membership of a group).

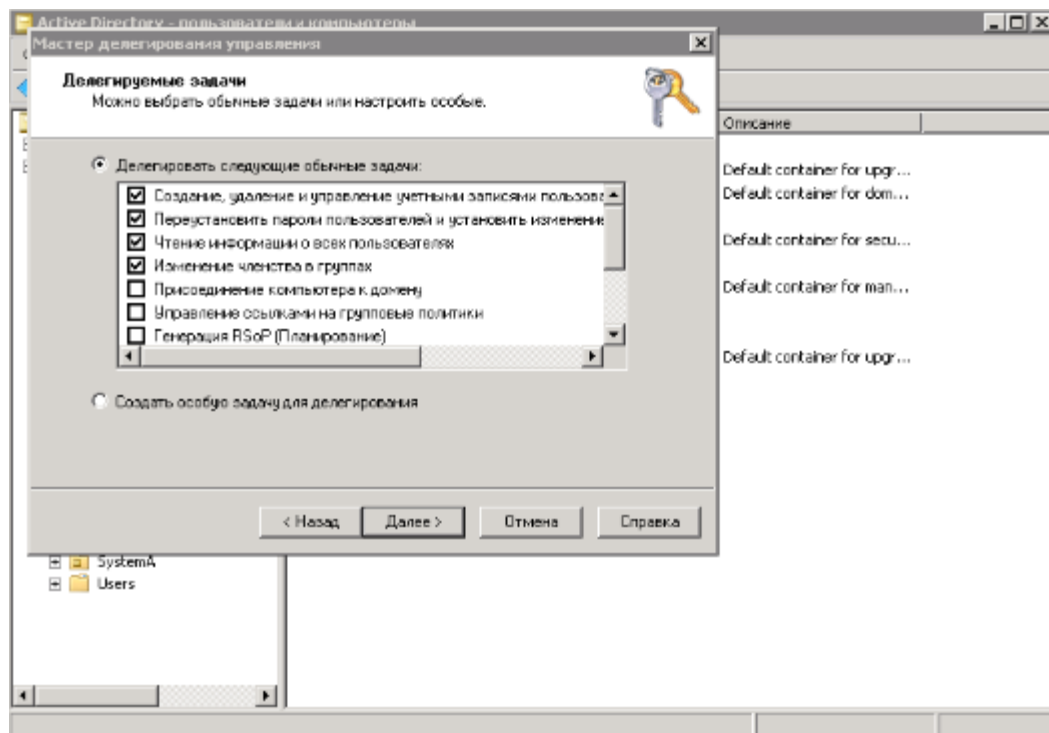


Рис. 88 – Настройка полномочий

10) Последовательно нажать кнопки **Далее** и **Завершить**.

4.5.2 Настройка мультидоменной структуры (кросс-ресурсных ассоциаций)

Иногда требуется включить пользователей одного домена (AD 1) в группы, находящиеся в другом домене (AD 2). При этом домены должны быть доверенными друг для друга. Следовательно, в ПС необходимо настроить кросс-ресурсные ассоциации. Для этого следует:

1) Подключить ресурсы AD1 и AD2, то есть создать и настроить ресурсы в ПС, связать с каждым роли и пользователей. Проверить, что пользователи, созданные в ПС, появляются в AD, а соответствующие атрибуты аккаунтов в AD изменяются при изменении данных пользователей ПС (например, при переименовании).

2) В каталог **idm_HOME/schema** добавить файл **Extension.xsd** со следующим содержимым:

```
<xsd:schema elementFormDefault="qualified"
targetNamespace="http://solarsecurity.ru/inview/idm
/org"
xmlns:tns="http://solarsecurity.ru/inview/idm/org"
xmlns:a="http://prism.solarsecurity.ru/xml/ns/publi
c/annotation-3"
xmlns:c="http://inrights.solarsecurity.ru/xml/ns/pu
blic/common/common-3"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="resourceRef"
type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
      <xsd:appinfo>
        <a:indexed>false</a:indexed>
        <a:displayName>Resource
ref</a:displayName>
<a:displayOrder>120</a:displayOrder>
      </xsd:appinfo>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

3) Перезапустить приложения **admin-gui** и **idm**.

4) В схему обоих ресурсов в разделе **Schema** добавить определение атрибута **objectSid** для типа **AccountObjectClass**:

```
<xsd:element minOccurs="0" name="objectSid"
type="xsd:base64Binary">
  <xsd:annotation>
```

```
<xsd:appinfo>
  <a:displayOrder>110</a:displayOrder>

<ra:frameworkAttributeName>objectSid</ra:frameworkA
ttributeName>
  </xsd:appinfo>
</xsd:annotation>
</xsd:element>
```

5) В разделе **schemaHandling** задать (для обоих ресурсов):

- маппинг атрибута **objectSid**:

```
<attribute>
  <c:ref>ri:objectSid</c:ref>
  <displayName>Object SID</displayName>
  <fetchStrategy>explicit</fetchStrategy>
</attribute>
```

- ассоциации для типа **AccountObjectClass**:

```
<association>
  <c:ref>ad2_group</c:ref>
  <exclusiveStrong>false</exclusiveStrong>
  <kind>entitlement</kind>
  <intent>default</intent>
  <associationProcessor>

<name>ADCrossForestAssociationProcessor</name>
  <parameter>
    <name>resourceOid</name>
    <value>63d03503-367d-4668-885c-
8eff890ae67e</value>
  </parameter>
</associationProcessor>
</association>
```

где **63d03503-367d-4668-885c-8eff890ae67e** — это идентификатор ресурса на домене AD 2, в котором хранятся группы.

6) Добавить метароль (подробнее о создании метароли см. в разделе 4.3.7.1):

```
<role
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

    oid="8ff3a8b6-c178-4083-add1-4682a123d6d6"
<!--идентификатор ресурса на домене, в котором
хранятся группы-->
    version="1">
    <name>Метароль для foreign групп AD 2</name>
    <displayName>Metarole for foreign
groups</displayName>
    <inducement id="1">
        <constructionExpression>
            <script>
                <code>
                    import
ru.solarsecurity.inrights.prism.*;
                    import
ru.solarsecurity.inrights.prism.path.ItemPath;
                    import
ru.solarsecurity.inrights.xml.ns._public.common.com
mon_3.*;
```

```
import
ru.solarsecurity.inrights.common.SpringApplicationC
ontextHolder;

import
ru.solarsecurity.inrights.model.impl.lens.Construct
ionTypeFactory;

//получить oid ресурса, с которым
связана назначаемая роль
def getResourceRef(evalAssignment)
{
    String resourceOid = null;
    for(def role :
evalAssignment.getRoles().getZeroSet()) {
        def assignmentType =
role.getAssignment();

if(assignmentType.getExtension() == null) continue;
        def resourceRef =

assignmentType.getExtension().asPrismContainerValue
().find(new ItemPath("resourceRef"));
        if(resourceRef == null)
continue;

        resourceOid = (String)
resourceRef.getRealValues().stream().findAny().orEl
se(null);
    }
    return resourceOid;
}

//получить oid shadow-объекта, с
которым связана назначаемая роль
def getShadowRef(evalAssignment) {
    for(def evalRole :
evalAssignment.getRoles().getZeroSet()) {
        def role =
evalRole.getRole().asObjectable();

if(role.getLinkRef().isEmpty()) return null;
```



```

        return
role.getLinkRef().get(0).getOid();
    }
    return null;
}
    def resourceRef =
getResourceRef(evalAssignment);
    if(resourceRef == null) return
null;

    def shadowRef =
getShadowRef(evalAssignment);
    if(shadowRef == null) return null;
    def constructionTypeFactory =
SpringApplicationContextHolder.getBean(Construction
TypeFactory.class);

    //создать construction, создающий
аккаунт на ресурсе resourceRef
    // с ассоциацией "ad2_group",
предоставляющей аккаунту членство в группе.
    //"ad2_group" - это название
ассоциации на ресурсе AD 1 или AD 2
    //"account" - это тип создаваемого
объекта на ресурсе. Нам нужно создать аккаунт
    //"default" - это intent
создаваемого объекта на ресурсе
    //resourceRef - это oid ресурса, на
котором нужно создать аккаунт
    //shadowRef - это oid shadow-
объекта, который связан с группой на стороне AD 2
    return
constructionTypeFactory.createConstructionType
("ad2_group", "account", "default",
resourceRef, shadowRef);
</code>
</script>
</constructionExpression>
<order>2</order>

```

```
</inducement>
</role>
```

7) Назначить добавленную в п.6 метароль всем ролям, которые связаны с группами на AD 2 (назначить метароль можно, например, с помощью шаблона объектов).

8) В **SystemConfiguration** добавить следующий скриптовый обработчик, предварительно исправив константы идентификаторов (помечены комментариями):

```
<modelHooks>
  <change>
    <hook>
      <name>Foreign resources hook</name>
      <state>secondary</state>
      <focusType>c:UserType</focusType>
      <script>
        <code>
          import
org.springframework.beans.factory.annotation.Autowired;

          import
ru.solarsecurity.inrights.model.api.context.ModelContext;

          import
ru.solarsecurity.inrights.model.api.context.ModelState;

          import
ru.solarsecurity.inrights.model.api.hooks.ChangeHook;

          import
ru.solarsecurity.inrights.model.api.hooks.HookOperationMode;

          import
ru.solarsecurity.inrights.model.impl.lens.LensFocusContext;
```

```
import
ru.solarsecurity.inrights.prism.*;

import
ru.solarsecurity.inrights.prism.delta.ItemDelta;

import
ru.solarsecurity.inrights.repo.api.RepositoryService;

import
ru.solarsecurity.inrights.schema.result.OperationResult;

import
ru.solarsecurity.inrights.task.api.Task;

import
ru.solarsecurity.inrights.util.exception.ObjectNotFoundException;

import
ru.solarsecurity.inrights.util.exception.SchemaException;

import
ru.solarsecurity.inrights.util.exception.SystemException;

import
ru.solarsecurity.inrights.xml.ns._public.common.common_3.*;

import javax.xml.namespace.QName;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.List;
import java.util.stream.Collectors;
import static
ru.solarsecurity.inrights.common.SpringApplicationContextHolder.*;

import static
ru.solarsecurity.inrights.model.api.AssignmentUtil.getAssignmentTypesToAdd;

import static
ru.solarsecurity.inrights.model.impl.lens.LensUtil.getAllModifications;
```

```

        def OID_METAROLE_FOREIGN =
"8ff3a8b6-c178-4083-add1-4682a123d6d6"; //oid мета-
роли для foreign групп

        def trustedResources =
Arrays.asList(
            "b4683219-1275-42f1-a95c-
af7beffd3ba9" /* oid AD 1 */,
            "63d03503-367d-4668-885c-
8eff890ae67e" /* oid AD 2 */
        );

        def repositoryService =
getBean(RepositoryService.class);

        //получить список oid'ов ресурсов,
на которых у пользователя есть учетные записи

        def getUserResourceOids(objectNew,
repositoryService) {
            def result = new ArrayList();
            if(objectNew == null) return
result;

            for(ObjectReferenceType ref :
objectNew.asObjectable().getLinkRef()) {

if(!ref.getType().getLocalPart().equals(ShadowType.
class.getSimpleName())) continue;

                try {
                    ShadowType shadowType =
repositoryService.getObject(ShadowType.class,
                        ref.getOid(), null, new
OperationResult("getUserResourceOids")).asObjectabl
e();

result.add(shadowType.getResourceRef().getOid());

                } catch
(ObjectNotFoundException e) {
                    continue;
                } catch (SchemaException e)
{

```

```

throw new
SystemException(e);
    }
}
return result;
}
//установить extension/resourceRef
для назначения
def setResourceRef(AssignmentType
assignmentType, String oid) {
    PrismPropertyDefinition
propertyDef =
getPrismContext().getSchemaRegistry().

findPropertyDefinitionByElementName(new
QName("resourceRef"));

    PrismProperty resourceRef =
propertyDef.instantiate();

    resourceRef.setRealValue(oid);

    ExtensionType extensionType =
new ExtensionType();
    try {

extensionType.asPrismContainerValue().add(resourceR
ef);

assignmentType.setExtension(extensionType);

getPrismContext().adopt(assignmentType);
    } catch (SchemaException e) {
        throw new
SystemException(e);
    }
}
//получить список добавляемых
пользователю назначений, которые связаны с foreign
группами

```

```

        //назначение считается связанным с
foreign группами, если оно назначает роль,
        //у которой есть назначение на
метароль для foreign групп
        def
getForeignAssignmentsToAdd(allModifications,
repositoryService, OID_METAROLE_FOREIGN) {
            def assignmentsToAdd =
getAssignmentTypesToAdd(allModifications);
            return assignmentsToAdd.stream()
                .filter({ a ->
a.getTargetRef() != null &&&
a.getTargetRef().getType().
getLocalPart().equals(RoleType.class.getSimpleName(
)) })
                .filter({ a -> L:{
                    try {
                        RoleType targetType
= repositoryService.getObject
                        (RoleType.class,
a.getTargetRef().getOid(), null, new
OperationResult("script hook")).asObjectable();
                        for(AssignmentType
assignmentType : targetType.getAssignment()) {
if(assignmentType.getTargetRef() != null &&&
assignmentType.getTargetRef().getOid().equals(OID_M
ETAROLE_FOREIGN))
                            return true;
                        }
                        return false;
                    } catch
(ObjectNotFoundException | SchemaException e) {
                        return false;
                    }
                }

```

```
        } })
        .collect(Collectors.toList());
    }
    def focusContext =
modelContext.getFocusContext();
    def foreignAssignmentsToAdd =

getForeignAssignmentsToAdd(getAllModifications(focu
sContext), repositoryService,
OID_METAROLE_FOREIGN);

if(foreignAssignmentsToAdd.isEmpty()) return;
    def currentUserResourceOids =
        getUserResourceOids(focusContext.getObjectNew(),
repositoryService).stream()
            .filter({ r ->
trustedResources.contains(r) })
            .collect(Collectors.toList());

        for(AssignmentType foreignAssignment
: foreignAssignmentsToAdd) {

if(currentUserResourceOids.isEmpty()) {
            //у пользователя нет учеток
ни в одной из AD
            //создаем учетку в той AD, в
которой расположена назначаемся группа
            def roleOid =
foreignAssignment.getTargetRef().getOid();
            def targetRole =
repositoryService.getObject
                (RoleType.class, roleOid, null, new
OperationResult("1")).asObjectable();
            def shadowOid =
targetRole.getLinkRef().get(0).getOid();
            def groupShadow =
```

```

        repositoryService.getObject(ShadowType.class,
shadowOid, null, new
OperationResult("1")).asObjectable();

                def resourceOid =
groupShadow.getResourceRef().getOid();

setResourceRef(foreignAssignment, resourceOid);

                } else {

                        //У пользователя есть
учетка в AD 1 или AD 2

                        //используем существующую
учетку

setResourceRef(foreignAssignment,
currentUserResourceOids.get(0));

                }

        }

        modelContext.setFresh(false);
</code>
</script>
</hook>
</change>
</modelHooks>

```

Примечание

В случае с 3-мя доменами/лесами AD необходимо создать еще один ресурс, проверить его работоспособность и настроить аналогично остальным двум ресурсам. При этом ассоциация **ad2_group** во всех ресурсах AD будет выглядеть следующим образом:

```

<association>
    <c:ref>ad2_group</c:ref>
    <exclusiveStrong>false</exclusiveStrong>
    <kind>entitlement</kind>
    <intent>default</intent>
    <associationProcessor>

<name>ADCrossForestAssociationProcessor</name>

```



```
<parameter>
  <name>resourceOid</name>
  <value>oid AD 1</value>
</parameter>
<parameter>
  <name>resourceOid</name>
  <value>oid AD 2</value>
</parameter>
<parameter>
  <name>resourceOid</name>
  <value>oid AD 3</value>
</parameter>
</associationProcessor>
</association>
а константа в скриптовом обработчике в
SystemConfiguration будет выглядеть так:
def trustedResources = Arrays.asList(
  "oid AD 1",
  "oid AD 2",
  "oid AD 3"
);
```

4.6 Настройка взаимодействия ПС и Microsoft Exchange Server

4.6.1 Минимальные полномочия УЗ для настройки коннектора Exchange Server

Для управления Exchange рекомендуется создать учетную запись пользователя с административными правами на выполнение следующих команд **Exchange PowerShell** (командлет):

- Enable-Mailbox;
- Disable-Mailbox;
- Set-Mailbox;
- Enable-MailUser;

- Disable-MailUser;
- Set-MailUser;
- Get-MailUser;
- Get-User;
- Get-Mailbox;
- Set-User.

Настройка прав выполняется в соответствии с документацией на установленную версию **Exchange Server**.

4.6.2 Подключение к нескольким Exchange-серверам

Начиная с версии 1.4.2.778 коннектора Exchange, поддерживается подключение к нескольким Exchange-серверам. Для настройки подключения к нескольким серверам следует при создании / редактировании ресурса (см. раздел 3.6.1, 3.6.2.2.2):

1) В разделе **Configuration** задать значения параметров:

- **Version of Exchange** – версия Exchange;
- **URI of RemotePowerShell** – доступ к Exchange-серверам.

Формат: POWER-SHELL_URI[;EXCHANGE_VERSION]. Если EXCHANGE_VERSION не указана, то используется версия по умолчанию.

Примечание

URI является ключом, который надо использовать в outbound-mappings в разделе **Schema Handling**.

Конфигурация

Подразделения из ЕКБД ⚙️

configurationProperties

Name Quoting ⓘ	
Host ⓘ	db
TCP Port ⓘ	5432
User ⓘ	postgres
User Password ⓘ	
Database ⓘ	HR
Table ⓘ	public.departments
Key Column ⓘ	obj_id
Password Column ⓘ	
JDBC Driver ⓘ	org.postgresql.Driver
JDBC Connection URL ⓘ	jdbc:postgresql://db:5432/HR
Enable writing empty string ⓘ	Не определено
Rethrow all SQLExceptions ⓘ	Не определено

Рис. 89 – Конфигурирование Exchange-коннектора

2) Изменить схему, добавив новый атрибут (чтобы можно было передать параметр, по которому коннектор определит, на какой именно сервер следует отправлять запрос):

```
<xsd:complexType name="AccountObjectClass">
  <xsd:sequence>
    . . . .
    <xsd:element minOccurs="0"
name="advancedOptions" type="xsd:string">
      <xsd:annotation>
        <xsd:appinfo>
          <a:displayOrder>1210</a:displayOrder>
        </xsd:appinfo>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

3) Настроить Schema Handling:

```
<attribute>
  <c:ref>ri:advancedOptions</c:ref>
  <displayName>Дополнительные
параметры</displayName>
  <exclusiveStrong>false</exclusiveStrong>
  <outbound>
    <expression>
      <script>
        <code>'timestamp=' +
System.currentTimeMillis() +
';uri=http://dcmain.inrights.ru/PowerShell'</code>
      </script>
    </expression>
  </outbound>
</attribute>
```

В OutboundMapping передаётся список параметров в виде PAR1_NAME=PAR1_VAL;PAR2_NAME=PAR2_VAL. На стороне коннектора значение атрибута **advancedOptions** разбирается на список параметров. При этом параметр uri является ключом и его значение должно совпадать с одним из значений, которое было указано в поле **URI of RemotePowerShell** (см. п.1).

ВАЖНО! В данном случае используется только URI (без версии).

Атрибуты передаются в коннектор, если они были изменены. Поэтому следует указать **timestamp** или другой счётчик.

4.7 Настройка серверных задач

4.7.1 Общие сведения

Серверные задачи выполняются на сервере ПС в фоновом режиме. В зависимости от назначения, серверные задачи подразделяются на следующие:

- **Import** – задача для первоначального импорта фокусных объектов (аккаунтов/подразделений/полномочий/должностей) из кадрового источника или целевой системы (с созданием shadow (теневых) объектов в ПС);
- **Reconciliation** – задача, аналогичная по алгоритму работы задаче импорта, используется для повторной загрузки фокусных объектов или обновления значений их атрибутов (и атрибутов shadow (теневых) объектов);
- **Recomputation** – задача для «пересчета» фокусных объектов, их назначений, назначенных ролей, учетных записей и полномочий;
- **Live synchronization** – задача для синхронизации фокусных и ресурсных объектов с кадровым источником или целевой системой. Задача отслеживает изменения в кадровом источнике или целевой системе и применяет их к фокусному объекту (и shadow объекту) в соответствии с настроенными правилами;
- **Custom** – задача для запуска прикладного кода по расписанию;
- **Audit** – служебная задача, которая выполняет индексацию событий аудита для вывода данных в отчетах.

Для создания серверных задач и управления ими в AGUI имеется специальный раздел (см. раздел 3.7).

4.7.2 Создание и настройка задач типа Custom

4.7.2.1 Создание обработчика задач типа Custom

Обработчик задач типа **Custom** (далее кастомный обработчик задач) представляет собой набор классов Java, которые упакованы в jar-архив. Кастомный обработчик задач должен:

- реализовывать интерфейс **ru.inrights.task.api.TaskHandler**, помеченный аннотацией **org.springframework.stereotype.Component**.
- содержать уникальный **handlerURI**, по которому будут связываться обработчики и задачи. При конфликте **handlerURI** последний зарегистрированный обработчик заменит собой все предыдущие.

Для создания обработчика необходимо подключение модулей, от работы которых данный обработчик будет зависеть (как минимум, это модуль **task-api-{app_version}.jar** приложения **admin-gui.war**).

В обработчике необходимо указать **TaskManager** и зарегистрироваться в нем в качестве обработчика задач с заданным **handlerURI**:

```
@Component
class CustomTaskHandler {
    public static final String HANDLER_URI =
        "testURI";

    @Autowired(required=true)
    private TaskManager taskManager;

    @PostConstruct
    private void initialize() {
        taskManager.registerHandler(HANDLER_URI,
this);
    }
}
```

Можно подключать все необходимые для работы бины из контекста **spring**. Кроме того, кастомный обработчик должен иметь наименование в контексте **spring**, которое можно задать явно в поле **value** аннотации **@Component** (например, **value = "**

InrightsConfiguration"). Если наименование не указано, то в качестве имени будет использоваться наименование класса.

ВАЖНО! Наименование кастомного обработчика заданий в контексте **spring** должно быть уникальным.

Собранный jar-архив необходимо скопировать в каталог **INRIGHTS_HOME/libs**, откуда он будет обработан в момент запуска приложения.

4.8 Настройка таймаута сессии пользователя

Администратор ПС может настраивать таймаут сессии пользователя приложений **пользовательского** и **admin-gui**. По истечению таймаута, заданного в настройках, пользователь автоматически перенаправляется на страницу авторизации.

Чтобы настроить таймаут сессии пользователя, следует:

- 1) На сервере приложений **idm / admin-gui** выполнить команду:

```
c:\tomcat\tomcat8_inrights\bin\tomcat8w  
//ES//INRIGHTS
```

- 2) В окне редактора свойств сервиса перейти на вкладку Java и в поле **Java Options** прописать ключ **Dinrights.httpSession.timeout=PT1M** (см. Рис. 90). Его значением является время жизни сессии в определенном формате.

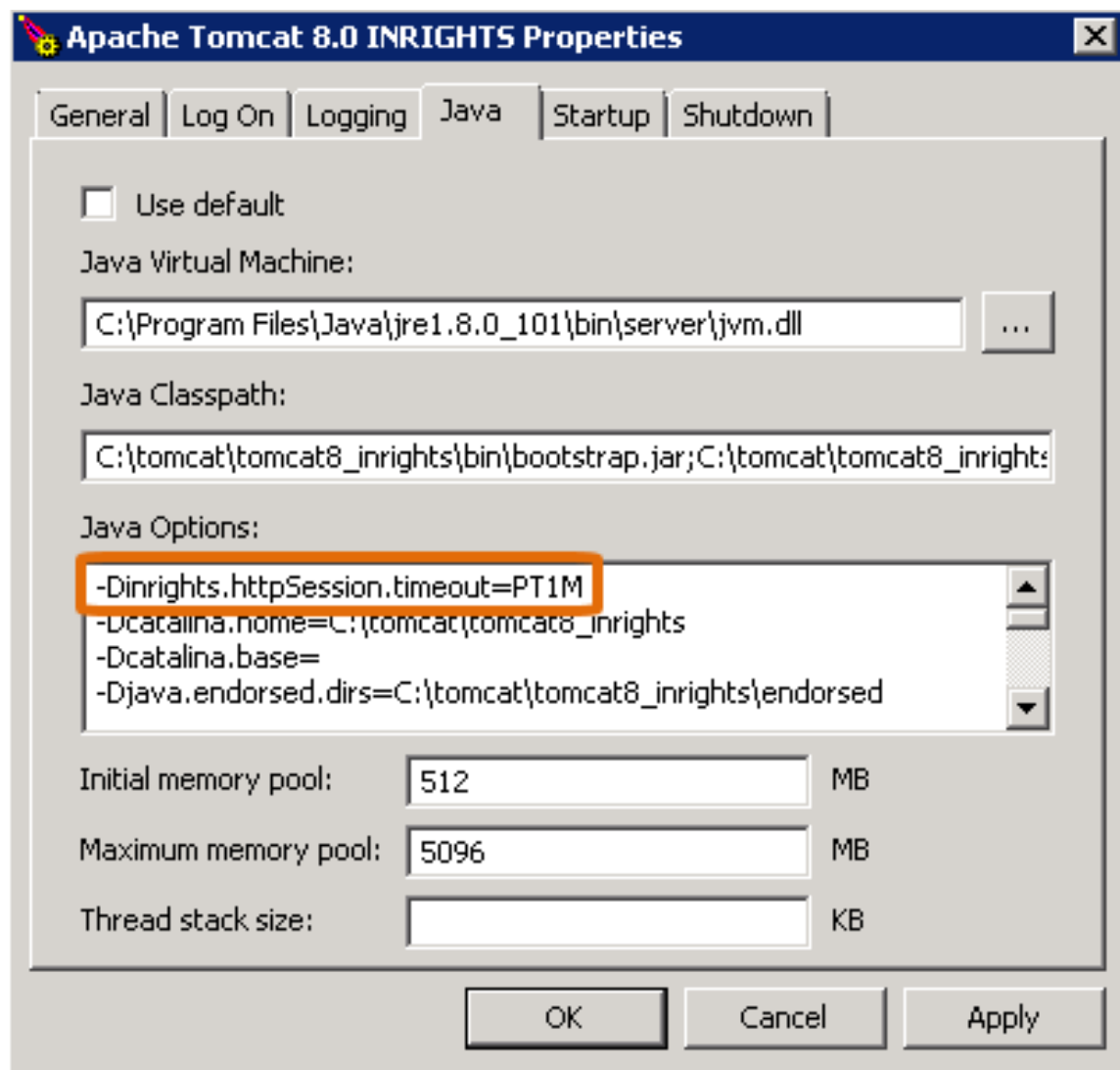


Рис. 90 – Настройка таймаута сессии

Примеры записи значения **Dinrights.httpSession.timeout**:

- PT20.345S – 20.345 seconds;
- PT15M – 15 minutes;
- PT10H – 10 hours;
- P2D – 2 days;
- P2DT3H4M – 2 days, 3 hours and 4 minutes.

4.9 Настройка журналирования событий информационной безопасности

4.9.1 Общие сведения

События информационной безопасности в ПС регистрируются с помощью syslog. Можно настроить запись журналов как в syslog, так и в отдельный файл. По умолчанию настройки записи отключены.

После настройки syslog каждый день записывается новый файл журнала, к имени файла добавляется дата его создания. Администратор определяет срок хранения файла журнала, а также события, которые необходимо регистрировать (по умолчанию записываются все события). Настройка журналирования описана в разделе 4.9.2.

ПС регистрирует следующие события:

- **Вход и выход из системы** (Успешный вход – введены верные логин и пароль, введены верные логин и пароль подключенной системы; Не удалось выполнить вход – пользователь заблокирован, неверный пароль, неверный логин; Выход выполнен по команде пользователя, по таймауту, из-за штатного останова сервера);
- **Операции с ресурсными объектами** (Ресурсный объект создан, удален, заблокирован, разблокирован; К ресурсному объекту добавлена ассоциация/отозвана ассоциация; Пароль/атрибуты ресурсного объекта изменены);
- **Операции с пользователями** (Пользователь создан, удален, заблокирован, разблокирован; Пароль пользователя изменен; Руководитель пользователя изменен; Пользователь включен в подразделение/исключен из подразделения; Shadow привязана к пользователю/отвязана от пользователя; Атрибуты пользователя изменены);
- **Операции с трудоустройствами** (Трудоустройство создано/удалено/заблокировано/разблокировано; Руководитель трудоустройства изменен; Трудоустройство включено в подразделение/исключено из подразделения; Атрибуты трудоустройства изменены);

- **Операции с назначениями** (Назначение создано/удалено; Срок действия назначения изменен);
- **Операции с ролями** (Роль создана/удалена; Inducement включен в роль/исключен из роли; Роль добавлена/удалена из каталога; Владелец/Атрибуты роли изменены);
- **Операции с подразделениями** (Подразделение создано/удалено; Подразделение включено в родительское подразделение/исключено из родительского подразделения; Базовая роль добавлена к подразделению/исключена из подразделения; Атрибуты подразделения изменены);
- **Операции с должностями** (Должность создана/удалена; Должность включена в родительское подразделение/исключена из родительского подразделения; Базовая роль добавлена к должности/исключена из должности; Атрибуты должности изменены);
- **Операции с каталогами ролей** (Каталог ролей создан/удален; Атрибуты каталога изменены);
- **Операции с системными объектами** (Информационная система создана/удалена; Владелец/Атрибуты информационной системы изменены; Коннектор-хост создан/удален; Атрибуты коннектор-хоста изменены);
- **Операции с серверными задачами** (Серверная задача создана/удалена/приостановлена/возобновлена; Атрибуты серверной задачи изменены; Серверная задача отправлена на немедленное выполнение/запущена/завершена);
- **Операции с заявками** (Заявка/Запрос созданы; Запущен/завершен процесс согласования; Произошла ошибка вычисления согласующего/ исполнителя/исполнения; Срок шага согласования/исполнения истек; Шаг процесса согласования/исполнения завершен/пропущен; Запрос/Заявка завершены);

- **Факты доступа к объектам** (Отказ в доступе/Успешное обращение к объекту системы);
- **Интеграция с системами** (Операции на коннекторах Create/DeleteNORMAL/RunScript/AddAttributeValues/RemoveAttributeValues/Update);
- **Операции с отчетами** (Шаблон отчета создан/удален/изменен; Отчет по шаблону сформирован; Отчет экспортирован).

Подробное описание событий и их атрибутов приведено в приложении документа (см. Приложение А. Описание событий информационной безопасности).

4.9.2 Настройка журналирования событий в syslog или определенный файл

Для настройки журналирования событий ИБ в **syslog**/файл следует:

- 1) Перейти в раздел **Объекты из хранилища (Конфигурирование > Объекты в репозитории)**.
- 2) Выбрать группу объектов **Конфигурация**.
- 3) Отредактировать xml-описание объекта **Syslog Outbound Sink / File Outbound Sink**, задав необходимые параметры и их значения (см. Таблица 17 –).

Объекты из хранилища

Конфигурация

☐ Название	Читаемое имя	Экспорт	Удалить
☐ [SystemData] Security Extensions		Экспорт	Удалить
☐ Audit Outbound Sink		Экспорт	Удалить
☐ File Outbound Sink		Экспорт	Удалить
☐ Login Transliteration Configuration		Экспорт	Удалить
☐ Notification Outbound Sink		Экспорт	Удалить
☐ Project Configuration		Экспорт	Удалить
☐ Request Configuration		Экспорт	Удалить
☐ Roles Configuration		Экспорт	Удалить
☐ Security Configuration		Экспорт	Удалить
☒ Syslog Outbound Sink		Экспорт	Удалить

Показаны с 1 по 10 из 10 найденных.

Object details for 'Syslog Outbound Sink'

Options ☐ Edit ☒ Protected by encryption ☐ Validate schema ☐ Save in raw mode
☐ Overwrite as is ☐ Reevaluate search filters

```
1 <configurationObject xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3" xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3" xmlns:icfs="http://in
2 <name>Syslog Outbound Sink</name>
3 <description>Конфигурация записи событий в Syslog</description>
4 <configuration xmlns:os="http://inrights.solarsecurity.ru/outboundsink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="os:OutboundSinkConfiguration">
5 <os:sink xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="os:SyslogSink">
6 <os:hostname>localhost</os:hostname>
7 <os:port>514</os:port>
8 <os:format>RFC_3164</os:format>
9 <os:protocol>UDP</os:protocol>
10 </os:sink>
11 <os:enabled>false</os:enabled>
12 <os:default/>
13 </configuration>
14 </configurationObject>
15
```

Рис. 91 – Настройка записи событий ИБ в syslog

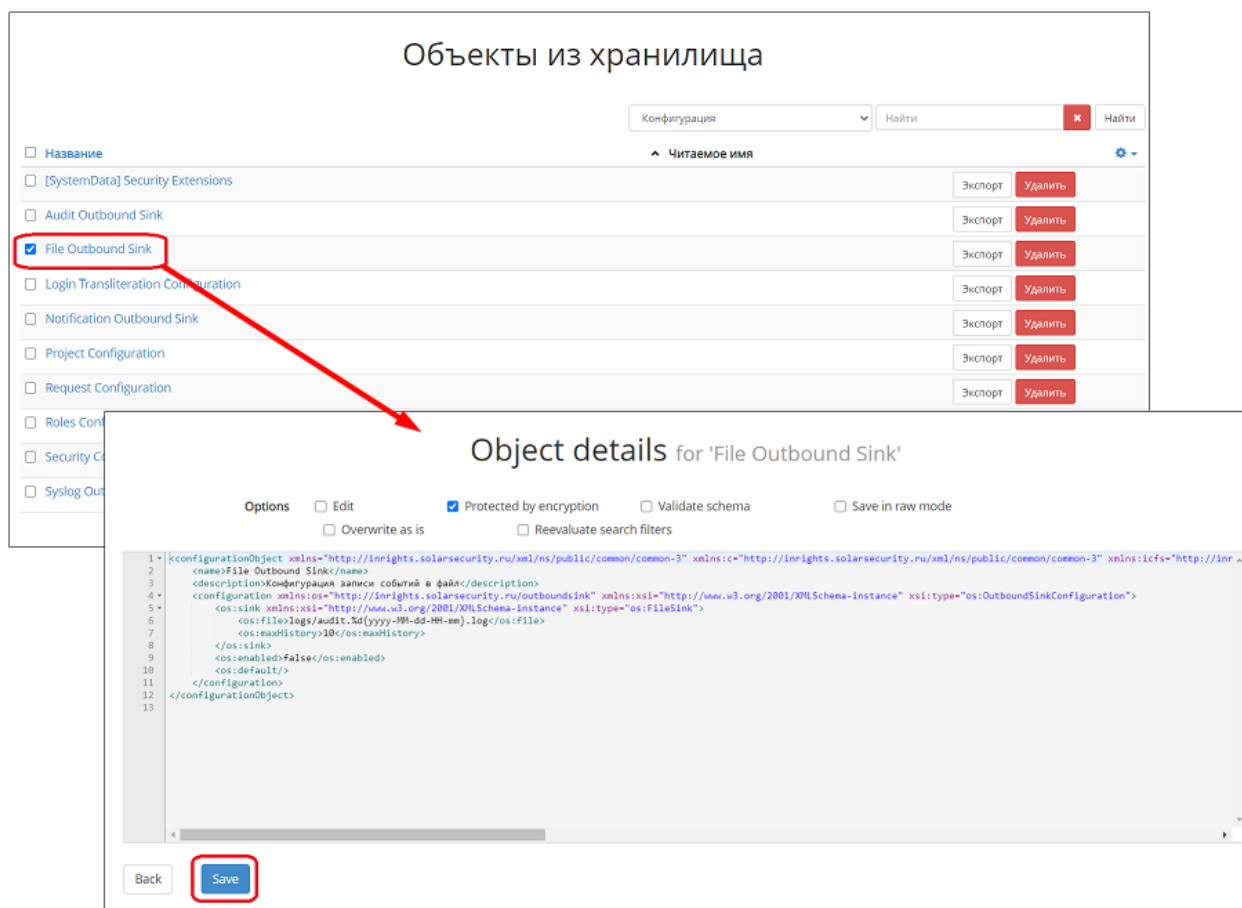


Рис. 92 – Настройка записи событий ИБ в файл

Таблица 17 – Параметры объектов Syslog Outbound Sink и File Outbound Sink

Параметр	Описание
Общие параметры	
enabled	активация (true) / деактивация (false) записи в syslog / файл. Значение по умолчанию - false
default	наличие этого параметра означает, что события будут записываться в формате, который принят по умолчанию
Параметры объекта Syslog Outbound Sink	
hostName	хост, на котором развернут syslog-сервер
port	порт, на котором syslog-сервер слушает входящие подключения
format	формат сообщения (RFC_3164 или RFC_5424)

protocol	протокол UDP или TCP
Параметры объекта File Outbound Sink	
file	Путь к файлу на хосте, где развернут сервер inrights/adminGui. Если интерфейсы пользовательский и adminGui развернуты на разных хостах, то файл создастся на каждом из них, и фиксируемые события будут распределены между ними в зависимости от того, какой узел обрабатывает конкретное событие

Для настройки ротации журналов – срока хранения журналов событий следует в xml-описание объекта внутри тега <os:sink/> добавить строки:

```
<os:file>c:\audit.%d.log</os:file>
<os:maxHistory>10</os:maxHistory>
```

где значение параметра <os:file/> – имя файла журнала, в которое записывается дата его создания (%Y%m%d), например, **audit.2017-10-31.log**; <os:maxHistory/> – срок хранения файла журнала (в днях).

По умолчанию в журнал записываются все события. Чтобы исключить при записи журнала определенные события, следует внутри тега <configuration/> добавить строки:

```
<os:mapping>
    <os:id>employment.deleted</os:id>
    <os:enabled>false</os:enabled>
</os:mapping>
```

где значение параметра <os:id/> – идентификатор события, которое необходимо исключить при записи журнала. Список всех идентификаторов событий приведен в приложении Приложение А. Описание событий информационной безопасности.

Ниже приведен пример настройки срока хранения файла журнала 7 дней и исключения записи следующих событий – **Информационная система удалена (resource.deleted), Трудоустройство исключено**

из подразделения (employment.parentOrg.removed),
Трудоустройство удалено (employment.deleted):

```
<configurationObject
xmlns="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:q="http://prism.solarsecurity.ru/xml/ns/public/query-3"

xmlns:c="http://inrights.solarsecurity.ru/xml/ns/public/common/common-3"

xmlns:t="http://prism.solarsecurity.ru/xml/ns/public/types-3"

xmlns:icfs="http://inrights.solarsecurity.ru/xml/ns/public/connector/icf-1/resource-schema-3"

xmlns:ri="http://inrights.solarsecurity.ru/xml/ns/public/resource/instance-3"

oid="configuration.outboundsink.file"
version="0">
  <name>File Outbound Sink</name>
  <description>Конфигурация записи событий в
  файл</description>
  <configuration
xmlns:os="http://inrights.solarsecurity.ru/outbound
sink"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:type="os:OutboundSinkConfiguration">
  <os:sink xsi:type="os:FileSink">
    <os:file>outbound-sink-
events.log</os:file>
    <os:file>c:\audit.%d.log</os:file>
```

```

        <os:maxHistory>7</os:maxHistory>
    </os:sink>
    <os:enabled>false</os:enabled>
    <os:default/>
<os:mapping>
    <os:id>employment.deleted</os:id>
    <os:id>resource.deleted</os:id>
    <os:id>employment.parentOrg.removed</os:id>
    <os:enabled>false</os:enabled>
</os:mapping>
    </configuration>
</configurationObject>

```

4.9.3 Защита от использования журнала событий безопасности в качестве скрытого канала передачи информации

В случае наличия на сервере ПС программных или аппаратных закладок, журнал аудита может быть использован как скрытый канал передачи данных. Для организации скрытого канала может быть использован тот факт, что при настроенном аудите попыток входа пользователей ПС в журнал аудита записывается информация о времени попытки входа, учетной записи и успехе операции.

В случае использования журнала аудита в виде файла на сервере ПС или на других серверах, к файлам которым имеется сетевой доступ, закладка, имеющая доступ к файлам журнала, имеет возможность отследить появление соответствующей записи в журнале и предпринять определенные деструктивные действия. При этом передача данных от закладки к пользователю ПС невозможна.

Особая опасность данного канала заключается в том, что для его использования не требуется действующая учетная запись в ПС, т.к. сохраняются в том числе и неуспешные попытки входа.

Для нейтрализации данного скрытого канала необходимо использовать журналирование по протоколу syslog на другой сервер. Сетевой доступ к этому серверу со стороны сервера ПС должен быть ограничен протоколом syslog.

4.9.4 Использование журнала событий безопасности для контроля скрытых каналов передачи информации

В случае наличия на сервере ПС программных закладок, действия пользователей по изменению данных на карточках пользователей или подразделений, а также добавление комментариев к заявкам, могут быть использованы как скрытый канал передачи данных. Закладка, размещенная внутри ПС в виде пользовательской задачи или скриптовой библиотеки, либо имеющая доступ к БД, используемой ПС для хранения данных, может фиксировать изменения атрибутов пользователя или подразделения, а также добавление комментариев к заявке. Также данный канал может использоваться для передачи данных закладке, находящейся в управляемой системе, которая может таким образом получать команды нарушителя для выполнения несанкционированных действий в системе.

Для исключения возможности использования данного канала необходимо осуществлять мониторинг изменений атрибутов объектов с проверкой передаваемых данных при помощи системы обнаружения вторжений.

Система обнаружения вторжений, используемая в организации, должна отслеживать события изменения атрибутов объектов с объёмом данных выше определенного значения и осуществлять поиск в значениях атрибутов последовательностей символов, похожих на кодированные двоичные данные или зашифрованные данные.

Необходимо осуществлять контроль за следующими событиями информационной безопасности:

- **UserModifiedEventType** – проверять значения параметров **user.attribute.***;
- **OrgModifiedEventType** – проверять значения параметров **org.attribute.***;
- **RequestSubmittedEventType** – проверять значение параметра **request.description**;
- **ApprovalStepCompletedEventType**,
ExecutionStepCompletedEventType – проверять значения параметров **approverComment.***.

5 ПОЛУЧЕНИЕ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ

Техническая поддержка ПС осуществляется по следующим каналам связи:

- по электронной почте **info@greenatom.ru** в режиме 24x7x365 (круглосуточно, включая выходные и праздничные дни);
- по телефону + 7 (499) 949-49-19 (по рабочим дням с 10:00 до 18:00 (время московское)).

ПРИЛОЖЕНИЕ А. ОПИСАНИЕ СОБЫТИЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Все события информационной безопасности, которые регистрирует ПС, описаны в таблицах ниже.

Вход и выход из системы:

Таблица 18 – Успешный вход - введены верные логин и пароль (идентификатор события **logged.in.event**)

Атрибут	Описание	Пример
cat	категория события: login_success	login_success

node.host	узел, на который выполнен вход	
node.name	имя узла, на который выполнен вход	adminGuiNode
sessionId	идентификатор созданной сессии	251f6630-2661-4693-9bc9-8836a80ba60e
userName	имя пользователя, под которым выполнен вход	administrator
principal.host	узел, с которого выполнен вход	
authenticationType	тип аутентификации: PASSWORD	PASSWORD
authenticationDomain	система, через которую выполнен вход	solar.local

Таблица 19 – Успешный вход - введены верные логин и пароль подключенной системы (идентификатор события logged.in.event)

Атрибут	Описание	Пример
cat	категория события: login_success	login_success
node.host	узел, на который выполнен вход	
node.name	имя узла, на который выполнен вход	adminGuiNode
sessionId	идентификатор созданной сессии	251f6630-2661-4693-9bc9-8836a80ba60e
userName	имя пользователя, под которым выполнен вход	administrator
principal.host	узел, с которого выполнен вход	
authenticationType	тип аутентификации: PASSWORD	PASSWORD
authenticationDomain	система, через которую выполнен вход	ra.local

Таблица 20 – Успешный вход (идентификатор события logged.in.event)

Атрибут	Описание	Пример
cat	категория события: login_success	login_success
node.host	узел, на который выполнен вход	
node.name	имя узла, на который выполнен вход	adminGuiNode
sessionId	идентификатор созданной сессии	251f6630-2661-4693-9bc9-8836a80ba60e
userName	имя пользователя, под которым выполнен вход	administrator
principal.host	узел, с которого выполнен вход	
authenticationType	тип аутентификации	PASSWORD
authenticationDomain	система, через которую выполнен вход	ra.local

Таблица 21 – Не удалось выполнить вход - пользователь заблокирован (идентификатор login.command.failed)

Атрибут	Описание	Пример
cat	категория события: login_failed	login_failed
node.host	узел, на который была предпринята попытка входа	
node.name	имя узла, на который была предпринята попытка входа	adminGuiNode
failure	описание ошибки: user_disabled	user_disabled
userName	имя пользователя, под которым была предпринята попытка входа	administrator

principal.host	узел, с которого выполнен вход	
authenticationType	тип аутентификации	PASSWORD
authenticationDomain	система, через которую выполнен вход	ra.local

Таблица 22 – Не удалось выполнить вход - неверный пароль
(идентификатор login.command.failed)

Атрибут	Описание	Пример
cat	категория события: login_failed	login_failed
node.host	узел, на который была предпринята попытка входа	
node.name	имя узла, на который была предпринята попытка входа	adminGuiNode
failure	описание ошибки: invalid_password	invalid_password
userName	имя пользователя, под была предпринята попытка входа	administrator
principal.host	узел, с которого выполнен вход	
authenticationType	тип аутентификации: PASSWORD	PASSWORD
authenticationDomain	система, через которую выполнен вход	ra.local

Таблица 23 – Не удалось выполнить вход - неверный логин
(идентификатор login.command.failed)

Атрибут	Описание	Пример

cat	категория события: login_failed	login_failed
node.host	узел, на который была предпринята попытка входа	
node.name	имя узла, на который была предпринята попытка входа	adminGuiNode
failure	описание ошибки: invalid_username	invalid_username
userName	имя пользователя, под была предпринята попытка входа	administrator
principal.host	узел, с которого выполнен вход	
authenticationType	тип аутентификации: PASSWORD	PASSWORD
authenticationDomain	система, через которую выполнен вход	ra.local

Таблица 24 – Выход выполнен по команде пользователя
(идентификатор logged.out.event)

Атрибут	Описание	Пример
cat	категория события: logout_success	logout_success
node.host	узел, на который выполнен вход	
node.name	имя узла, на который выполнен вход	adminGuiNode
sessionId	идентификатор завершенной сессии	68dc523b-a209-429e- 81fd-3c8853d3d902
userName	имя пользователя завершенной сессии	administrator
reason	причина завершения сессии: userRequest	userRequest

Таблица 25 – Выход выполнен по таймауту (идентификатор logged.out.event)

Атрибут	Описание	Пример
cat	категория события: logout_success	logout_success
node.host	узел, на который выполнен вход	
node.name	имя узла, на который выполнен вход	adminGuiNode
sessionId	идентификатор завершенной сессии	68dc523b-a209-429e-81fd-3c8853d3d902
userName	имя пользователя завершенной сессии	administrator
reason	причина завершения сессии: timeout	timeout

Таблица 26 – Выход выполнен из-за штатного останова сервера (идентификатор logged.out.event)

Атрибут	Описание	Пример
cat	категория события: logout_success	logout_success
node.host	узел, на который выполнен вход	
node.name	имя узла, на который выполнен вход	adminGuiNode
sessionId	идентификатор завершенной сессии	68dc523b-a209-429e-81fd-3c8853d3d902
userName	имя пользователя завершенной сессии	administrator
reason	причина завершения сессии: shutdown	shutdown

Операции с ресурсными объектами:

Таблица 27 – Ресурсный объект создан (идентификатор resourceobject.create)

Атрибут	Описание	Пример
cat	категория события: resourceObject_created	resourceObject_created
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da-e10c8437b621
resource.name	имя ресурса	Орг.структура из HR

Таблица 28 – Ресурсный объект удален (идентификатор resourceobject.delete)

Атрибут	Описание	Пример
cat	категория события: resourceObject_deleted	resourceObject_deleted
resourceObject.oid	oid ресурсного объекта	baa6166f-fede-4f04-981e-250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da-e10c8437b621
resource.name	имя ресурса	Орг.структура из HR

Таблица 29 – Ресурсный объект заблокирован (идентификатор resourceobject.disabled)

Атрибут	Описание	Пример
cat	категория события: resourceObject_disabled	resourceObject_disabled
resourceObject.oid	oid ресурсного объекта	baa6166f-fede-4f04-981e-250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da-e10c8437b621
resource.name	имя ресурса	Орг. структура из HR

Таблица 30 – Ресурсный объект разблокирован (идентификатор resourceobject.enabled)

Атрибут	Описание	Пример
cat	категория события: resourceObject_enabled	resourceObject_enabled
resourceObject.oid	oid ресурсного объекта	baa6166f-fede-4f04-981e-250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default

resource.oid	oid ресурса	0b579196-f6f5-4e76-82da-e10c8437b621
resource.name	имя ресурса	Орг. структура из HR

Таблица 31 – К ресурсному объекту добавлена ассоциация (идентификатор resourceobject.association.added)

Атрибут	Описание	Пример
cat	категория события: resourceObject_association_added	resourceObject_association_added
resourceObject.oid	oid ресурсного объекта	baa6166f-fede-4f04-981e-250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da-e10c8437b621
resource.name	имя ресурса	Орг. структура из HR
association.name	тип добавленной ассоциации	group
association.ref.oid	oid ассоциированного ресурсного объекта	8790e490-326a-46e9-ba35-9e0c1dcbb41d
association.ref.name	имя ассоциированного ресурсного объекта	Exchange
association.ref.resourc e.oid	oid ресурса, на котором находится ассоциированный ресурсный объект	37c21384-6996-45eb-8b27-3820e35c5004
association.ref.resourc e.name	имя ресурса, на котором находится ассоциированный ресурсный объект	Active directory (второй домен)

Таблица 32 – У ресурсного объекта отозвана ассоциация
(идентификатор resourceobject.association.deleted)

Атрибут	Описание	Пример
cat	категория события: resourceObject_association_removed	resourceObject_association_removed
resourceObject.oid	oid ресурсного объекта	baa6166f-fede-4f04-981e-250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da-e10c8437b621
resource.name	имя ресурса	Орг. структура из HR
association.name	тип добавленной ассоциации	group
association.ref.oid	oid ассоциированного ресурсного объекта	8790e490-326a-46e9-ba35-9e0c1dcbb41d
association.ref.name	имя ассоциированного ресурсного объекта	Exchange
association.ref.resource.oid	oid ресурса, на котором находится ассоциированный ресурсный объект	37c21384-6996-45eb-8b27-3820e35c5004
association.ref.resource.name	имя ресурса, на котором находится ассоциированный ресурсный объект	Active directory (второй домен)

Таблица 33 – Пароль ресурсного объекта изменен (идентификатор resourceobject.password.changed)

Атрибут	Описание	Пример
cat	категория события: resourceObject_password_cha nged	resourceObject_password_cha nged
resourceObject. oid	oid ресурсного объекта	baa6166f-fede-4f04-981e- 250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i
resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da- e10c8437b621
resource.name	имя ресурса	Active Directory

Таблица 34 – Атрибуты ресурсного объекта изменены (идентификатор resourceobject.attributes.changed)

Атрибут	Описание	Пример
cat	категория события: resourceOb ject_attribut es_change d	resourceObject_attributes_changed
resourceObject.oid	oid ресурсного объекта	baa6166f-fede-4f04-981e- 250079953350
resourceObject .name	имя ресурсного объекта	ivanov.i.i

resourceObject .objectClass	objectClass ресурсного объекта	ri:AccountObjectClass
resourceObject .kind	kind ресурсного объекта	account
resourceObject .intent	intent ресурсного объекта	default
resource.oid	oid ресурса	0b579196-f6f5-4e76-82da- e10c8437b621
resource.name	имя ресурса	Орг. структура из HR
resourceObject.attribute.[attr]. old	предыдущ ее значение атрибута [attr]	resourceObject.attribute.description. old=Стажёр
resourceObject.attribute.[attr]. new	новое значение атрибута [attr]	resourceObject.attribute.description. new=Стажёр
resourceObject.attribute.[attr]. added	список новых значений атрибута [attr]	resourceObject.attribute.phone.adde d=[123, 124]
resourceObject.attribute.[attr]. deleted	список удаленных значений атрибута [attr]	resourceObject.attribute.phone.delet ed=[123, 124]

Операции с пользователями:

Таблица 35 – Новый пользователь создан (идентификатор user.created)

Атрибут	Описание	Пример
cat	категория user_created события:	user_created

user.oid	oid созданного пользователя	baa6166f-fede-4f04-981e- 250079953350
user.name	имя созданного пользователя	ivanov.i.i
user.fullName	полное имя созданного пользователя	Иванов Иван Иванович

Таблица 36 – Пользователь удален (идентификатор user.deleted)

Атрибут	Описание	Пример
cat	категория события: user_deleted	user_deleted
user.oid	oid удаленного пользователя	baa6166f-fede-4f04-981e- 250079953350
user.name	имя удаленного пользователя	ivanov.i.i
user.fullName	полное имя удаленного пользователя	Иванов Иван Иванович

Таблица 37 – Пароль пользователя изменен (идентификатор user.password.changed)

Атрибут	Описание	Пример
cat	категория события: user_password_changed	user_password_changed
user.oid	oid пользователя	baa6166f-fede-4f04-981e- 250079953350
user.name	имя пользователя	ivanov.i.i
user.fullName	полное имя пользователя	Иванов Иван Иванович

Таблица 38 – Пользователь заблокирован (идентификатор user.disabled)

Атрибут	Описание	Пример
cat	категория события: user_disabled	user_disabled
user.oid	oid пользователя	baa6166f-fede-4f04-981e-250079953350
user.name	имя пользователя	ivanov.i.i
user.fullName	полное имя пользователя	Иванов Иван Иванович

Таблица 39 – Пользователь разблокирован (идентификатор user.enabled)

Атрибут	Описание	Пример
cat	категория события: user_enabled	user_enabled
user.oid	oid пользователя	baa6166f-fede-4f04-981e-250079953350
user.name	имя пользователя	ivanov.i.i
user.fullName	полное имя пользователя	Иванов Иван Иванович

Таблица 40 – Руководитель пользователя изменен (идентификатор user.manager.changed)

Атрибут	Описание	Пример
cat	категория события: user_manager_changed	user_manager_changed
user.oid	oid пользователя	baa6166f-fede-4f04-981e-250079953350
user.name	имя пользователя	ivanov.i.i
user.fullName	полное имя пользователя	Иванов Иван Иванович
user.manager.oid.old	oid предыдущего руководителя	45bf166f-fede-4f04-981e-250079957a3c

user.manager.name.old	имя предыдущего руководителя	volkov.n.s
user.manager.fullName.old	полное имя предыдущего руководителя	Волков Николай Сергеевич
user.manager.oid.new	oid нового руководителя	da2e166f-fede-4f04-981e- 25007995ad12
user.manager.name.new	имя нового руководителя	kurochkin.m.v
user.manager.fullName.new	полное имя нового руководителя	Курочкин Михаил Валерьевич

Таблица 41 – Пользователь включен в подразделение
(идентификатор user.parentOrg.added)

Атрибут	Описание	Пример
cat	категория события: user_parentOrg_added	user_parentOrg_added
user.oid	oid пользователя	baa6166f-fede-4f04-981e- 250079953350
user.name	имя пользователя	ivanov.i.i
parentOrg.oid	oid подразделения	0b579196-f6f5-4e76-82da- e10c8437b621
parentOrg.name	имя подразделения	Департамент продаж

Таблица 42 – Пользователь исключен из подразделения
(идентификатор user_parentOrg_removed)

Атрибут	Описание	Пример
cat	категория события: user_parentOrg_removed	user_parentOrg_removed
user.oid	oid пользователя	baa6166f-fede-4f04-981e- 250079953350
user.name	имя пользователя	ivanov.i.i

parentOrg.oid	oid подразделения	0b579196-f6f5-4e76-82da-e10c8437b621
parentOrg.name	имя подразделения	Департамент продаж

Таблица 43 – Shadow привязана к пользователю (идентификатор user_shadow_linked)

Атрибут	Описание	Пример
cat	категория события: user_shadow_linked	user_shadow_linked
user.oid	oid привязанной shadow	0b579196-f6f5-4e76-82da-e10c8437b621
user.name	имя пользователя	ivanov.i.i
shadow.name	имя привязанной shadow	ivanov.i.i
shadow.oid	oid подразделения	0b579196-f6f5-4e76-82da-e10c8437b621
shadow.kind	kind привязанной shadow	account
shadow.intent	intent привязанной shadow	default

Таблица 44 – Shadow отвязана от пользователя (идентификатор user_shadow_unlinked)

Атрибут	Описание	Пример
cat	категория события: user_shadow_unlinked	user_shadow_unlinked
user.oid	oid пользователя	baa6166f-fede-4f04-981e-250079953350
user.name	имя пользователя	ivanov.i.i
shadow.name	имя привязанной shadow	ivanov.i.i
shadow.oid	oid подразделения	0b579196-f6f5-4e76-82da-e10c8437b621
shadow.kind	kind привязанной shadow	account

shadow.intent	intent привязанной shadow	default

Таблица 45 – Атрибуты пользователя изменены (идентификатор user_attributes_changed)

Атрибут	Описание	Пример
cat	категория события: user_attributes_changed	user_attributes_changed
user.oid	oid созданного пользователя	baa6166f-fede-4f04-981e-250079953350
user.name	имя созданного пользователя	ivanov.i.i
user.fullName	полное имя созданного пользователя	Иванов Иван Иванович
user.attribute.[attr].old	старое значение атрибута [attr]	user.attribute.name.old=ivanov.i.i
user.attribute.[attr].new	новое значение атрибута [attr]	user.attribute.name.new=petrov.p.p

Операции с трудоустройствами:

Таблица 46 – Трудоустройство создано (идентификатор employment_created)

Атрибут	Описание	Пример
cat	категория события: employment_created	employment_created
employment.oid	oid трудоустройства	baa6166f-fede-4f04-981e-250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da-e10c8437b621
user.name	имя связанного пользователя	ivanov.i.i

user.fullName	полное имя связанного пользователя	Иванов Иван Иванович

Таблица 47 – Трудоустройство удалено (идентификатор employment_deleted)

Атрибут	Описание	Пример
cat	категория события: employment_deleted	employment_deleted
employment.oid	oid трудоустройства	baa6166f-fede-4f04-981e-250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da-e10c8437b621
user.name	имя связанного пользователя	ivanov.i.i
user.fullName	полное имя связанного пользователя	Иванов Иван Иванович

Таблица 48 – Трудоустройство заблокировано (идентификатор employment_disabled)

Атрибут	Описание	Пример
cat	категория события: employment_disabled	employment_disabled
employment.oid	oid трудоустройства	baa6166f-fede-4f04-981e-250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da-e10c8437b621
user.name	имя связанного пользователя	ivanov.i.i
user.fullName	полное имя связанного пользователя	Иванов Иван Иванович

Таблица 49 – Трудоустройство разблокировано (идентификатор employment_enabled)

Атрибут	Описание	Пример
cat	категория события: employment_enabled	employment_enabled
employment.oid	oid трудоустройства	baa6166f-fede-4f04-981e-250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da-e10c8437b621
user.name	имя связанного пользователя	ivanov.i.i
user.fullName	полное имя связанного пользователя	Иванов Иван Иванович

Таблица 50 – Руководитель трудоустройства изменен (идентификатор employment_manager_changed)

Атрибут	Описание	Пример
cat	категория события: employment_manager_changed	employment_manager_changed
employment.oid	oid трудоустройства	baa6166f-fede-4f04-981e-250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da-e10c8437b621
user.name	имя связанного пользователя	ivanov.i.i
user.fullName	полное имя связанного пользователя	Иванов Иван Иванович

employment.manager.old.oid	oid предыдущего руководителя	3c579133-f6f5-4e76-82da- e10c8437f2e4
employment.manager.old.name	имя предыдущего руководителя	samarin.v.e
employment.manager.old.fullName	полное имя предыдущего руководителя	Самарин Виктор Евгеньевич
employment.manager.new.oid	oid нового руководителя	fe359133-f6f5-4e76-82da- e10c843712ae
employment.manager.new.name	имя нового руководителя	korolev.s.n
employment.manager.new.fullName	полное имя нового руководителя	Королев Сергей Николаевич

Таблица 51 – Трудоустройство включено в подразделение
(идентификатор employment_parentOrg_added)

Атрибут	Описание	Пример
cat	категория события: employment_parentOrg_added	employment_parentOrg_added
employment.oid	oid трудоустройства	baa6166f-fede-4f04-981e- 250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da- e10c8437b621
user.name	имя связанного пользователя	ivanov.i.i
parentOrg.oid	oid подразделения	0b579196-f6f5-4e76-82da- e10c8437b621
parentOrg.name	имя подразделения	Департамент продаж

Таблица 52 – Трудоустройство исключено из подразделения
(идентификатор employment_parentOrg_removed)

Атрибут	Описание	Пример
cat	категория события: employment_parentOrg_remove d	employment_parentOrg_remove d
employment.oid	oid связанного трудоустройства	baa6166f-fede-4f04-981e- 250079953350
user.oid	oid связанного пользователя	0b579196-f6f5-4e76-82da- e10c8437b621
user.name	имя пользователя	ivanov.i.i
parentOrg.oid	oid подразделения	0b579196-f6f5-4e76-82da- e10c8437b621
parentOrg.name	имя подразделения	Департамент продаж

Таблица 53 – Атрибуты трудоустройства изменены (идентификатор user_attributes_changed)

Атрибут	Описание	Пример
cat	категория события: user_attributes_change d	user_attributes_changed
user.oid	oid связанного пользователя	baa6166f-fede-4f04-981e- 250079953350
user.name	имя связанного пользователя	ivanov.i.i
user.fullName	полное имя связанного пользователя	Иванов Иван Иванович
user.attribute.[attr].old	старое значение атрибута [attr]	user.attribute.name.old=ivanov.i.i
user.attribute.[attr].new	новое значение атрибута [attr]	user.attribute.name.new=petrov.p.p

Операции с назначениями:

Таблица 54 – Назначение создано (идентификатор assignment.created)

Атрибут	Описание	Пример
cat	категория события: role_assigned / position_assigned / substitute_assigned	role_assigned
object.role.oid	oid назначенной роли	28e8d402-4a64-46c1-ac12-1be783a8a487
object.role.name	имя назначенной роли	Доступ в Google
object.position.oid	oid назначенной должности	87b2f9eb-d237-4893-9008-e2628763e22a
object.position.name	имя назначенной должности	Старший работник отдела по связям с общественностью
object.substitute.oid	oid назначенного заместителя	f326acc4-e1f4-4697-8617-844a396f5e7a
object.substitute.name	имя назначенного заместителя	pushkin.v.g
object.substitute.fullName	полное имя назначенного заместителя	Пушкин Вениамин Григорьевич
subject.user.oid	oid пользователя, которому назначен object	52586920-2eb7-46dd-bfaa-e2fc591f06fd
subject.user.name	имя пользователя, которому назначен object	ivanov.a.s
subject.user.fullName	полное имя пользователя, которому назначен object	Иванов Александр Сергеевич
subject.employment.oid	oid трудоустройства, которому назначен object	e987939a-f0c2-4cc7-9a8a-29dc0308d525

subject.employment.user.oid	oid пользователя, связанного с трудоустройством, которому назначен object	00000000-0000-0000-0000-000000000002
subject.employment.user.name	имя пользователя, связанного с трудоустройством, которому назначен object	petrov.s.s
subject.employment.user.fullName	полное имя пользователя, связанного с трудоустройством, которому назначен object	Петров Александр Александрович
subject.role.oid	oid роли, которой назначен object	4898ee32-53bc-4d28-bc4a-1847917d81ea
subject.role.name	имя роли, которой назначен object	Доступ к 1С
subject.org.oid	oid организации, которому назначен object	00797cd7-4b45-4a08-9113-c997d33e65a5
subject.org.name	имя организации, которому назначен object	Отдел закупок
validFrom	время начала действия назначения	2016-11-02T10:42:37.968+03:00
validTo	время окончания действия назначения	2017-09-15T12:13:38.182+03:00

Таблица 55 – Назначение удалено (идентификатор assignment.deleted)

Атрибут	Описание	Пример
cat	категория события: role_unassigned / position_unassigned /	role_unassigned

	substitute_unassigned	
object.role.oid	oid отозванной роли	28e8d402-4a64-46c1-ac12-1be783a8a487
object.role.name	имя отозванной роли	Доступ в Google
object.position.oid	oid отозванной должности	87b2f9eb-d237-4893-9008-e2628763e22a
object.position.name	имя отозванной должности	Старший работник отдела по связям с общественностью
object.substitute.oid	oid отозванного заместителя	f326acc4-e1f4-4697-8617-844a396f5e7a
object.substitute.name	имя отозванного заместителя	pushkin.v.g
object.substitute.fullName	полное имя отозванного заместителя	Пушкин Вениамин Григорьевич
subject.user.oid	oid пользователя, у которого отозван object	52586920-2eb7-46dd-bfaa-e2fc591f06fd
subject.user.name	имя пользователя, у которого отозван object	ivanov.a.s
subject.user.fullName	полное имя пользователя, у которого отозван object	Иванов Александр Сергеевич
subject.employment.oid	oid трудоустройства, у которого отозван object	e987939a-f0c2-4cc7-9a8a-29dc0308d525
subject.employment.user.oid	oid пользователя, связанного с трудоустройством, у которого отозван object	00000000-0000-0000-0000-000000000002
subject.employment.user.name	имя пользователя, связанного с трудоустройством,	petrov.s.s

	у которого отозван object	
subject.employment.user.fullName	полное имя пользователя, связанного с трудоустройством, у которого отозван object	Петров Александр Александрович
subject.role.oid	oid роли, у которой отозван object	4898ee32-53bc-4d28- bc4a-1847917d81ea
subject.role.name	имя роли, у которой отозван object	Доступ к 1С
subject.org.oid	oid организации, у которой отозван object	00797cd7-4b45-4a08- 9113-c997d33e65a5
subject.org.name	имя организации, у которой отозван object	Отдел закупок
validFrom	время начала действия назначения	2016-11- 02T10:42:37.968+03:0 0
validTo	время окончания действия назначения	2017-09- 15T12:13:38.182+03:0 0

Таблица 56 – Срок действия назначения изменен (идентификатор assignment.validity.changed)

Атрибут	Описание	Пример
cat	категория события: role_assignment_validity_ch anged / position_assignment_validit y_changed / substitute_assignment_validi ty_changed	position_assignment_validi ty_changed
object.role.oid	oid назначенной роли	28e8d402-4a64-46c1- ac12-1be783a8a487
object.role.name	имя назначенной роли	Доступ в Google

object.position.oid	oid назначенной должности	87b2f9eb-d237-4893-9008-e2628763e22a
object.position.name	имя назначенной должности	Старший работник отдела по связям с общественностью
object.substitute.oid	oid назначенного заместителя	f326acc4-e1f4-4697-8617-844a396f5e7a
object.substitute.name	имя назначенного заместителя	pushkin.v.g
object.substitute.fullName	полное имя назначенного заместителя	Пушкин Вениамин Григорьевич
subject.user.oid	oid пользователя, которому назначен object	52586920-2eb7-46dd-bfaa-e2fc591f06fd
subject.user.name	имя пользователя, которому назначен object	ivanov.a.s
subject.user.fullName	полное имя пользователя, которому назначен object	Иванов Александр Сергеевич
subject.employment.oid	oid трудоустройства, которому назначен object	e987939a-f0c2-4cc7-9a8a-29dc0308d525
subject.employment.user.oid	oid пользователя связанного с трудоустройством, которому назначен object	00000000-0000-0000-0000-000000000002
subject.employment.user.name	имя пользователя связанного с трудоустройством, которому назначен object	petrov.a.a
subject.employment.user.fullName	полное имя пользователя связанного с трудоустройством, которому назначен object	Петров Александр Александрович
subject.role.oid	oid роли, которой назначен object	4898ee32-53bc-4d28-bc4a-1847917d81ea
subject.role.name	имя роли, которой назначен object	Доступ к 1С
subject.org.oid	oid организации, которому назначен object	00797cd7-4b45-4a08-9113-c997d33e65a5

subject.org.name	имя организации, которому назначен object	Отдел закупок
validFrom.old	предыдущее время начала действия назначения	2016-11- 02T10:42:37.968+03:00
validFrom.new	новое время начала действия назначения	
validTo.old	предыдущее время окончания действия назначения	2017-09- 15T12:13:38.182+03:00
validTo.new	новое время окончания действия назначения	

Операции с ролями:

Таблица 57 – Роль создана (идентификатор role_create)

Атрибут	Описание	Пример
cat	категория события: role_created	role_created
role.oid	oid роли	baa6166f-fede-4f04-981e- 250079953350
role.name	имя роли	Доступ в интернет

Таблица 58 – Роль удалена (идентификатор role_delete)

Атрибут	Описание	Пример
cat	категория события: role_deleted	role_deleted
role.oid	oid роли	baa6166f-fede-4f04-981e- 250079953350
role.name	имя роли	Доступ в интернет

Таблица 59 – Inducement включен в роль (идентификатор role_inducement_add)

Атрибут	Описание	Пример
cat	категория события: role_inducement_added	role_inducement_added
role.oid	oid роли	baa6166f-fede-4f04-981e-250079953350
role.name	имя роли	Доступ в интернет
role.inducement.target.oid	oid включаемой роли	0b579196-f6f5-4e76-82da-e10c8437b621
role.inducement.target.name	имя включаемой роли	Доступ к Яндекс
role.inducement.construction.resource.oid	oid ресурса, на котором создается ресурсный объект	87ad6c5f-f6f5-4e76-82da-e10c8437b621
role.inducement.construction.resource.name	имя ресурса, на котором создается ресурсный объект	Active Directory
role.inducement.construction.kind	kind создаваемого ресурсного объекта	account
role.inducement.construction.intent	intent создаваемого ресурсного объекта	default

Таблица 60 – Inducement исключен из роли (идентификатор role_inducement_remove)

Атрибут	Описание	Пример
cat	категория события: role_inducement_removed	role_inducement_removed
role.oid	oid роли	baa6166f-fede-4f04-981e-250079953350
role.name	имя роли	Доступ в интернет

role.inducement.target.oid	oid включаемой роли	0b579196-f6f5-4e76-82da-e10c8437b621
role.inducement.target.name	имя включаемой роли	Доступ к Яндексу
role.inducement.construction.resource.oid	oid ресурса, на котором создается ресурсный объект	87ad6c5f-f6f5-4e76-82da-e10c8437b621
role.inducement.construction.resource.name	имя ресурса, на котором создается ресурсный объект	Active Directory
role.inducement.construction.kind	kind создаваемого ресурсного объекта	account
role.inducement.construction.intent	intent создаваемого ресурсного объекта	default

Таблица 61 – Владелец роли изменен (идентификатор role_owner_change)

Атрибут	Описание	Пример
cat	категория события: role_owner_changed	role_owner_changed
role.oid	oid роли	baa6166f-fede-4f04-981e-250079953350
role.name	имя роли	Доступ в интернет
role.owner.old.oid	oid прежнего владельца	87ad6c5f-f6f5-4e76-82da-e10c8437b621
role.owner.old.name	системное имя прежнего владельца	ivanov.i.i
role.owner.old.fullName	полное имя прежнего владельца	Иванов Игорь Ильич
role.owner.new.oid	oid нового владельца	0b579196-f6f5-4e76-82da-e10c8437b621
role.owner.new.name	системное имя нового владельца	petrov.p.p

role.owner.new.fullName	полное имя нового владельца	Петров Порфирий Павлович

Таблица 62 – Атрибуты роли изменены (идентификатор role_attributes_change)

Атрибут	Описание	Пример
cat	категория события: role_attributes_changed	role_attributes_changed
role.oid	oid роли	baa6166f-fede-4f04-981e- 250079953350
role.name	имя роли	Доступ к portalу
role.attribute.[attr].old	старое значение атрибута [attr]	role.attribute.name.old=Доступ к portalу
role.attribute.[attr].new	новое значение атрибута [attr]	role.attribute.name.new=Доступ к корпоративному portalу

Операции с должностями:

Таблица 63 – Новая должность создана (идентификатор position_create)

Атрибут	Описание	Пример
cat	категория события: position_created	position_created
position.oid	oid должности	baa6166f-fede-4f04-981e- 250079953350
position.name	имя должности	Руководитель отдела PR

Таблица 64 – Должность удалена (идентификатор position_delete)

Атрибут	Описание	Пример
cat	категория события: position_deleted	position_deleted

position.oid	oid должности	baa6166f-fede-4f04-981e-250079953350
position.name	имя должности	Руководитель отдела PR

Таблица 65 – Должность включена в родительское подразделение (идентификатор position_parentOrg_added)

Атрибут	Описание	Пример
cat	категория события: position_parentOrg_added	position_parentOrg_added
position.oid	oid должности	baa6166f-fede-4f04-981e-250079953350
position.name	имя должности	Руководитель отдела PR
parentOrg.oid	oid родительского подразделения	0b579196-f6f5-4e76-82da-e10c8437b621
parentOrg.name	имя родительского подразделения	Департамент маркетинга

Таблица 66 – Должность исключена из родительского подразделения (идентификатор position.parentOrg.removed)

Атрибут	Описание	Пример
cat	категория события: position.parentOrg.removed	position.parentOrg.removed
position.oid	oid должности	baa6166f-fede-4f04-981e-250079953350
position.name	имя должности	Руководитель отдела PR
parentOrg.oid	oid родительского подразделения	0b579196-f6f5-4e76-82da-e10c8437b621
parentOrg.name	имя родительского подразделения	Департамент маркетинга

Таблица 67 – Базовая роль добавлена к должности (идентификатор position_contextAssignment_added)

Атрибут	Описание	Пример
cat	категория события: position_contextAssignment_added	position_contextAssignment_added
position.oid	oid должности	baa6166f-fede-4f04-981e-250079953350
position.name	имя должности	Уборщица
position.contextAsmt.target.oid	oid добавляемой базовой роли	0b579196-f6f5-4e76-82da-e10c8437b621
position.contextAsmt.target.name	имя добавляемой базовой роли	Доступ в Интернет

Таблица 68 – Базовая роль исключена из должности (идентификатор position_contextAssignment_deleted)

Атрибут	Описание	Пример
cat	категория события: position_contextAssignment_deleted	position_contextAssignment_deleted
position.oid	oid должности	baa6166f-fede-4f04-981e-250079953350

position.name	имя должности	Уборщица
position.contextAsmt.target.oid	oid исключаемой базовой роли	0b579196-f6f5-4e76-82da-e10c8437b621
position.contextAsmt.target.name	имя исключаемой базовой роли	Доступ в Интернет

Таблица 69 – Атрибуты должности изменены (идентификатор position_attributes_changed)

Атрибут	Описание	Пример
cat	категория события: position_attributes_changed	position_attributes_changed
position.oid	oid должности	baa6166f-fede-4f04-981e-250079953350
position.name	имя должности	Уборщица
position.attribute.[attr].old	старое значение атрибута [attr]	org.attribute.name.old=Специалист
position.attribute.[attr].new	новое значение атрибута [attr]	org.attribute.name.new=Специалист по клирингу

Операции с системными объектами:

Таблица 70 – Новая информационная система создана (идентификатор resource.created)

Атрибут	Описание	Пример
cat	категория события: resource_created	resource_created
resource.oid	oid информационной системы	baa6166f-fede-4f04-981e-250079953350

resource.name	имя информационной системы	Active Directory

Таблица 71 – Новая информационная система создана (идентификатор resource.deleted)

Атрибут	Описание	Пример
cat	категория события: resource_deleted	resource_deleted
resource.oid	oid информационной системы	baa6166f-fede-4f04-981e-250079953350
resource.name	имя информационной системы	Active Directory

Таблица 72 – Владелец информационной системы изменен (идентификатор resource.owner.changed)

Атрибут	Описание	Пример
cat	категория события: resource_owner_change d	resource_owner_change d
resource.oid	oid информационной системы	baa6166f-fede-4f04-981e-250079953350
resource.name	имя информационной системы	Active Directory
resource.owner.old.oid	oid предыдущего владельца	87ad6c5f-f6f5-4e76-82da-e10c8437b621
resource.owner.old.name	имя предыдущего владельца	ivanov.i.i
resource.owner.old.fullName	полное имя предыдущего владельца	Иванов Игорь Ильич
resource.owner.new.oid	oid нового владельца	0b579196-f6f5-4e76-82da-e10c8437b621
resource.owner.new.name	name нового владельца	petrov.p.p

resource.owner.new.fullName	fullName владельца	нового	Петров Поликарпович	Павел

Таблица 73 – Атрибуты информационной системы изменены (идентификатор resource.attributes.changed)

Атрибут	Описание	Пример
cat	категория события: resource.attributes.changed	resource.attributes.changed
resource.oid	oid информационной системы	baa6166f-fede-4f04-981e-250079953350
resource.name	имя информационной системы	Active Directory
resource.attribute.[attr].old	новое значение атрибута [attr]	87ad6c5f-f6f5-4e76-82da-e10c8437b621
resource.owner.old.name	старое значение атрибута [attr]	org.attribute.name.old=Каталог
resource.owner.old.fullName	полное имя предыдущего владельца	Иванов Игорь Ильич
resource.attribute.[attr].new	новое значение атрибута [attr]	org.attribute.name.new=Каталог ролей доступа к Интернету
resource.sections	список секций, в которых произошли изменения (schemaHandling, capabilities, containerConfiguration, connectorRef, consistency, schema, scripts, synchronization)	[schema, schemaHandling]

Таблица 74 – Новый коннектор-хост создан (идентификатор connectorHost.created)

Атрибут	Описание	Пример

cat	категория события: connectorHost_created	connectorHost_created
connectorHost.oid	oid коннектор-хоста	baa6166f-fede-4f04-981e-250079953350
connectorHost.name	имя информационной системы	AD Domain Controller

Таблица 75 – Коннектор-хост удален (идентификатор connectorHost.deleted)

Атрибут	Описание	Пример
cat	категория события: connectorHost_deleted	connectorHost_deleted
connectorHost.oid	oid коннектор-хоста	baa6166f-fede-4f04-981e-250079953350
connectorHost.name	имя информационной системы	AD Domain Controller

Таблица 76 – Атрибуты коннектор-хоста изменены (идентификатор connectorHost.attributes.changed)

Атрибут	Описание	Пример
cat	категория события: connectorHost_attributes_changed	connectorHost.attributes.changed
connectorHost.oid	oid коннектор-хоста	baa6166f-fede-4f04-981e-250079953350
connectorHost.name	имя информационной системы	AD Domain Controller
connectorHost.attribute.[attr].old	старое значение	org.attribute.name.old=AD DC

	атрибута [attr]	
connectorHost.attribute.[attr].new	новое значение атрибута [attr]	org.attribute.name.new=AD Domain Controller

Операции с серверными задачами:

Таблица 77 – Новая серверная задача создана (идентификатор task.created)

Атрибут	Описание	Пример
cat	категория события: task_created	task_created
task.oid	oid созданной задачи	e02909bf-b97f-473b-a64b-a30e869be48b
task.name	имя созданной задачи	CREATE_OR_UPDATE CN=1477916766050.1.1,OU=Центральный офис,OU=XXXX,DC=XXXX,DC=ru <GUID=f3285adaa9bdcf4d9299e59af2712153 >
task.category	категория задачи	Change synchronization
task.parent.oid	oid родительской задачи	a8f00839-14cc-4ed5-af28-e1837efe882e
task.parent.name	имя родительской задачи	Livesync Exchange accounts
task.recurrence	рекуррентность (single / recurring)	single

Таблица 78 – Серверная задача удалена (идентификатор task.deleted)

Атрибут	Описание	Пример

cat	категория события: task_deleted	task_deleted
task.oid	oid удаленной задачи	e02909bf-b97f-473b-a64b-a30e869be48b
task.name	имя удаленной задачи	CREATE_OR_UPDATE CN=1477916766050.1.1,OU=Центральный офис,OU=XXXX,DC=XXXX,DC=ru <GUID=f3285adaa9bdcf4d9299e59af2712153>
task.category	категория удаленной задачи	Change synchronization
task.parent.oid	oid родительской задачи	a8f00839-14cc-4ed5-af28-e1837efe882e
task.parent.name	имя родительской задачи	Livesync Exchange accounts

Таблица 79 – Атрибуты серверной задачи изменены
(идентификатор task.attributes.changed)

Атрибут	Описание	Пример
cat	категория события: task_attributes_changed	task_attributes_changed
task.oid	oid измененной задачи	e02909bf-b97f-473b-a64b-a30e869be48b
task.name	имя измененной задачи	Livesync Exchange accounts
task.category	категория измененной задачи	LiveSynchronization
task.parent.oid	oid родительской задачи	a8f00839-14cc-4ed5-af28-e1837efe882e
task.parent.name	имя родительской задачи	Suped God Task
task.attribute.[attr].old	старое значение атрибута	task.attribute.binding.old=loose

task.attribute.[attr].new	новое значение атрибута	task.attribute.binding.new=tight

Таблица 80 – Серверная задача приостановлена (идентификатор task.suspended)

Атрибут	Описание	Пример
cat	категория события: task_suspended	task_suspended
task.oid	oid приостановленной задачи	e02909bf-b97f-473b-a64b-a30e869be48b
task.name	имя приостановленной задачи	Livesync Exchange accounts
task.category	категория приостановленной задачи	LiveSynchronization
task.parent.oid	oid родительской задачи	a8f00839-14cc-4ed5-af28-e1837efe882e
task.parent.name	имя родительской задачи	Suped God Task

Таблица 81 – Серверная задача возобновлена (идентификатор task.resumed)

Атрибут	Описание	Пример
cat	категория события: task_resumed	task_resumed
task.oid	oid возобновленной задачи	e02909bf-b97f-473b-a64b-a30e869be48b
task.name	имя возобновленной задачи	Livesync Exchange accounts
task.category	категория возобновленной задачи	LiveSynchronization
task.parent.oid	oid родительской задачи	a8f00839-14cc-4ed5-af28-e1837efe882e
task.parent.name	имя родительской задачи	Suped God Task

Таблица 82 – Серверная задача отправлена на немедленное выполнение (идентификатор task.scheduled)

Атрибут	Описание	Пример
cat	категория события: task_scheduled	task_scheduled
task.oid	oid выполняемой задачи	e02909bf-b97f-473b-a64b-a30e869be48b
task.name	имя выполняемой задачи	Livesync Exchange accounts
task.category	категория выполняемой задачи	LiveSynchronization
task.parent.oid	oid родительской задачи	a8f00839-14cc-4ed5-af28-e1837efe882e
task.parent.name	имя родительской задачи	Suped God Task

Таблица 83 – Серверная задача запущена (идентификатор task.started)

Атрибут	Описание	Пример
cat	категория события: task_started	task_started
task.oid	oid запущенной задачи	00000000-0000-0000-0000-000000000005
task.name	имя запущенной задачи	Cleanup
task.recurrence	рекуррентность (single / recurring)	single

Таблица 84 – Серверная задача завершена (идентификатор task.completed)

Атрибут	Описание	Пример

cat	категория события: task_competed	task_competed
task.oid	oid завершенной задачи	00000000-0000-0000-0000-000000000005
task.name	имя завершенной задачи	Cleanup
task.recurrence	рекуррентность (single / recurring)	single

Операции с заявками:

Таблица 85 – Заявка создана (идентификатор request.submitted)

Атрибут	Описание	Пример
cat	категория события: request_submitted	request_submitted
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requester.oid	oid инициатора	00000000-0000-0000-0000-000000000002
requester.name	имя инициатора	administrator
requester.fullName	полное имя инициатора	Пушкин Вениамин Григорьевич
request.type	тип заявки (requestType Uri)	http://inrights/xml/ns/public/request/types#createRoleAssignments
request.args.[arg]	параметры заявки	request.0:args.0:assignment.0:ref.oid=[91642977-d97a-4626-bfd8-030a087b119c] Параметры заявки могут быть произвольными и внутренне представлены как prism-значения. Для отображения параметров в событиях используется алгоритм, анализирующий структуру prism-значений. На этом примере читать аргументы следует так: у заявки есть секция с аргументами (может содержать что угодно)

		в которой на 0-ой позиции содержится секция с именем args в которой на 0-ой позиции содержится секция assignment, в которой на 0-ой позиции содержится объект ref, у которого есть свойство oid со значением [91642977-d97a-4626-bfd8-030a087b119c]

Таблица 86 – Запрос создан (идентификатор requestItem.created)

Атрибут	Описание	Пример
cat	категория события: requestItem_created	requestItem_created
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
requestItem.type	тип запроса (requestItemActionUri)	http://inrights/xml/ns/public/request/actions#createRoleAssignment
requestItem.args.[args]	параметры запроса	requestItem.0:args.0:assignmentRef.oid=[91642977-d97a-4626-bfd8-030a087b119c]requestItem.0:args.0:assignmentObjectRef.oid=[00000000-role-0000-0000-00000000116]requestItem.0:args.0:assignmentSubjectRef.oid=[851616df-719e-41f3-9bc9-006d412c0656]

Таблица 87 – Запущен процесс согласования (идентификатор approvalProcess.started)

Атрибут	Описание	Пример
cat	категория события: approvalProcess_started	approvalProcess_started

request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
approvalSchema.oid	oid маршрута согласования	00000000-0000-0000-0000-000000000034
approvalSchema.name	имя маршрута согласования	Маршрут для заявки на пересмотр полномочий

Таблица 88 – Назначен согласующий / исполнитель
(идентификатор approver.assigned / executor.assigned)

Атрибут	Описание	Пример
cat	категория события: requestParticipant_assigned	requestParticipant_assigned
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
requestItem.processType	тип процесса (approval / execution)	approval
step.id	идентификатор шага	2
step.name	имя шага (workItemName)	Согласование у владельца роли
participant.oid	oid участника	4130f89c-49d9-447c-8ef3-f28851049629
participant.name	имя участника	f.f.preobrazhenskiy
participant.fullName	полное имя участника	Преображенский Филипп Филиппович

assignmentReason	причина назначения	assignmentReason=RoleOwnerAssignmentReasonType assignmentReason.roleRef=372a9fa9-e693-4be3-bb91-89967ee02046

Таблица 89 – Произошла ошибка вычисления согласующего / исполнителя (идентификатор `approver.evaluation.failed`)

Атрибут	Описание	Пример
cat	категория события: <code>requestParticipant_evaluation_failed</code>	<code>requestParticipant_evaluation_failed</code>
request.oid	oid заявки	<code>9b07e87c-657c-4134-a208-2a3179095dc3</code>
request.id	номер заявки	<code>1556</code>
requestItem.id	идентификатор запроса в заявке	<code>0</code>
requestItem.processType	тип процесса (<code>approval / execution</code>)	<code>approval</code>
step.id	идентификатор шага процесса	<code>2</code>
step.name	имя шага процесса (<code>workItemName</code>)	Согласование у владельца роли
errorCode	код ошибки (атрибут <code>errorCode</code>)	<code>error.approver_evaluation.assignee_not_exists</code>

Таблица 90 – Срок шага согласования / исполнения истек (идентификатор `approvalstep.timeout / executionstep.timeout`)

Атрибут	Описание	Пример
cat	категория события: <code>requestStep_timeout</code>	<code>requestStep_timeout</code>
request.oid	oid заявки	<code>9b07e87c-657c-4134-a208-2a3179095dc3</code>

request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
requestItem.processType	тип процесса (approval / execution)	approval
step.id	идентификатор шага процесса	2
step.name	имя шага процесса (workItemName)	Согласование у владельца роли

Таблица 91 – Шаг процесса согласования / исполнения завершен (идентификатор approvalstep.completed / executionstep.completed)

Атрибут	Описание	Пример
cat	категория события: requestStep_completed	requestStep_completed
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
requestItem.processType	тип процесса (approval / execution)	approval
step.id	идентификатор шага процесса	2
step.name	имя шага процесса (workItemName)	Согласование у владельца роли
approvalDecision	принятое решение (только для процессов согласования)	approved

Таблица 92 – Шаг процесса согласования / исполнения пропущен (идентификатор approvalstep.skipped)

Атрибут	Описание	Пример
cat	категория события: requestStep_skipped	requestStep_skipped
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
requestItem.processType	тип процесса (approval / execution)	approval
reason	причина пропуска шага	Для роли Интернет согласование руководителем не требуется

Таблица 93 – Процесс согласования завершен (идентификатор approvalprocess.completed)

Атрибут	Описание	Пример
cat	категория события: approvalProcess_completed	approvalProcess_completed
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
result	результат процесса согласования	APPROVED / REJECTED NONE в случае, если маршрута согласования не было

Таблица 94 – Произошла ошибка исполнения (идентификатор executionstep.failed)

Атрибут	Описание	Пример
cat	категория события: executionStep_failed	executionStep_failed
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0
executionStep.id	идентификатор шага исполнения	0
executionStep.name	имя шага исполнения (workItemName)	На исполнении у руководителя
failure	описание ошибки	

Таблица 95 – Запрос завершен (идентификатор requestItem.completed)

Атрибут	Описание	Пример
cat	категория события: requestItem_completed	requestItem_completed
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3
request.id	номер заявки	1556
requestItem.id	идентификатор запроса в заявке	0

Таблица 96 – Заявка завершена (идентификатор request.completed)

Атрибут	Описание	Пример
cat	категория события: request_completed	request_completed
request.oid	oid заявки	9b07e87c-657c-4134-a208-2a3179095dc3

request.id	номер заявки	1556

Факты доступа к объектам:

Таблица 97 – Отказ в доступе при обращении к объекту (идентификатор model.getobject.failed)

Атрибут	Описание	Пример
cat	категория события: access_denied	access_denied
object.type	тип объекта	UserType
object.oid	oid объекта	9b07e87c-657c-4134-a208-2a3179095dc3
object.name	наименование объекта	ivanov.i.i

Таблица 98 – Успешное обращение к объекту системы (идентификатор model.getobject.success)

Атрибут	Описание	Пример
cat	категория события: access_granted	access_granted
object.type	тип объекта	UserType
object.oid	oid объекта	9b07e87c-657c-4134-a208-2a3179095dc3
object.name	наименование объекта	ivanov.i.i

Интеграция с системами:

Таблица 99 – Операция на коннекторе Create (идентификатор connector.operation.create)

Атрибут	Описание	Пример
cat	категория события: connectorOperation_create	connectorOperation_create

resource.oid	oid ресурса	9b07e87c-657c-4134-a208-2a3179095dc3
resource.name	наименование ресурса	Active Directory
objectClass	objectClass создаваемого объекта	__ACCOUNT__
attribute.[attr]	значения атрибутов создаваемого объекта	attribute.cn=Иванов Иван Иванович attribute.sn=Иванов
result	результат выполнения команды: (статус) success / failed	success
failure	описание ошибки, если result=failed	Connection timed out

Таблица 100 – Операция на коннекторе Delete (идентификатор connector.operation.delete)

Атрибут	Описание	Пример
cat	категория события: connectorOperation_delete	connectorOperation_delete
resource.oid	oid ресурса	9b07e87c-657c-4134-a208-2a3179095dc3
resource.name	наименование ресурса	Active Directory
objectClass	objectClass создаваемого объекта	__ACCOUNT__
uid	uid удаляемого объекта	<GUID=30d533feb01735419ab7758ba7cc99a2>
result	результат выполнения команды: (статус) success / failed	success
failure	описание ошибки, если result=failed	Connection timed out

Таблица 101 – Операция на коннекторе RunScript (идентификатор connector.operation.run.script)

Атрибут	Описание	Пример
cat	категория события: connectorOperation_runScript	connectorOperation_runScript
resource.oid	oid ресурса	9b07e87c-657c-4134-a208-2a3179095dc3
resource.name	наименование ресурса	Active Directory
executionHost	где выполняется скрипт: connector / resource	resource
result	результат (статус) выполнения команды: success / failed	success
failure	описание ошибки, если result=failed	Connection timed out

Таблица 102 – Операция на коннекторе AddAttributeValues (идентификатор connector.operation.add.attribute)

Атрибут	Описание	Пример
cat	категория события: connectorOperation_addAttributeValues	connectorOperation_addAttributeValues
resource.oid	oid ресурса	9b07e87c-657c-4134-a208-2a3179095dc3
resource.name	наименование ресурса	Active Directory
objectClasses	objectClass создаваемого объекта	__ACCOUNT__
uid	uid удаляемого объекта	<GUID=30d533feb01735419ab7758ba7cc99a2>
attribute.[attr]	значения атрибутов создаваемого объекта	attribute.cn=Иванов Иван Иванович attribute.sn=Иванов

result	результат выполнения команды: success / failed	(статус)	success
failure	описание ошибки, если result=failed		Connection timed out

Таблица 103 – Операция на коннекторе RemoveAttributeValues (идентификатор connector.operation.remove.attribute)

Атрибут	Описание	Пример
cat	категория события: connectorOperation_removeAttributeValues	connectorOperation_removeAttributeValues
resource.oid	oid ресурса	9b07e87c-657c-4134-a208-2a3179095dc3
resource.name	наименование ресурса	Active Directory
objectClasses	objectClass создаваемого объекта	__ACCOUNT__
uid	uid удаляемого объекта	<GUID=30d533feb01735419ab7758ba7cc99a2>
attribute.[attr]	значения атрибутов создаваемого объекта	attribute.cn=Иванов Иван Иванович attribute.sn=Иванов
result	результат (статус) выполнения команды: success / failed	success
failure	описание ошибки, если result=failed	Connection timed out

Таблица 104 – Операция на коннекторе Update (идентификатор connector.operation.update)

Атрибут	Описание	Пример
cat	категория события: connectorOperation_update	connectorOperation_update

resource.oid	oid ресурса	9b07e87c-657c-4134-a208-2a3179095dc3
resource.name	наименование ресурса	Active Directory
objectClass	objectClass создаваемого объекта	__ACCOUNT__
uid	uid удаляемого объекта	<GUID=30d533feb01735419ab7758ba7cc99a2>
attribute.[attr]	значения атрибутов создаваемого объекта	attribute.cn=Иванов Иван Иванович attribute.sn=Иванов
result	результат (статус) выполнения команды: success / failed	success
failure	описание ошибки, если result=failed	Connection timed out

Операции с отчетами:

Таблица 105 – Шаблон отчета создан (идентификатор report.created)

Атрибут	Описание	Пример
cat	категория события: reportTemplate_created	reportTemplate_created
template.oid	oid шаблона отчета	aaae7ca6-1b16-4f5e-88d4-f12f7a5c97aa
template.name	наименование шаблона отчета	Анализ ролевой модели

Таблица 106 – Шаблон отчета удален (идентификатор report.deleted)

Атрибут	Описание	Пример
cat	категория события: reportTemplate_deleted	reportTemplate_deleted
template.oid	oid шаблона отчета	aaae7ca6-1b16-4f5e-88d4-f12f7a5c97aa
template.name	наименование шаблона отчета	Анализ ролевой модели

Таблица 107 – Шаблон отчета изменен (идентификатор report.modified)

Атрибут	Описание	Пример
cat	категория события: reportTemplate_modified	reportTemplate_modified
template.oid	oid шаблона отчета	aaaae7ca6-1b16-4f5e-88d4-f12f7a5c97aa
template.name	наименование шаблона отчета	Анализ ролевой модели

Таблица 108 – Отчет по шаблону сформирован (идентификатор report.generated)

Атрибут	Описание	Пример
cat	категория события: report_generated	report_generated
template.oid	oid шаблона отчета	aaaae7ca6-1b16-4f5e-88d4-f12f7a5c97aa
template.name	наименование шаблона отчета	Анализ ролевой модели
param.[name]	параметры отчета, где [name] - имя параметра отчета	param.ORG_LIST=[371af2d3-3631-4f7f-a0fa-162eb149bbf8; 9f5c955c-3ce5-4fb5-8588-d61ed800533a] param.MIN_USER=5 param.MIN_USER_RATE=10
result.md5	md5 полученного файла	9e107d9d372bb6826bd81d3542a419d6
result.format	формат файла отчета	pdf

Таблица 109 – Отчет экспортирован (идентификатор report.exported)

Атрибут	Описание	Пример

cat	категория события: report_exported	report_exported
result.md5	md5 полученного файла	9e107d9d372bb6826bd81d3542a419d6
result.format	формат файла отчета	pdf

Лист регистрации изменений