



Платформа реализации и предоставления  
мультимедийных услуг

## **Platform v3**

---

### **Установка и настройка компонентов**

Версия 3.6

Руководство администратора

## Содержание

О компании Bercut.....	9
Назначение документа.....	11
Термины и определения.....	12
1. Общие сведения.....	15
2. Архитектура Platform v3.....	16
2.1. Интеграционная шина (RTSIB).....	17
2.1.1. Распределенная транзакция (транзакционность).....	19
2.1.2. Сессионность.....	21
2.1.3. Корреляция.....	21
2.1.4. Совместимость версий веб-сервиса.....	22
2.1.5. Управление сессиями.....	23
2.1.6. HTTP Binding.....	24
2.1.7. OpenAPI.....	25
2.2. Выполнение бизнес-процессов (SLES).....	25
2.3. Контейнер LW SA Container.....	26
2.3.1. Нотификации.....	27
2.4. Внешний доступ (Service Gateway).....	28
2.5. Сервисные системы.....	29
2.5.1. Взаимодействие со сторонними системами и дополнительными сервисами.....	29
2.5.2. Система Subscriber Service Profile.....	30
2.5.3. Service Profile Management.....	31
2.5.4. Взаимодействие с BSS.....	31
2.5.5. Взаимодействие с DWH.....	32
2.5.6. Взаимодействие с мобильной сетью оператора.....	33
2.5.7. Планирование и запуск заданий по расписанию (Scheduler).....	33
2.6. Системы представления.....	34
2.6.1. Портал взаимодействия по SMS и USSD.....	34
2.6.2. Портал голосовой и видеосвязи.....	35

2.7. Системы разработки.....	36
2.7.1. Инструмент создания интерактивных меню.....	36
2.7.2. Среда разработки бизнес-процессов.....	37
2.7.3. Общий словарь типов данных.....	38
2.8. Система управления.....	38
2.8.1. Система Operations&Management.....	39
3. Принципы построения бизнес-приложений.....	43
4. Требования.....	44
4.1. Аппаратное обеспечение.....	44
4.2. Программное обеспечение.....	46
4.3. Персонал.....	47
5. Подготовка к установке.....	48
5.1. Установка системы ATLAS Core.....	48
5.1.1. Миграция системы ATLAS Agent 2.0.....	50
5.2. Общие сведения о настройке компонентов.....	51
5.2.1. MIB-переменные.....	51
5.2.2. Настройка логирования трейсовых сообщений.....	52
5.2.3. Настройка интервала обновления статистики.....	52
5.2.4. Загрузка XML-описаний и справочников трейсовых сообщений.....	53
6. Запуск и останов компонентов системы.....	55
7. Настройка Platform v3.....	57
8. Шина RTSIB.....	58
8.1. Варианты подключения компонентов.....	58
8.2. Версионность компонентов в MIB-деревьях.....	61
8.3. Общие настройки.....	62
8.4. Лицензирование шины RTSIB.....	63
8.5. Настройка портов приложений или взаимодействия между компонентами..	63
8.5.1. Настройка взаимодействия по проприетарному протоколу.....	64
8.5.2. Настройка взаимодействия по HTTP-протоколу.....	65
8.5.3. Настройка взаимодействия между компонентами.....	68
8.5.4. Настройка внешних соединений.....	69

8.5.5. Настройка локальных соединений.....	71
8.6. Настройка маршрутов.....	71
8.6.1. Порядок выбора маршрута.....	75
8.7. Настройка условий маршрутизации.....	78
8.8. Маршрутизация по бизнес-параметрам ответа из внешней системы.....	80
8.8.1. Настройка маршрутизации по бизнес-параметрам ответа внешней системы.....	80
8.8.2. Синтаксис адресации к элементам сообщений в XPath-выражениях.....	84
8.9. Обработка широковещательных сообщений.....	85
8.10. Настройка тайм-аутов.....	85
8.10.1. Настройка тайм-аутов внешних вызовов.....	85
8.10.2. Настройка тайм-аутов активности реализации.....	87
8.10.3. Настройка тайм-аутов ожидания ответа.....	88
8.11. Настройка адаптивных очередей.....	89
8.12. Настройка транзакционности.....	90
8.13. Настройка сессионности.....	94
8.14. Совместимость версий WSDL.....	97
8.14.1. Обратная совместимость (Backward compatibility).....	97
8.14.2. Прямая совместимость (Forward compatibility).....	98
8.14.3. Изменение контракта.....	99
8.14.4. Правила расширения WSDL-операций.....	102
8.15. Кеширование ответов на внешние вызовы.....	104
8.16. Настройка уровня логирования по бизнес-параметрам.....	106
8.17. Настройки программы очистки памяти.....	107
8.18. REST-представление с использованием HTTP Binding.....	108
8.18.1. Создание HTTP Binding для существующего SOAP-сервиса.....	109
8.18.2. Создание HTTP Binding из существующего SOAP Binding.....	112
8.18.3. Автоматическое создание HTTP Binding.....	113
8.18.4. Публикация.....	116
8.18.5. Прием и отправка REST-сообщений.....	118
8.18.6. Структуры данных.....	120

8.18.7. Примеры сообщений.....	125
8.19. REST-представление с использованием OpenAPI.....	129
8.19.1. Интеграция RTSIB C++ SDK с OpenAPI Repository.....	129
8.19.2. Настройки OpenAPI Repository.....	130
8.19.3. Описание протокола для взаимодействия с RTSIB.....	131
8.20. Статистические параметры.....	133
9. Система SLES.....	137
9.1. Установка и запуск.....	137
9.2. Развертывание и переустановка BP и SE.....	138
9.2.1. Полная переустановка BP и SE.....	140
9.2.2. Изменение адреса сервиса deployer.....	141
9.2.3. Особенности выполнения команды undeploy_all.....	141
9.3. Настройка SLES.....	141
9.3.1. Настройка ActivationManager.....	142
9.3.2. Настройка BPELEngine.....	146
9.3.3. Настройка Scheduler.....	149
9.3.4. Настройка SeEngine.....	150
9.3.5. Настройка SlesManagement.....	152
9.3.6. Настройка TransactionCoordinator.....	153
9.3.7. Настройка TransportEngine.....	155
9.4. Настройка конфигурационных параметров BP.....	157
9.5. Сервисный режим.....	158
9.6. Логирование.....	159
9.6.1. Параметр Bussines Application для логирования.....	160
9.6.2. Параметр SessionId для логирования.....	160
9.7. Статистика по развернутым бизнес-процессам.....	160
9.8. Настройка Java-опций.....	162
9.9. Синтаксис, используемый при создании условий.....	165
10. Система LW SA Container.....	167
10.1. Установка и запуск.....	167
10.2. Подключение SA.....	168

10.3. Настройка источников данных.....	169
10.3.1. Настройка подключений к схеме данных.....	170
10.3.2. Настройка JMS-подключений СУБД Oracle.....	175
10.3.3. Настройка мониторинга статуса подключения к БД.....	179
10.3.4. Соответствие параметров Tomcat DataSource и MIB.....	181
10.4. Настройка конфигурационных параметров SA.....	184
10.5. Менеджер транзакций.....	184
10.6. Настройка UserPrincipal и SystemPrincipal.....	187
10.7. Логирование.....	188
10.7.1. Параметр Bussines Application для логирования.....	188
10.7.2. Параметр SessionId для логирования.....	188
10.7.3. Запись логов SA.....	188
10.8. Статистические параметры JMS-соединения.....	190
10.9. Статистические параметры обработки транзакций.....	192
11. Сборщик мусора Garbage Collector G1.....	196
12. Система Service Gateway.....	199
12.1. Установка и запуск.....	199
12.2. Настройка Service Gateway.....	200
12.2.1. Настройка параметров аутентификации и авторизации в MIB.....	204
12.2.2. Настройка параметров аутентификации и авторизации в профиле Service Gateway.....	212
12.2.3. Настройка ограничений трафика во внутренней сети.....	218
12.2.4. Настройка автоматического обновления интерфейсов.....	219
12.2.5. Настройка трассировки сообщений.....	220
12.2.6. Управляющие системные SOAP-заголовки.....	221
12.2.7. Настройка тайм-аутов.....	221
12.2.8. Статистические параметры.....	222
13. Система Subscriber Service Profile.....	224
13.1. Установка Subscriber Service Profile Subsystem.....	224
13.2. Настройка Subscriber Service Profile Subsystem.....	224
14. Система Session Management.....	227

14.1. Установка системы Session Management.....	227
14.2. Настройка системы Session Management.....	227
15. Система Operations&Management.....	229
15.1. Установка и настройка системы ATLAS Agent.....	231
15.1.1. Настройка авторизации в Kafka.....	232
15.2. Установка и настройка компонента logDecoder.....	232
15.3. Установка и настройка компонента traceDecoder.....	233
15.3.1. Постобработка цепочки трейсов.....	236
15.4. Установка и настройка компонента logstash.....	237
15.5. Установка и настройка компонента Zookeeper.....	238
15.5.1. Подписка на добавление, изменение и удаление portType.....	239
15.5.2. Конфигурация маршрутов на шине RTSIB.....	240
15.5.3. Соединения.....	240
15.5.4. Маршруты.....	240
15.5.5. Пример YAML-описания.....	241
16. Платформа Tarantool.....	244
16.1. Tarantool RTSIB Driver.....	244
16.1.1. Установка Tarantool RTSIB Driver.....	246
16.1.2. Интерфейс для Lua.....	246
16.1.3. Примеры использования Tarantool RTSIB Driver.....	250
16.2. Tarantool SOAP Adapter.....	253
16.2.1. SOAP-интерфейсы.....	253
16.2.2. Примеры использования Tarantool SOAP Adapter.....	255
17. Специализированный адаптер SA LDAP.....	259
18. Специализированный адаптер SA SAL SMPP.....	260
18.1. SMPP-соединение.....	261
18.2. Режимы взаимодействия с SMSC.....	262
18.3. Отправка SMS-сообщений в SMSC.....	263
18.4. Прием SMS-сообщений от SMSC.....	264
18.5. Получение отчета о доставке.....	264
18.6. Получение статуса доставки сообщения.....	266

18.7. Отмена доставки сообщения.....	266
18.8. Обработка длинных сообщений.....	267
18.9. Режим немедленного ответа.....	269
18.10. Поддержка диалогов в рамках одной сессии.....	270
18.11. Кодировка SMPP-сообщений.....	271
18.12. Указание SMPP-параметров в SOAP-запросе.....	271
19. Специализированный адаптер SA Multilink.....	272
20. Установка и настройка компонентов платформы Unica для CareM v3.....	273
20.1. Установка и запуск Unica SLR.....	273
20.2. Развертывание Unica-профилей.....	274
20.3. Настройка взаимодействия Service Gateway и Unica SLR.....	274
21. Расширение системы.....	276
22. Резервирование и масштабирование.....	277
22.1. Балансировка нагрузки и резервирование шины RTSIB.....	277
23. Мониторинг системы.....	279
23.1. Сообщения о работе системы.....	279
23.2. Статистика работы систем и компонентов с помощью ATLAS.....	280
Приложение 1. Использование Service Profile Management. Пример работы с сервисом.....	281
Приложение 2. Недопустимые символы в строковых данных.....	287
История изменений системы.....	288
Источники информации.....	289
История изменений.....	290



## О компании Bercut

*Bercut — мировой поставщик решений в области ИТ, который предлагает уникальный подход к развитию и управлению услугами совместно с оператором и абонентом.*

### Техническая поддержка

*Компания Bercut предлагает заказчикам полную техническую поддержку продуктов.*

Bercut осуществляет гарантийное и послегарантийное сопровождение поставляемых комплексов по отдельному договору.

При возникновении в процессе эксплуатации ситуаций, не указанных в пакете эксплуатационной документации, пользователь может обратиться в группу технической поддержки компании Bercut одним из указанных ниже способов:

- на сайте <https://support.bercut.com> создать заявку (раздел **Заявки**);
- отправить электронное письмо на адрес [support@bercut.com](mailto:support@bercut.com);
- позвонить по телефону +7 (812) 327-3231.

### Уведомление об авторских правах

Компания Bercut обладает исключительным правом на данные материалы.

Не допускается полностью или частично воспроизводить или передавать данный документ в какой-либо форме, любым способом и в любом формате, электронными или механическими средствами, включая фотокопирование, запись и хранение в системе базы данных, не получив предварительное согласие в письменном виде от компании Bercut.

## Обратная связь

Уважаемый читатель!

Наша цель — улучшение документации с точки зрения удобства ее использования, полноты и понятности изложенного материала. Свои вопросы, предложения, замечания об ошибках, неясности в изложении, нехватке примеров вы можете передать одним из указанных ниже способов:

- на сайте <https://support.bercut.com> создать заявку (раздел **Заявки**);
- отправить электронное письмо на адрес [techwriters@bercut.com](mailto:techwriters@bercut.com).

Пожалуйста, укажите:

- версию системы;
- название документа;
- номер версии документа;
- по возможности — главу, раздел и страницу, к которым относятся ваши замечания.

После исправления текста по замечаниям мы известим вас о выходе новой версии документа.

**i Примечание.** В соответствии с положениями политики конфиденциальности мы принимаем обратную связь от компаний, с которыми установлены соответствующие договорные обязательства. Если вы являетесь третьей стороной, пожалуйста, обратитесь к представителям компании, с которой у вас заключен договор.

# Назначение документа

В документе описаны:

- архитектура и компонентный состав Platform v3;
- установка компонентов платформы;
- настройка.

Документ предназначен для пользователей, которые выполняют установку и настройку компонентов Platform v3.

## Термины и определения

### **АОИ**

Abstract Object Interface. Абстрактный интерфейс сервисов и бизнес-процессов Platform v3 на основе WSDL-описания.

### **API**

Application Programming Interface. Стандартизованный программный интерфейс, который применяется для доступа внешних приложений к функциям и данным системы.

### **ATLAS**

Administration Tools Layer for Applications and Services. Система администрирования и мониторинга приложений и бизнес-процессов. Расширенная версия системы ATOMS, предназначенная для управления компонентами и бизнес-процессами Платформы, для наблюдения за ними в режиме реального времени и оповещения об авариях и сбоях, возникающих в работе приложений.

### **BP**

Business Process. Сценарий на языке BPEL, выполняемый на Platform v3 для реализации бизнес-задачи. Реализует одну или несколько бизнес-операций.

### **BPEL**

Business Process Execution Language. Язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия. Задача BPEL — координация совместной работы веб-служб.

### **BPMN**

Business Process Model Notation. Система условных обозначений для моделирования бизнес-процессов.

### **DWH**

Data Warehouse. Информационная база данных, которая получает данные из используемых компанией баз данных и других источников данных, а затем трансформирует их в структуру, подходящую для выполнения бизнес-анализа.

### **Garbage Collector**

Сокращенно — GC. Программа очистки памяти.

### **LW SA Container**

Light Weight Specialized Adapter Container. Контейнер на базе Tomcat, который используется для подключения SA к RTSIB.

### **MIB**

Management Information Base. База управляющей информации. MIB содержит настройки для приложений и бизнес-процессов Bercut, которые выполняются на сервере. Для доступа к MIB используется внутренний протокол.

### **MIB Explorer**

Сокращенно — ME. Приложение, которое предоставляет пользователю интерфейс для удаленного администрирования и управления системами компании Bercut. Входит в состав системы [ATLAS](#).

**OpenAPI**

Спецификация для описания *REST API*, которая не зависит от языка реализации.

**Platform v3**

Платформа компании Bercut для разработки приложений (VAS-услуг), реализуемых как набор бизнес-процессов.

**portType**

Элемент *WSDL*, абстрактный интерфейс веб-сервиса. Содержит список возможных операций с входящими и исходящими сообщениями.

**REST**

Representational State Transfer. Репрезентативная передача состояния — архитектурный принцип организации взаимодействия систем с обменом информацией об их текущем состоянии.

**RTSIB**

Real-Time Service Integration Bus. Интеграционная шина для обмена данными между *бизнес-процессами*, *SA*, *SE* и компонентами Platform v3 в онлайн-режиме.

**SA**

Specialized Adapter. Специализированный адаптер, развернутый на *LW SA Container*. Обеспечивает взаимодействие внешней системы с уровнем логики *Platform v3* с помощью шины *RTSIB* по веб-интерфейсу.

**Scheduler Subsystem**

Система на Platform v3 для создания расписаний, запуска задач и бизнес-процессов.

**SCS**

Service Creation Studio. Система разработки бизнес-процессов на языке *BPEL* и интерфейсов на языке *WSDL*. С помощью SCS можно создавать, настраивать и тестировать Service Adapter и Service Enabler на языке Java.

**SE**

Service Enabler. Логическая группа вспомогательных операций на языке Java. Бизнес-процесс использует SE для решения технологических задач.

**Service Gateway**

Сокращенно — *SG*. Шлюз доступа к сервисам Platform v3 для внешних приложений.

**Service Profile Management**

Система хранения и управления профилями сервисов. В разных источниках используются названия *Service Profile* или *Service Management*. Сокращенно — *SP* или *SPM*.

**SLES**

Service Logic Execution System. Система выполнения бизнес-процессов на Platform v3.

**Subscriber Service Profile Management**

Сокращенно — *SSP*. Хранилище профилей абонентов сети мобильной связи на Platform v3. В разных источниках используются названия *Subscriber Service Profile* или *Subscriber Service Profile Management*.

## WSDL

Web Services Description Language. Язык описания веб-служб, основанный на языке XML.

## Аудит

Мониторинг действий пользователя в системе *ATLAS*: подключение и отключение пользователя от компонента ATLAS Agent, неудачные попытки подключения, операции с MIB-группами и MIB-переменными — создание, удаление, переименование, изменение значения. Сообщения аудита записываются в отдельный журнал событий.

## Логирование

Запись в журнал событий системы *ATLAS* следующей информации:

- *Трейс* — о работе сервера и приложений на нем;
- *Аудит* — об изменении конфигурации приложений пользователями системы.

Просмотр журнала событий (трейса или аудита) выполняется следующими способами:

- в онлайн-режиме в приложении *ATLAS MIB Explorer*;
- с помощью выгрузки из ATLAS Agent в бинарные файлы .blf;
- в офлайн-режиме в приложении ATLAS Log Viewer.

## Операция

Логическое действие, которое выполняется в *BP* (бизнес-операция) или *SE* (технологическая операция).

## Профиль сервиса

Набор параметров, который хранится в базе данных и используется внешними системами для взаимодействия с данным сервисом.

## Сервис

Набор операций, выполняемых с определенной целью.

## Трассировка

Запись в трейс информации о пакетах данных и сообщениях, которые приложения получают и отправляют по транспортным соединениям. Трейс содержит следующую информацию:

- транспортные свойства пакета: адреса, номера портов, заголовки;
- бизнес-свойства пакета: имя и версия источника, бизнес-приложение, идентификатор сессии, метainформация для связывания цепочки вызовов;
- сообщение, которое передается между приложениями.

Пользователь может просматривать трейсы в системах Kibana или Grafana.

## Трейс

Сообщения, порождаемые приложениями Bercut, о событиях в процессе их работы. Представляют собой текстовую древовидную структуру с параметрами разных типов. В простейшем случае трейс — это текстовая строка.

# 1. Общие сведения

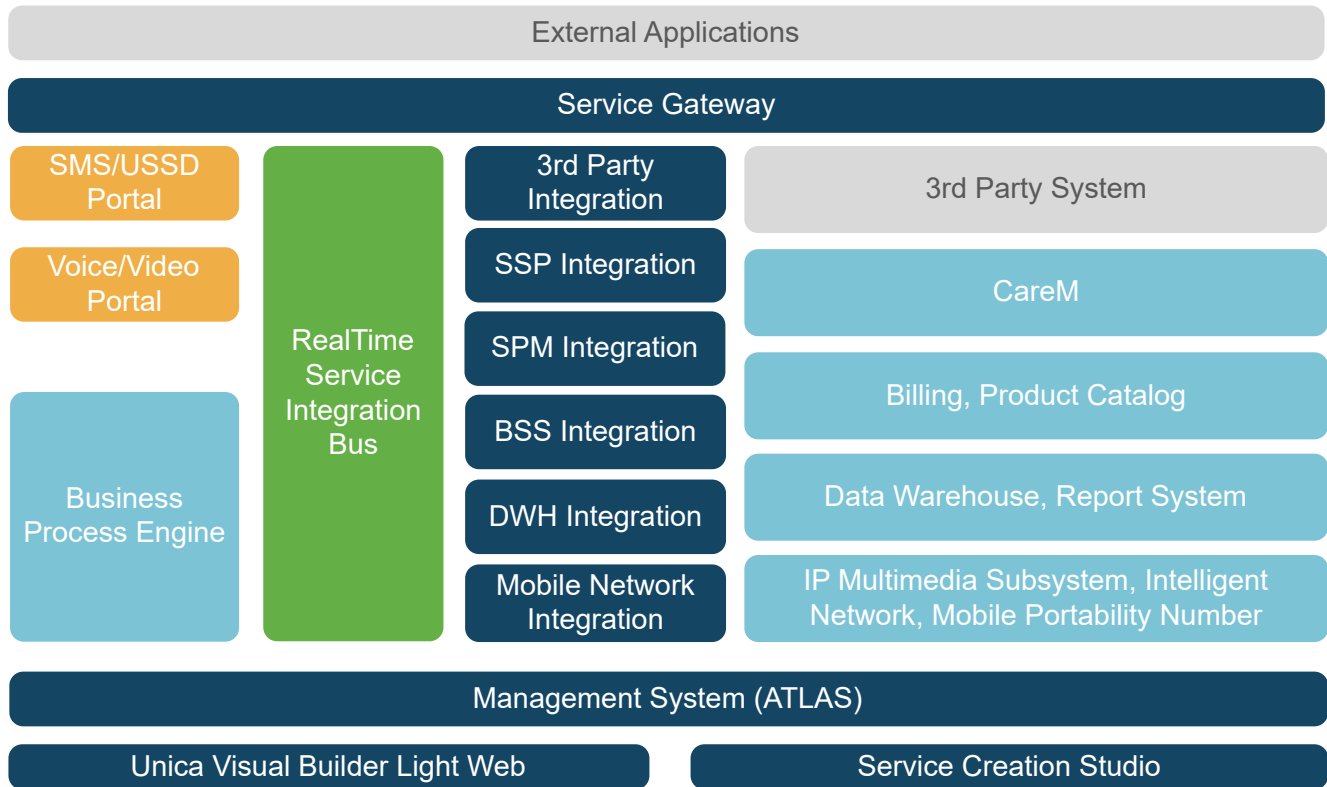
*Platform v3 — мультимедийная платформа для интеграции, реализации и предоставления услуг абонентам в сетях фиксированной и подвижной телефонной связи стандартов GSM 900/1800, CDMA, UMTS.*

Platform v3 выполняет следующие функции:

- Создание отдельных бизнес-ориентированных процессов и законченных бизнес-приложений.
- Быстрая разработка бизнес-ориентированных интерфейсов и специализированных адаптеров к внешним и внутренним системам оператора.
- Доступ внешних приложений (Web, SOA applications) к бизнес-ориентированным процессам и сервисам платформы.
- Выполнение функций IP Multimedia Subsystem (IMS) и Intelligent Network (IN).
- Создание SMS-, USSD-, IVVR-порталов.
- Создание и выполнение сценариев обработки событий (BRMS).
- Поддержка событийной EDA-модели.
- Мониторинг и поиск проблем (troubleshooting) распределенных приложений.
- Обеспечение атомарности бизнес-действий в бизнес-процессах при поддержке распределенных транзакций в базах данных.
- Организация горизонтального масштабирования и резервирования компонентов распределенных приложений.
- Разработка REST-сервисов и выполнение REST-вызовов из сервисов.

## 2. Архитектура Platform v3

Platform v3 состоит из внутренних и внешних компонентов, объединенных с помощью интеграционной шины RTSIB. Управление компонентами выполняет система ATLAS.



**Рис. 1. Архитектура Platform v3**

Platform v3 включает в себя следующие компоненты:

- **External Applications.** Внешние SOA- и веб-приложения, которые используют сервисы и бизнес-процессы Platform v3.
- **Service Gateway.** Шлюз доступа к сервисам Platform v3 для внешних приложений.
- **Системы представления Platform v3:**
  - **SMS/USSD Portal.** Портал представления SMS- и USSD-витрин.
  - **Voice/Video Portal.** Портал представления IVVR-витрин.
- **Business Process Engine.** Система выполнения бизнес-процессов Platform v3.
- **Real Time Service Integration Bus.** Интеграционная шина для обмена данными между бизнес-процессами, SA, SE и компонентами Platform v3 в режиме реального времени.
- **Сервисные системы Platform v3:**
  - **3d Party Integration.** Компоненты для быстрого включения функциональных возможностей внешних сервисов в Platform v3.
  - **SSP Integration.** Система интеграции различных сервисов взаимодействия с пользователем в Platform v3. Решения Bercut:
    - CareM. Система самообслуживания абонентов.
  - **SPM Integration.** Система хранения и управления профилями сервисов



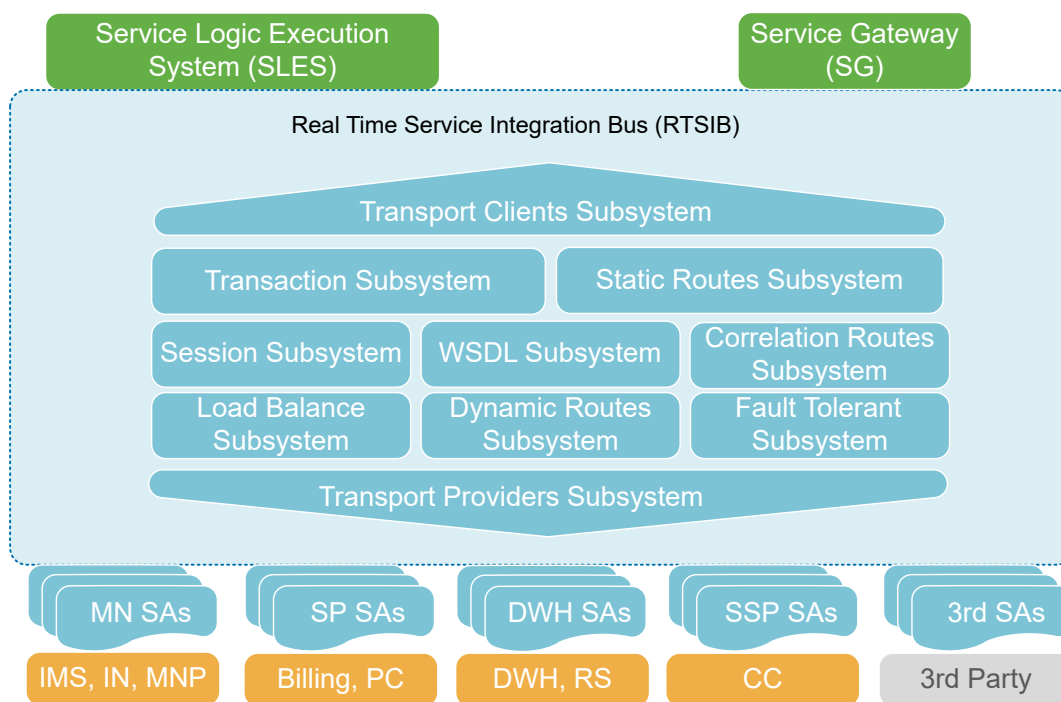
- **BSS Integration.** Система интеграции сервисов для выполнения биллинговых и других бизнес-операций в Platform v3. Решения Bercut:
  - Billing System (IN@Voice). Биллинговая система компании Bercut.
  - Product Catalog. Инструмент для создания продуктов и управления ими.
  - SPM (Service Profile Subsystem). Система хранения и управления профилями сервисов.
- **DWH Integration.** Система интеграции DWH в Platform v3. Решения Bercut:
  - Data Warehouse (Audit). Подсистема хранения данных о бизнес-операциях Platform v3. Данные используются при подготовке информации для отчетов.
  - Report System (Jasper). Подсистема формирования и управления отчетами.
- **Network Integration** . Система интеграции сервисов мобильных операторов для PS- и CS-доменов на базе Platform v3. Решения Bercut:
  - Intelligent Network (STP, SCP, SSP). Сетевая архитектура для CS доменов операторов связи.
  - IP Multimedia Subsystem (S-CSCF, MRFC, MGW). Сетевая архитектура для CS- и PS-доменов операторов связи.
- **Management System (ATLAS).** Система управления и мониторинга всех компонентов и систем Platform v3.
- **Системы разработки Platform v3:**
  - **Unica Visual Builder Light Web.** Средство визуального проектирования SMS-, USSD- и IVVR-витрин (меню) для абонентов оператора связи.
  - **Service Creation Studio.** Средство визуального проектирования бизнес-процессов.

## 2.1. Интеграционная шина (RTSIB)

*RTSIB логически объединяет все интерфейсы к сервисам и бизнес-процессам Platform v3, обеспечивая высокую скорость обмена данными.*

Основные функции RTSIB:

- Поддержка распределенных транзакций при последовательном вызове сервисов и (или) бизнес-процессов, которые поддерживают транзакционный режим работы.
- Поддержка сессионного режима работы, позволяет обеспечить последовательность вызовов к сессионным сервисам (SA) или бизнес-процессам (BP).
- Маршрутизация вызовов к конкретному экземпляру сервиса или бизнес-процесса — выбор маршрута в зависимости от параметров вызова.
- Динамическое определение маршрута к конкретному экземпляру сервиса или бизнес-процесса — вызов дополнительного сервиса или бизнес-процесса для выбора необходимого направления вызова.
- Общий логический реестр всех интерфейсов сервисов и бизнес-процессов, опубликованных в Platform v3.
- Подсистема балансировки и резервирования конкретных экземпляров сервисов и бизнес-процессов.
- Транспортные системы предоставления и использования сервисов и бизнес-процессов Platform v3.



**Рис. 2. Архитектура RTSIB**

## **Transport Clients Subsystem и Transport Providers Subsystem**

Позволяют предоставлять и использовать сервисы и бизнес-процессы Platform v3.

## **Transaction Subsystem**

Реализует распределенные транзакции при последовательном вызове сервисов и (или) бизнес-процессов, которые поддерживают транзакционный режим работы.

## **Static Routes Subsystem**

Маршрутизирует вызовы к конкретному экземпляру сервиса или бизнес-процесса — выбор маршрута в зависимости от параметров вызова.

## **Session Subsystem**

Поддерживает сессионный режим работы для обеспечения последовательности вызовов к сессионным сервисам или бизнес-процессам.

## **WSDL Subsystem**

Предоставляет WSDL-интерфейс.

## **Correlation Routes Subsystem**

Определяет маршрут отправленного сообщения.

## **Load Balance Subsystem**

Выполняют балансировку конкретных экземпляров сервисов и бизнес-процессов.

## **Dynamic Routes Subsystem**

Динамически определяет маршрут к конкретному экземпляру сервиса или бизнес-процесса — вызывает дополнительный сервис или бизнес-процесс для выбора необходимого направления вызова.

## **Fault Tolerant Subsystem**

Резервирует конкретные экземпляры сервисов и бизнес-процессов.

Установка и настройка — в разделе 8.

### 2.1.1. Распределенная транзакция (транзакционность)

*Platform v3 работает с распределенными транзакциями: данные распределены по нескольким источникам. Механизм распределенной транзакции основан на спецификациях WS-AtomicTransaction и WS-Coordination.*

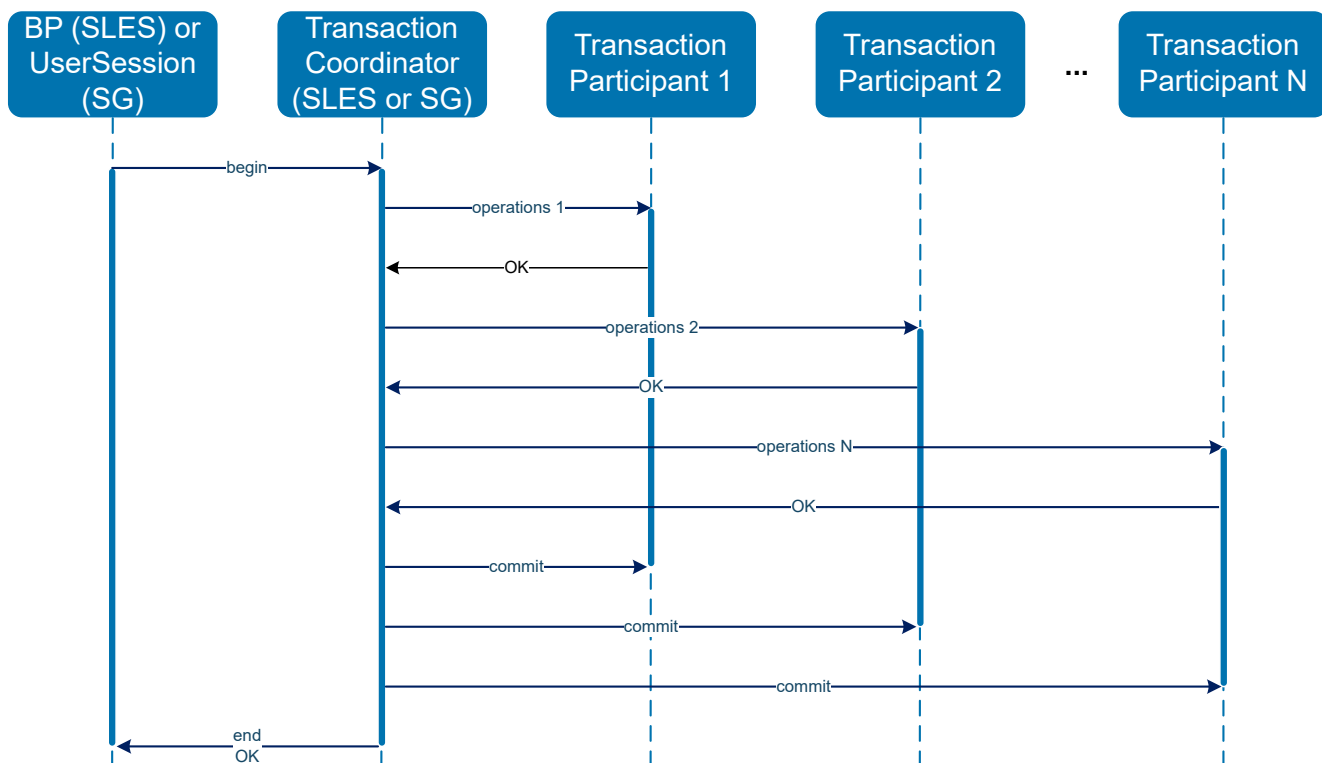
При работе с распределенной транзакцией используются следующие сущности:

- Transaction Manager (TM) — управление распределенной транзакцией. Содержит два компонента:
  - Transaction Participant (TP) — выполнение транзакции в источнике.
  - Transaction Coordinator (TC) — координация управляющих сообщений и синхронизация работы всех TP.
- Coordination Context (CC) — контекст транзакции в операциях.
- transaction Partner Link Type (PLTt) — транзакционная связь в интерфейсах.

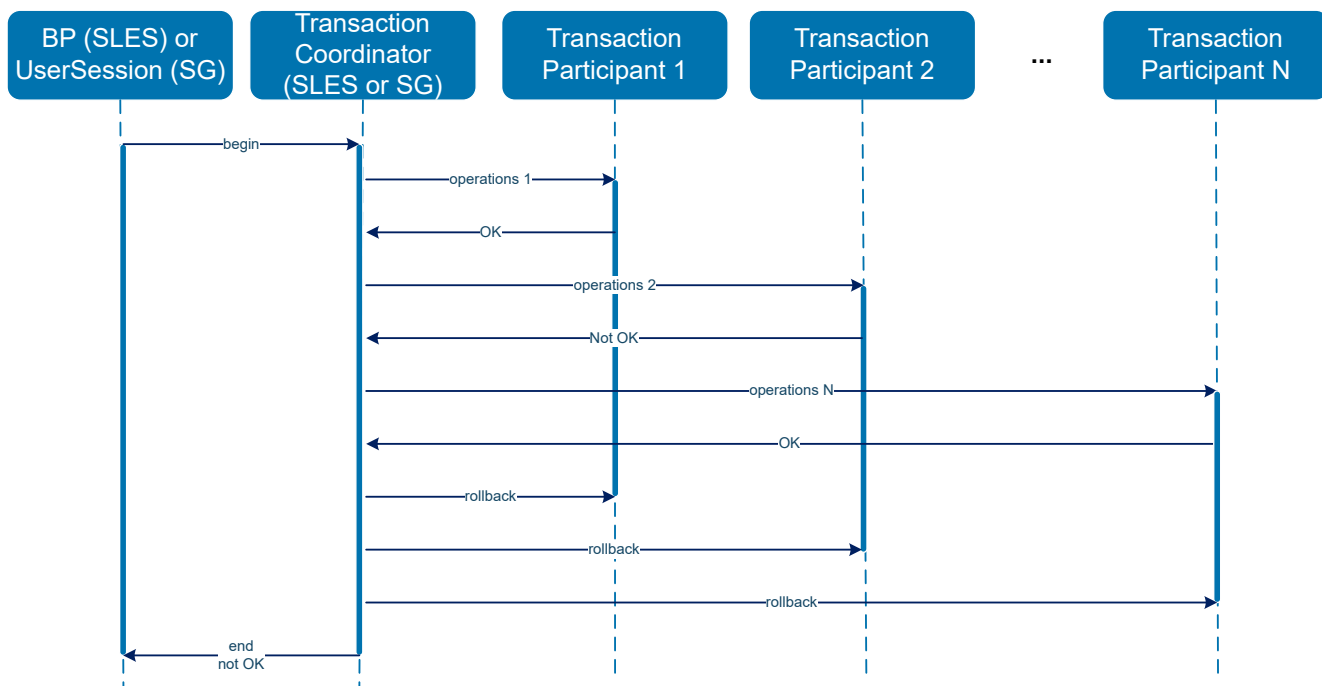
Transaction Coordinator работает на SLES. Алгоритм работы механизма кратко описан ниже.

1. Transaction Coordinator ожидает ответов от всех источников, чтобы завершить распределенную транзакцию.
2. После получения сообщений об успешном выполнении всех операций от всех источников Transaction Coordinator регистрирует данные транзакции.

**i Примечание.** Транзакция завершается с ошибкой, если хотя бы от одного из источников не поступило сообщение об успешном выполнении операций или поступило сообщение об ошибке. В этом случае Transaction Coordinator отменяет все изменения данных внутри транзакции до начального состояния.



**Рис. 3. Успешное завершение распределенной транзакции**



**Рис. 4. Неуспешное завершение распределенной транзакции**

В RTSIB заголовок вызова содержит следующие параметры:

- TransactionIdentifier — уникальный идентификатор транзакции (tid).
- Identifier — идентификатор контекста, соответствует tid.
- Expires — время жизни транзакции, в секундах. Необязательный параметр.
- CoordinationType — тип поддерживаемого координатора.

- `RegistrationService` — адрес сервиса регистрации.

Считается, что бизнес-процесс, все элементы которого являются транзакционными, выполняется полностью в транзакционном режиме.

Настройка транзакционности — в разделе [8.12](#).

### 2.1.2. Сессионность

*RTSIB поддерживает сессионный режим работы.*

Сессия используется для маршрутизации последовательности сообщений между двумя удаленными точками или объектами. У каждой сессии есть уникальный сессионный идентификатор. Сессионным идентификатором является пара «уникальный идентификатор локальной точки или объекта — уникальный идентификатор удаленной точки или объекта».

Типы сессий:

- на `Service Gateway`. Запросы в рамках этой сессии поступают от внешних систем.
- на `SLES`. В рамках этой сессии выполняется взаимодействие со `SLES`.
- Пользовательская. В рамках сессии выполняются пользовательские запросы.

В WSDL-описании хранятся параметры сессии, информирующие о ее состоянии: создании, продолжении или окончании. Операции начала, продолжения или окончания сессии вызываются для сервера-владельца сессии.

Настройка сессионности — в разделе [8.13](#).

### 2.1.3. Корреляция

*Система `Correlation System` выполняет динамическую маршрутизацию отправленного сообщения с использованием принципа корреляции.*

Корреляция — это динамическая маршрутизация сообщения в зависимости от ключа корреляции. Маршрутизация называется динамической, потому что маршрут до адресата определяется в процессе отправки сообщения.

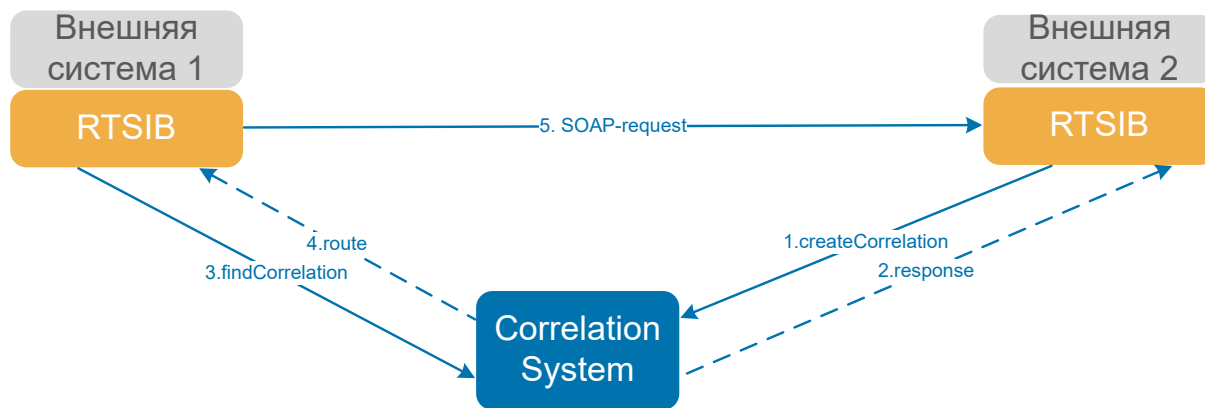
Ключ корреляции — это набор атрибутов (пар имя-значение). При этом имя фиксировано, а значение вычисляется по SOAP-сообщению, например:

```
msisdn=9236547897  
sessionType=GPRS
```

При использовании корреляции внешняя система по ключу корреляции получает информацию о конечной точке маршрута этого сообщения. Для этого внешняя система отправляет запрос по специализированному интерфейсу `Correlation Manager` в систему корреляции.

В качестве внешних систем по отношению к системе корреляции выступают компоненты, использующие `RTSIB` и выполняющие маршрутизацию по корреляции.

Внешние системы не требуют специальных доработок для использования системы корреляции, так как ключ корреляции вычисляется по SOAP-запросу.



**Рис. 5. Принцип работы системы Correlation System**

1. Внешняя система 2 регистрирует ключ корреляции в системе Correlation System по бизнес-параметрам.
2. Система Correlation System передает результат создания корреляции.
3. Пользователь настраивает соединение с системой Correlation System в библиотеке RTSIB внешней системы 1 для маршрутизации по корреляции.

При отправке SOAP-сообщения библиотека RTSIB внешней системы 1 определяет условие отправки по бизнес-параметрам. Бизнес-параметры формируют ключ корреляции по исходящему SOAP-запросу.

Если условие соблюдено, библиотека RTSIB отправляет служебный запрос findCorrelation в Correlation System.

4. Correlation System ищет корреляцию. Если корреляция найдена — возвращает маршрут в библиотеку RTSIB внешней системы 1. Корреляция в системе Correlation System заранее создана внешней системой 2 (шаг 1).
5. Библиотека RTSIB внешней системы 1 отправляет SOAP-сообщение по переданному маршруту во внешнюю систему 2.

Установка и настройка Correlation System — [16].

#### 2.1.4. Совместимость версий веб-сервиса

Совместимость версий веб-сервиса зависит от его контракта и может быть прямой или обратной.

Контракт веб-сервиса может содержать:

- одно или несколько WSDL-описаний;
- одну или несколько XML- или XSD-схему;
- WS-Policy или WS-описания.

При создании или изменении контракта используются первые два пункта.

Типы совместимости веб-сервисов:

- *backward compatibility* — обратная;
- *forward compatibility* — прямая.

### 2.1.5. Управление сессиями

Для управления сессиями Platform v3 — хранения и обработки — предназначена система Session Management.

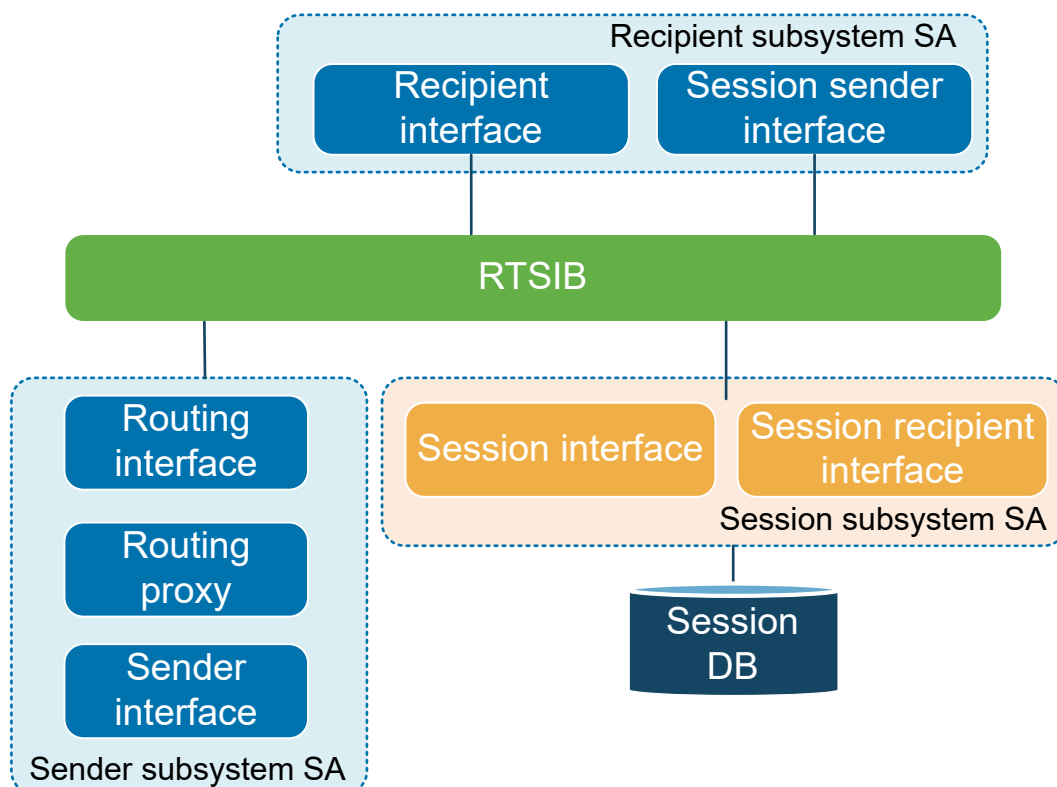
Platform v3 использует следующие типы сессий при работе с протоколом авторизации OAuth:

- User session. Сессия обеспечивает прозрачную для пользователя аутентификацию.
- Token session. Сессия обеспечивает авторизацию внешних приложений на Service Gateway для ограниченного доступа к услугам Platform v3.

Система использует интерфейс AOI\_SessionManagement, включающий следующие методы:

- Создание сессии — createSession.
- Изменение или обновление сессии — updateSession.
- Поиск сессии в БД по условию — getSession.
- Получение данных сессии — findSessions.
- Удаление сессии — deleteSession.

При маршрутизации сообщений система использует сессию маршрутизации для нотификаций сообщений. Сессия устанавливает соответствие пользовательского идентификатора и уникального идентификатора удаленной точки или объекта.



**Рис. 6. Маршрутизация сообщений**

Система Session Management включает в себя следующие компоненты:

- Sender subsystem SA. Формирование и отправка сообщений на Recipient subsystem SA. Состоит из блоков:

- Sender interface — интерфейс, формирующий исходящее сообщение для Recipient subsystem SA. Содержит ключевой параметр, например, идентификатор абонента. Не содержит сессионный ключ в поле **To**. Система отправляет сформированное сообщение на один из экземпляров Recipient subsystem SA в зависимости от правил маршрутизации RTSIB. В случае включения *Routing component* интерфейс перехватывает и обрабатывает сообщение.
- Routing interface и Routing proxy обрабатывают сформированное Sender interface сообщение, выполняя следующие шаги:
  - Чтение ключевого параметра. Правило поиска параметра является XPath-выражением.
  - Формирование запроса с ключевым параметром на Session subsystem SA.
  - Получение сообщения от Session subsystem SA, содержащего уникальный идентификатор локальной точки или объекта.
  - Подстановка уникального идентификатора локальной точки или объекта в поле **To**.
- Routing interface и Routing proxy отправляют сформированное сообщение.
- Recipient subsystem SA. Взаимодействует с Sender subsystem SA и с Session subsystem SA. Состоит из блоков:
  - Recipient interface. Интерфейс, принимающий сообщения от Sender subsystem SA.
  - Session sender interface. Интерфейс, формирующий исходящее сообщения для Session subsystem SA. Сообщение содержит данные об открытых и закрытых сессиях.
- Session subsystem SA.
  - Session interface. Взаимодействует с Session DB. Функции интерфейса:
    - Прием запросов с ключевым параметром.
    - Поиск сессии в Session DB.
    - Возврат ответного сообщения. Сообщение может содержать: уникальный идентификатор локальной точки (объекта) или сообщение о неудачной попытке поиска сессии в Session DB.
  - Session recipient interface. Получение сообщений от Recipient subsystem SA с данными об открытых и закрытых сессиях. Система добавляет данные в Session DB.
- Session DB — база данных текущих сессий Recipient subsystem SA.

Установка и настройка системы Session Management — в разделе [14](#).

## 2.1.6. HTTP Binding

*HTTP Binding* — это описание реализации операций *portType* с использованием HTTP-протокола.

HTTP Binding описывает:

- имена ресурсов (URL);
- типы кодирования запросов и ответов;
- маппинг HTTP-кодов для сообщений *fault*.



Настройка HTTP Binding — [8.18](#).

### 2.1.7. OpenAPI

*OpenAPI — спецификация для описания REST API, не зависящая от языка реализации.*

Для описания REST API в RTSIB созданы две реализации спецификации OpenAPI.

#### **Реализация спецификации OpenAPI на основе языка программирования Java**

1. Поддержка REST-представления для интерфейса, описанного в WSDL, одним из следующих способов:
  - 1.1. На основе описания REST-представления, имеющегося в WSDL.
  - 1.2. С помощью автоматического генерирования:
    - 1.2.1. *REST-представления при развертывании BP или SE.*
    - 1.2.2. *REST-представления при разработке SA при разработке SA.*
    - 1.2.3. REST-представления при публикации сервиса:
      - 1.2.3.1. В системе API Gateway [[13](#)].
      - 1.2.3.2. *В шлюзе Service Gateway.*
2. *Представление в виде REST-сервера* любого серверного интерфейса, у которого реализовано REST-представление.
3. *Отправка REST-запросов* для всех интерфейсов, у которого реализовано REST-представление.
4. *Публикация OpenAPI-описаний* для всех интерфейсов с REST-представлениями.

#### **Реализация спецификации OpenAPI на основе языка программирования C++**

1. Использование репозитория для хранения OpenAPI-описаний в виде файлов в формате YAML.
2. Предоставление по запросу файлов в формате YAML из репозитория.

Подробнее — [8.19](#).

## 2.2. Выполнение бизнес-процессов (SLES)

*Система SLES выполняет бизнес-процессы (BPEL-сценарии) и вспомогательные сервисы (Java-сценарии).*

Бизнес-процессы и вспомогательные сервисы могут вызывать другие бизнес-процессы и сервисы Platform v3 (далее — *Платформы*).

Основные функции системы SLES:

- Выполнение бизнес-процессов и вспомогательных сервисов.
- Поиск и запуск необходимого бизнес-процесса, в зависимости от параметров поступившего внешнего события.
- Использование сессий, транзакций и корреляционных связей, взятых из бизнес-процессов Платформы.
- Сохранение текущего состояния и данных бизнес-процесса, который требует длительного ожидания внешнего события для продолжения выполнения — «Long Running BP Subsystem».

Логику составляет набор бизнес-процессов и сервисные драйверы, используемые ими, которые размещаются на сервере. Связь с внешним и локальным окружением обеспечивают интерфейсы абстрактных объектов — AOI.

**Примечание.** Развертывание новых BP, SE, AOI в SLES не приводит к остановке работающих приложений.

SLES реализует подключение к RTSIB для взаимодействия сервисных логик с объектами внешнего окружения — локально, с помощью интерфейса шины.

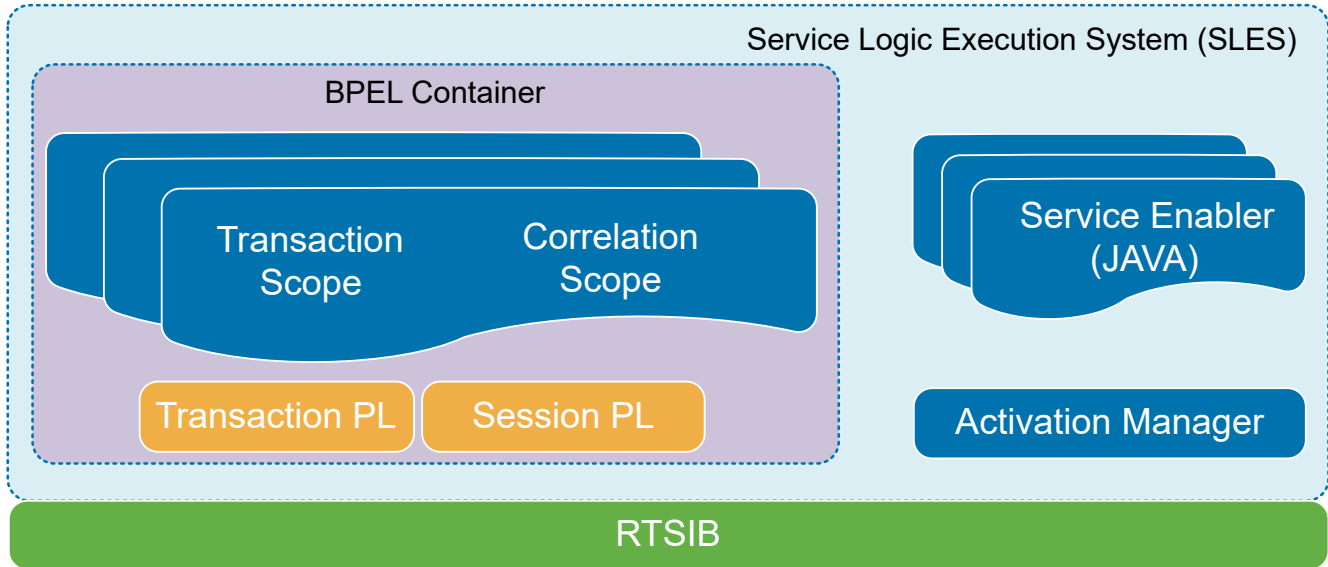


Рис. 7. Архитектура SLES

### Service Enabler

Абстрактный интерфейс, реализованный на Java и выполняемый в SLES.

### Activation Manager

Поиск и запуск необходимого бизнес-процесса, в зависимости от входящих параметров внешнего события.

### BPEL Container

Выполнение бизнес-процессов и Service Enabler.

### Transaction PL

Поддержка транзакционного режима работы.

### Session PL

Поддержка сессионного режима работы.

Установка и настройка SLES — в разделе 9.

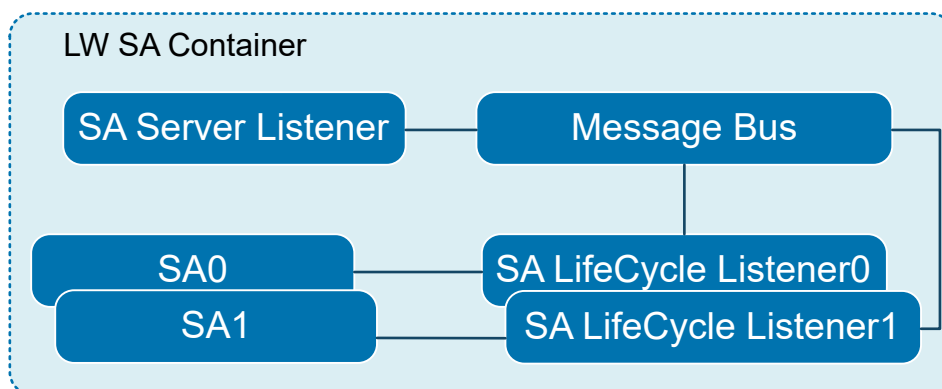
## 2.3. Контейнер LW SA Container

Система LW SA Container — контейнер на базе Tomcat, который используется для подключения SA к RTSIB. Используется для всех SA, в том числе и SA с доступом к JDBC DataSource, а также JAXWS SA и SADB.

Функции LW SA Container:

- Развертывание SA, интегрированных в RTSIB.

- Развертывание SA, которые работают с использованием JAXWS.
- Обслуживание жизненного цикла SA.
- Предоставление ресурсов веб-контейнера.



**Рис. 8. Архитектура LW SA Container**

Установка и настройка — в разделе [10](#).

### 2.3.1. Нотификации

*LWSA Container с помощью JMS-адаптера получает с сервера сообщения о событиях и затем отправляет соответствующие нотификации всем подписанным получателям — компонентам.*

#### Нотификации от Oracle Server

Сценарий состоит из следующих шагов:

- Система помещает поступившее сообщение в очередь соответствующего типа.
- JMS-адаптер выбирает поступившее сообщение из очереди. Для каждой очереди используется свой JMS-адаптер.
- JMS-адаптер отправляет сообщение компонентам, которые подписаны на данный тип сообщений.

#### Нотификации от PostgreSQL Server

Сценарий состоит из следующих шагов:

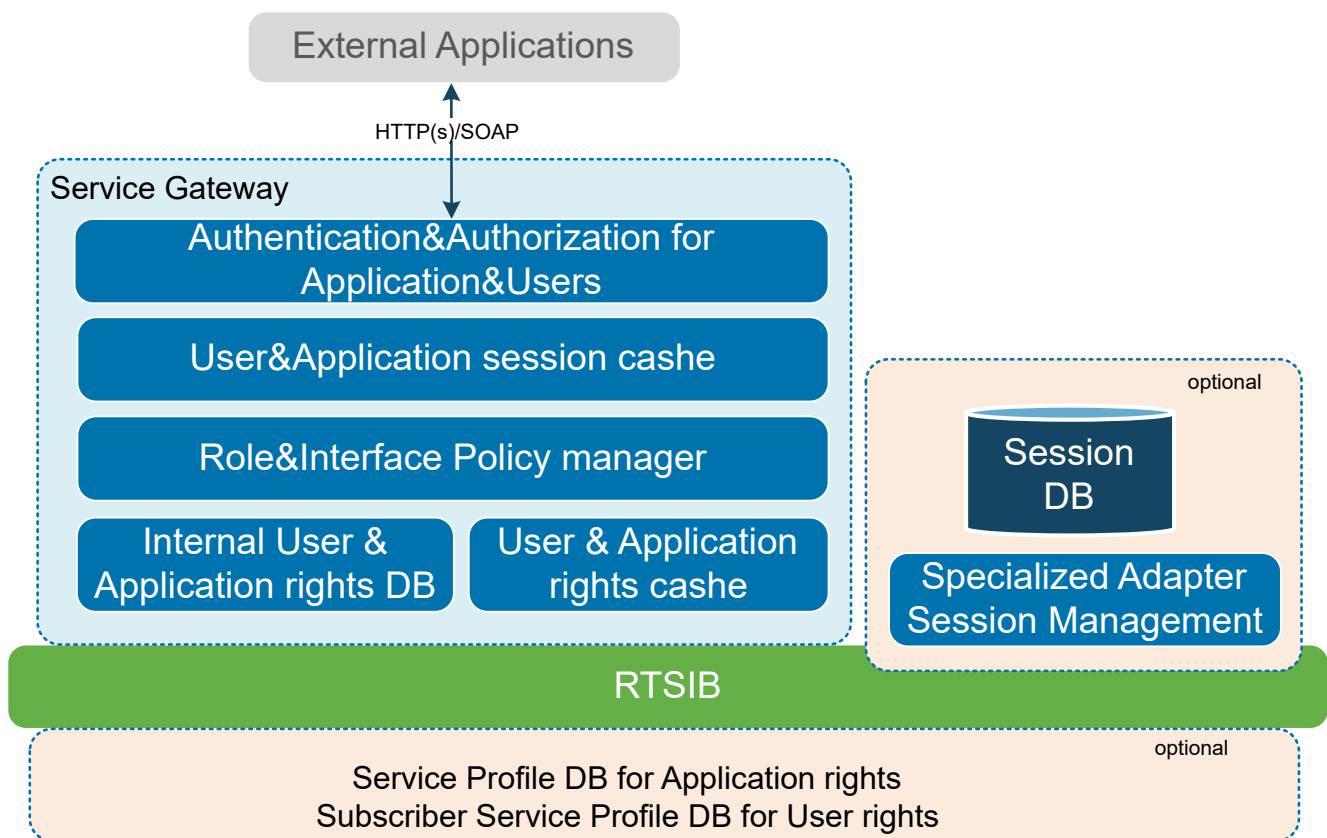
- Триггер срабатывает при наступлении события и создает сообщение для нотификации.
- Система сохраняет сообщение в БД.
- Notification SA с заданным интервалом опрашивает БД и выбирает поступившие сообщения.
- Система отправляет сообщения всем адресатам, подписанным на событие. Перечень адресов доставки формирует шина RTSIB по параметрам корреляции, маршрутизации и масштабирования.

## 2.4. Внешний доступ (Service Gateway)

Система Service Gateway служит для подключения внешних приложений и предоставления доступа к интерфейсам Platform v3.

Система Service Gateway выполняет следующие функции:

- Подключение внешних приложений по протоколу HTTP(S) и протоколу на базе SOAP.
- Использование HTTP Basic, HTTP Digest, OAuth 2.0 или другой индивидуальной логики аутентификации внешних запросов.
- Авторизация запросов для сессий с использованием cookies.
- Наличие внутреннего кеша для хранения текущих параметров авторизации сессий и возможность внешнего хранения для длительных сессий.
- Настройка ролей доступа для ограничения использования внешними приложениями и пользователями интерфейсов Platform v3, вплоть до доступа по определенным шаблонам параметров операций.
- Наличие внутреннего или внешнего хранилища и параметров аутентификации и авторизации внешних приложений и внешних пользователей. В качестве внутреннего система использует MIB-хранилище, в качестве внешнего — SP DB-хранилище.
- Ограничение скорости поступления запросов от внешних приложений и пользователей в течение заданного периода времени.



**Рис. 9. Архитектура Service Gateway**

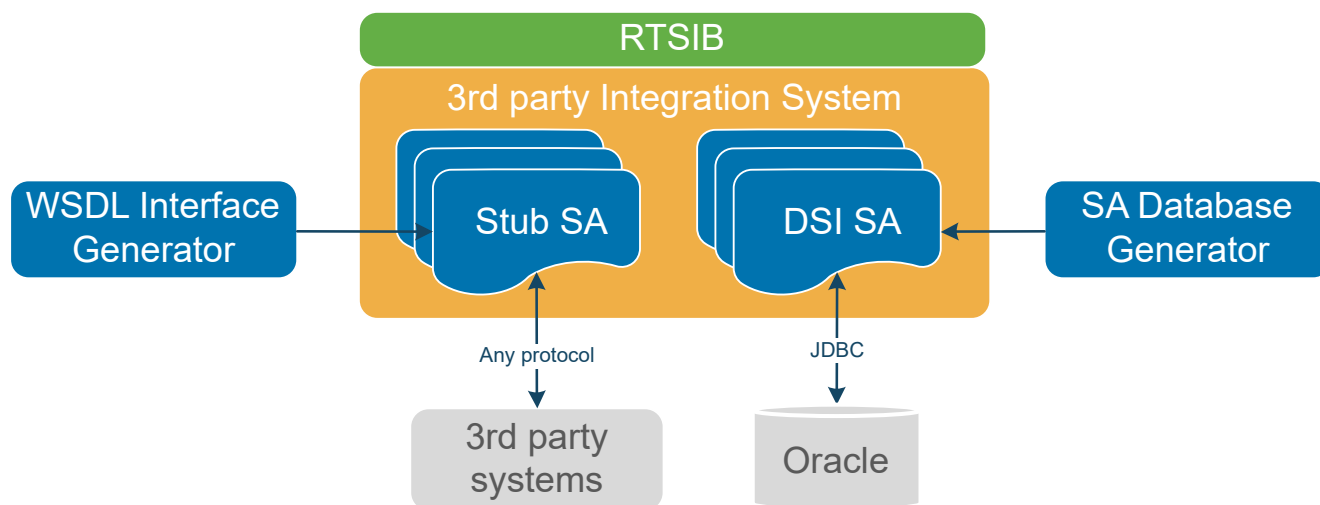
Установка и настройка — в разделе [12](#).

## 2.5. Сервисные системы

### 2.5.1. Взаимодействие со сторонними системами и дополнительными сервисами

*Platform v3* предоставляет набор компонентов и утилит для быстрой настройки и разработки бизнес-ориентированных интерфейсов, а также адаптеры к внешним системам.

Использование компонентов и утилит позволяет быстро интегрировать сервисы внешних поставщиков или дополнительные сервисы оператора.



**Рис. 10. Компоненты и утилиты для быстрой интеграции внешних систем с Platform v3**

#### WSDL Interface Generator

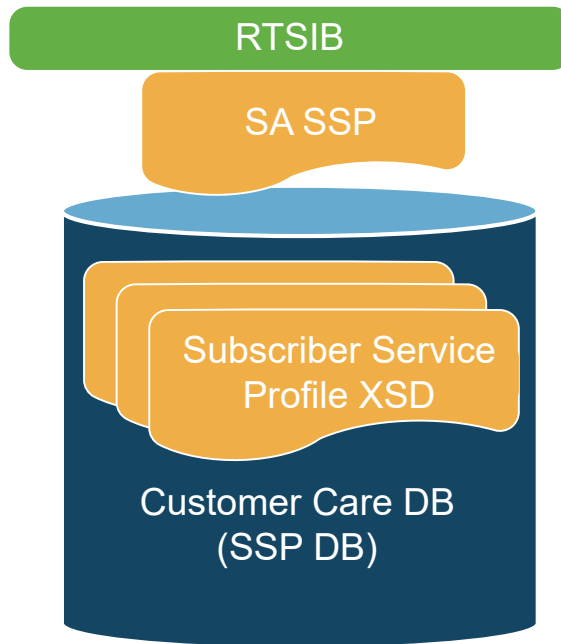
Генератор интерфейсов для создания прототипа Specialize Adapter с требуемым бизнес-интерфейсом на языке Java. В дальнейшем система заполняет прототип Specialize Adapter логикой взаимодействия с внешней системой.

#### SA Database Generator

Генератор Specialize Adapters для баз данных. Выполняет полную генерацию интерфейса и Java-кода по пакету процедур в базе данных Oracle.

## 2.5.2. Система Subscriber Service Profile

*Platform v3* позволяет интегрировать различные сервисы взаимодействия с пользователем.



**Рис. 11. Интеграция с SSP**

### Subscriber Service Profile (SSP)

Система хранения данных по сервисам для конкретного абонента.

#### XSD Profile

Профиль сервиса для абонента. Наличие профиля позволяет разработать сервис с любыми параметрами без изменения интерфейсов платформы.

Установка и настройка Subscriber Service Profile Subsystem — в разделе [13](#).

#### 2.5.2.1. Хранение данных о сервисах пользователей (SSP)

*Для хранения данных о сервисах пользователей на Platform v3 используется подсистема Subscriber Service Profile (SSP).*

С SSP взаимодействуют бизнес-процессы или внешние системы, которые работают с параметрами сервиса пользователя.

В состав SSP входят следующие компоненты:

- SA SSP — специализированный адаптер. Реализует интерфейс DSI Subscriber Service Profile Management и конвертирует вызовы из формата SOAP в хранимые процедуры БД.
- SSP DB — база данных с информацией о профилях услуг пользователей.
- *Словари* типов данных.

### 2.5.3. Service Profile Management

*Service Profile Management* позволяет пользователю создавать описания типов сервиса, добавлять новые сервисы на основе этих типов и управлять их параметрами с помощью набора операций.

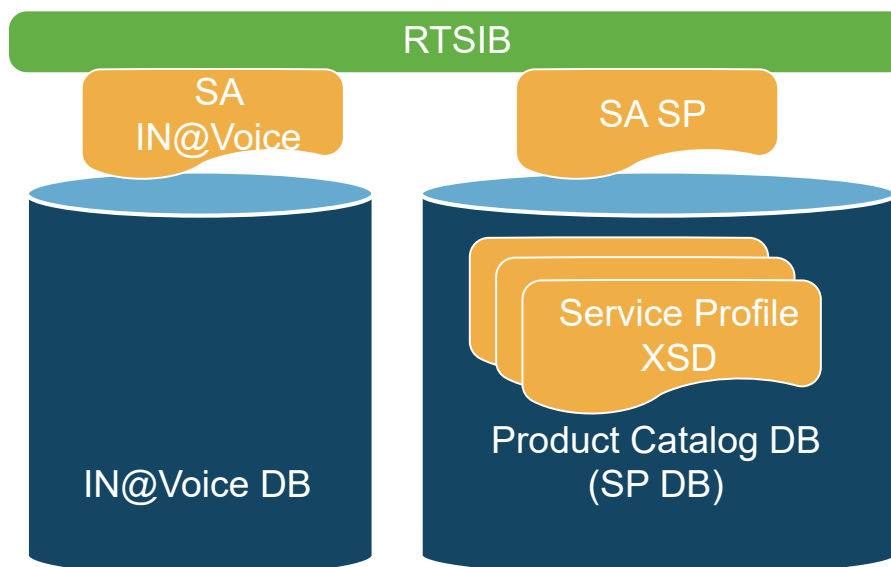
В состав SPM входят следующие компоненты:

- SA SP — специализированный адаптер, который предоставляет интерфейс DSI Service Profile Management и преобразует вызовы из SOAP в SQL.
- SP DB — база данных для хранения сообщений и параметров сервисов.
- *Словари* типов данных.

Установка и настройка Service Profile Management — [15].

### 2.5.4. Взаимодействие с BSS

*Platform v3* позволяет интегрировать сервисы для выполнения биллинговых и других бизнес-операций.



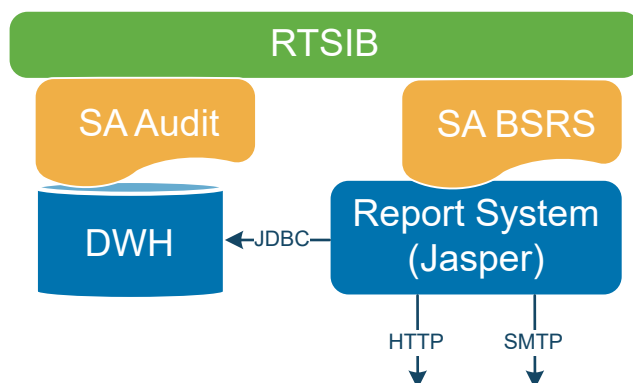
**Рис. 12. Взаимодействие с BSS**

Основные функции:

- единый бизнес-ориентированный интерфейс по доступу к биллинговой информации в IN@Voice для любых сервисов и бизнес-процессов;
- возможность задания любого профиля сервиса в XSD-виде. Это позволяет качественно и быстро выполнять разработку без изменения интерфейсов платформы;
- использование подсистемы хранения профилей сервисов.

## 2.5.5. Взаимодействие с DWH

Platform v3 позволяет формировать статистические и бизнес-отчеты на основе любого обмена данными.



**Рис. 13. Взаимодействие с DWH**

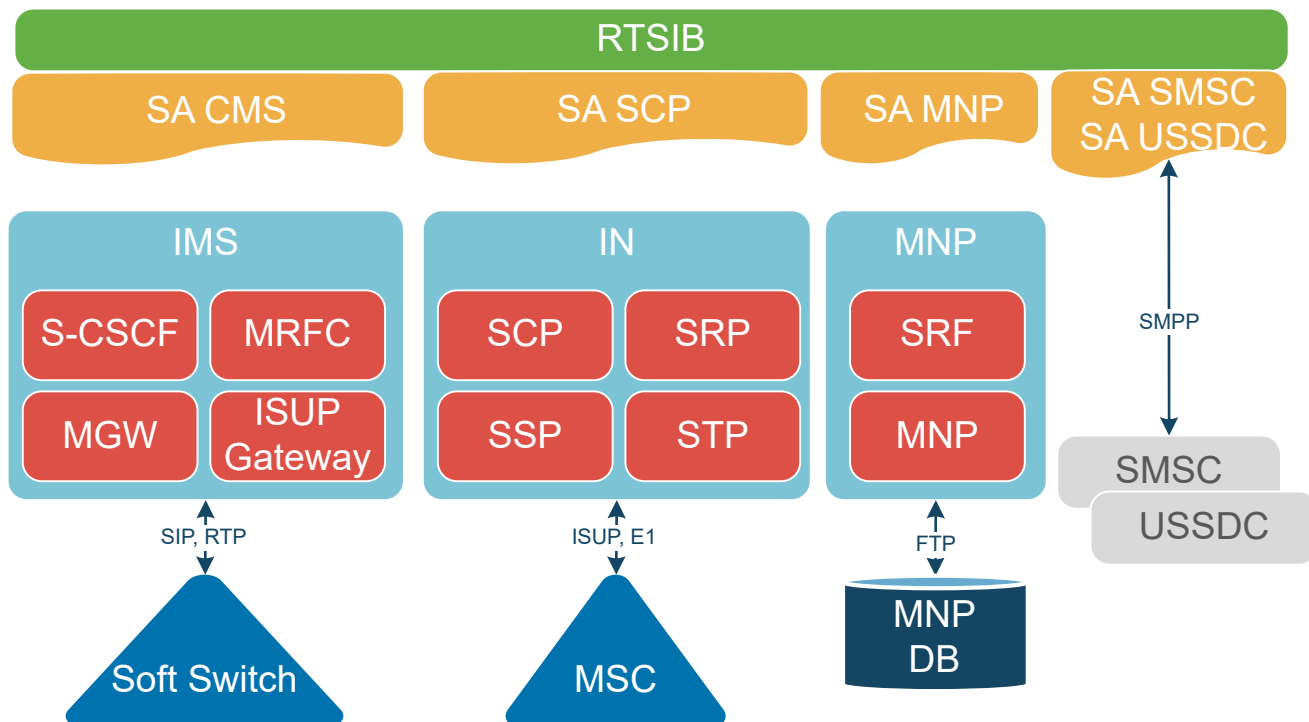
Основные функции:

- сохранение информации о вызовах сервисов и бизнес-процессов в частности:
  - идентификаторов сервиса и бизнес-процесса;
  - времени исполнения операции;
  - параметров операций.
- планирование и отправка отчетов на адрес электронной почты;
- просмотр или скачивание отчетов по HTTP-протоколу.



### 2.5.6. Взаимодействие с мобильной сетью оператора

Platform v3 обеспечивает доступ к сервисам мобильной сети оператора.



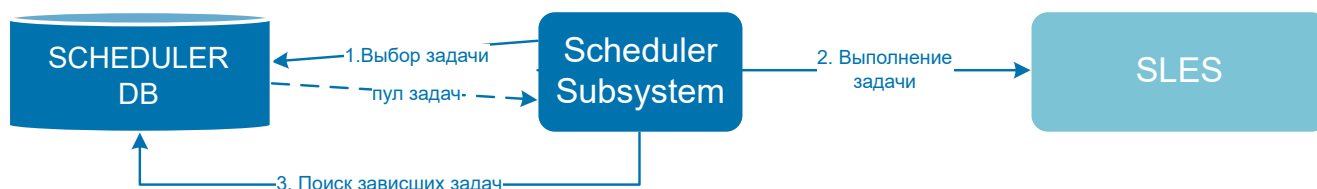
**Рис. 14. Взаимодействие с мобильной сетью оператора**

Основные функции:

- управление голосовым вызовом вне зависимости от типа сети;
- управление SMS- и USSD-сообщениями при неинтерактивном взаимодействии;
- работа с перенесенными номерами других операторов (MNP).

### 2.5.7. Планирование и запуск заданий по расписанию (Scheduler)

Система Scheduler Subsystem управляет задачами по расписанию.



**Рис. 15. Принцип работы системы Scheduler Subsystem**

1. Scheduler Subsystem выбирает задачу из БД.
2. Scheduler Subsystem блокирует задачу для других экземпляров системы и запускает задачу на выполнение. Например, в SLES.
3. Scheduler Subsystem проверяет в БД зависящие задачи. Если они находятся в зависшем состоянии больше заданного интервала, система разблокирует их.

Задачи выбираются по следующему алгоритму:

1. Scheduler Subsystem ищет задачи, у которых интервал события соответствует периоду выборки.
2. Scheduler Subsystem исключает из сформированного списка уже запущенные задачи.
3. Scheduler Subsystem помещает выбранные задачи в таблицу L\_R\_JOBS. Задачи из этой таблицы выполняются группами и распределяются на все работающие экземпляры Scheduler Subsystem. Количество задач в группе настраивает администратор системы.

Установка и настройка — [18].

## 2.6. Системы представления

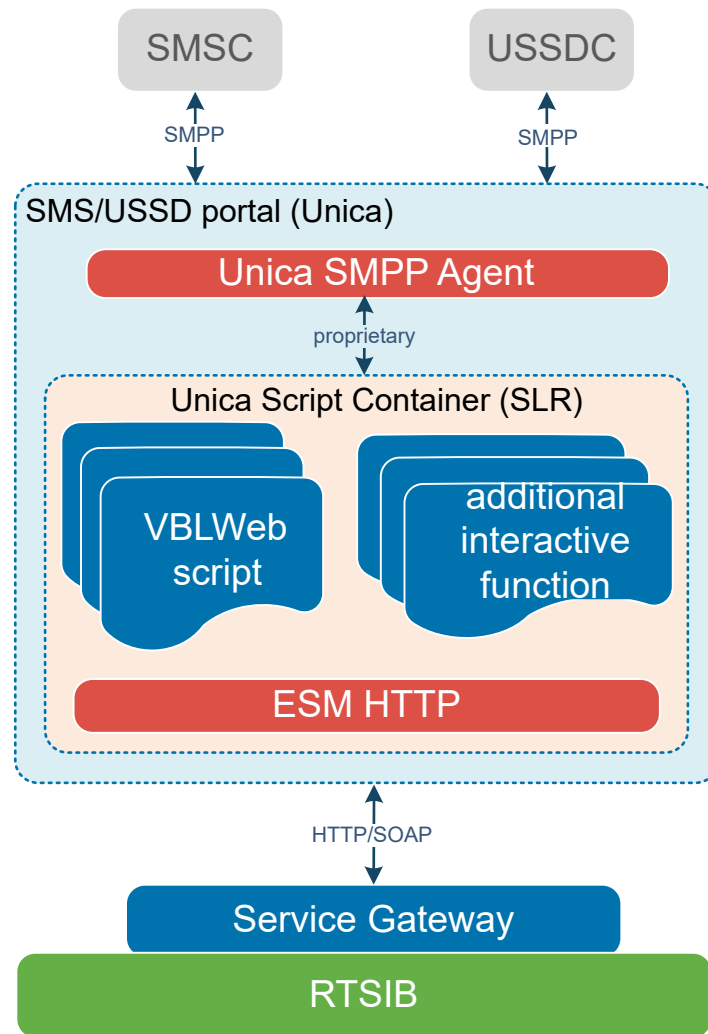
### 2.6.1. Портал взаимодействия по SMS и USSD

*Для выполнения логик при интерактивном текстовом взаимодействии с пользователем предназначен портал — SMS или USSD Portal.*

Портал реализован на базе компонентов платформы Unica.

Соблюдайте следующие правила:

- выстраивайте взаимодействие с интерфейсами платформы с использованием системы Service Gateway;
- используйте логики представления, созданные в редакторе Unica Visual Builder Light Web [7].



**Рис. 16. Архитектура SMS или USSD Portal**

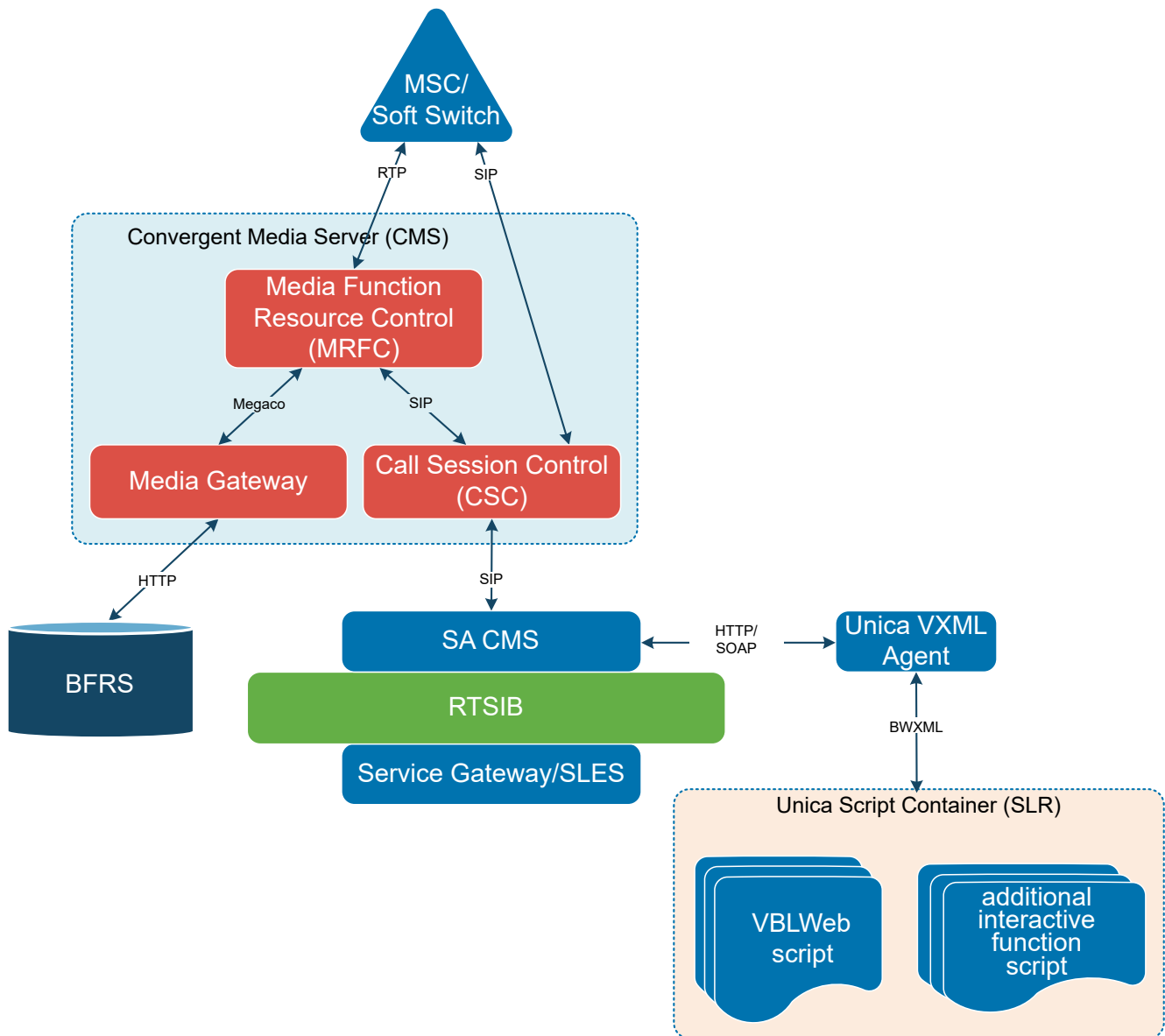
### 2.6.2. Портал голосовой и видеосвязи

Для выполнения логик при взаимодействии с пользователем с помощью голосовых или видео сессий используется портал — Voice/Video Portal.

Портал реализован на базе компонента CMS.

Соблюдайте следующие правила:

- выстраивайте взаимодействие с интерфейсами платформы через Service Gateway или SLES;
- используйте логики представления, созданные в редакторе Unica Visual Builder Light Web [7].



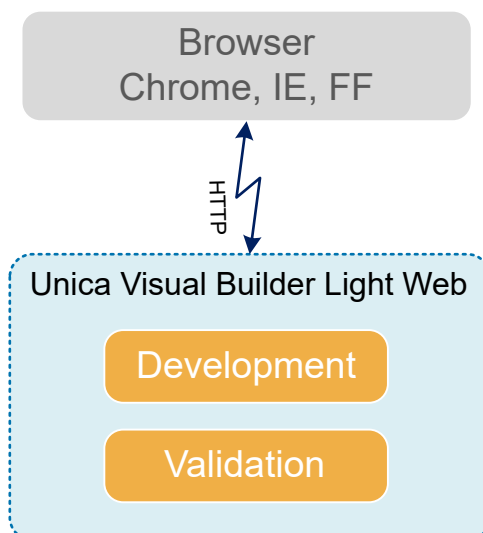
**Рис. 17. Архитектура Voice или Video Portal**

## 2.7. Системы разработки

### 2.7.1. Инструмент создания интерактивных меню

*В качестве среды разработки логик интерактивного взаимодействия с пользователем используется Unica Visual Builder Light Web.*

Unica Visual Builder Light Web [7] визуально выстраивает сессию взаимодействия с пользователем независимо от типа транспорта: по SMS-, USSD-, с помощью голосовой связи и видеосвязи. Создает дерево интерактивных меню и настройки вызовов внешних сервисов и бизнес-процессов. Инструмент представляет собой веб-приложение.



**Рис. 18. Средство визуальной разработки SMS-, USSD-, Voice-, Video-логик интерактивного взаимодействия с пользователем**

Основные функции:

- разработка целостных логик взаимодействия с пользователем;
- разработка дополнительных отдельных сервисных логик для выполнения уточняющих действий, запрашиваемых из бизнес-процесса в зависимости от его условий. Например, в зависимости от параметров в бизнес-процессе может выполняться или не выполняться запрос Advice of Charge.
- проверка логик;
- разграничение прав пользователей — разработка и просмотр логик;
- эмуляция и отладка логик витрин с заданием входящих параметров и ответов на внешние запросы;
- интеграция с системой управления.

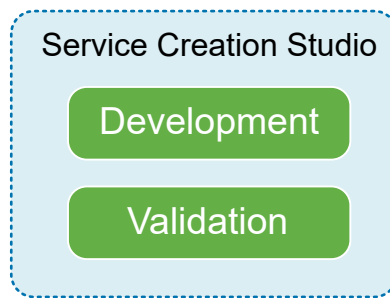
### 2.7.2. Среда разработки бизнес-процессов

Для разработки бизнес-процессов Platform v3 используется программная среда — Service Creation Studio.

Service Creation Studio [20] позволяет визуально отобразить в бизнес-процессе:

- вызов операций сервисов или других бизнес-процессов;
- выполнение простейших арифметических, строковых операций;
- обращение к вспомогательным сервисам для выполнения сложных вычислительных или поисковых действий с полученными данными.

Service Creation Studio запускается в среде разработки NetBeans.



**Рис. 19. Средство визуальной разработки бизнес-процессов**

Основные функции:

- разработка бизнес-процессов;
- проверка бизнес-процессов.

### 2.7.3. Общий словарь типов данных

*Platform v3 использует стандартный набор словарей типов данных.*

Сервисы используют следующие словари типов данных:

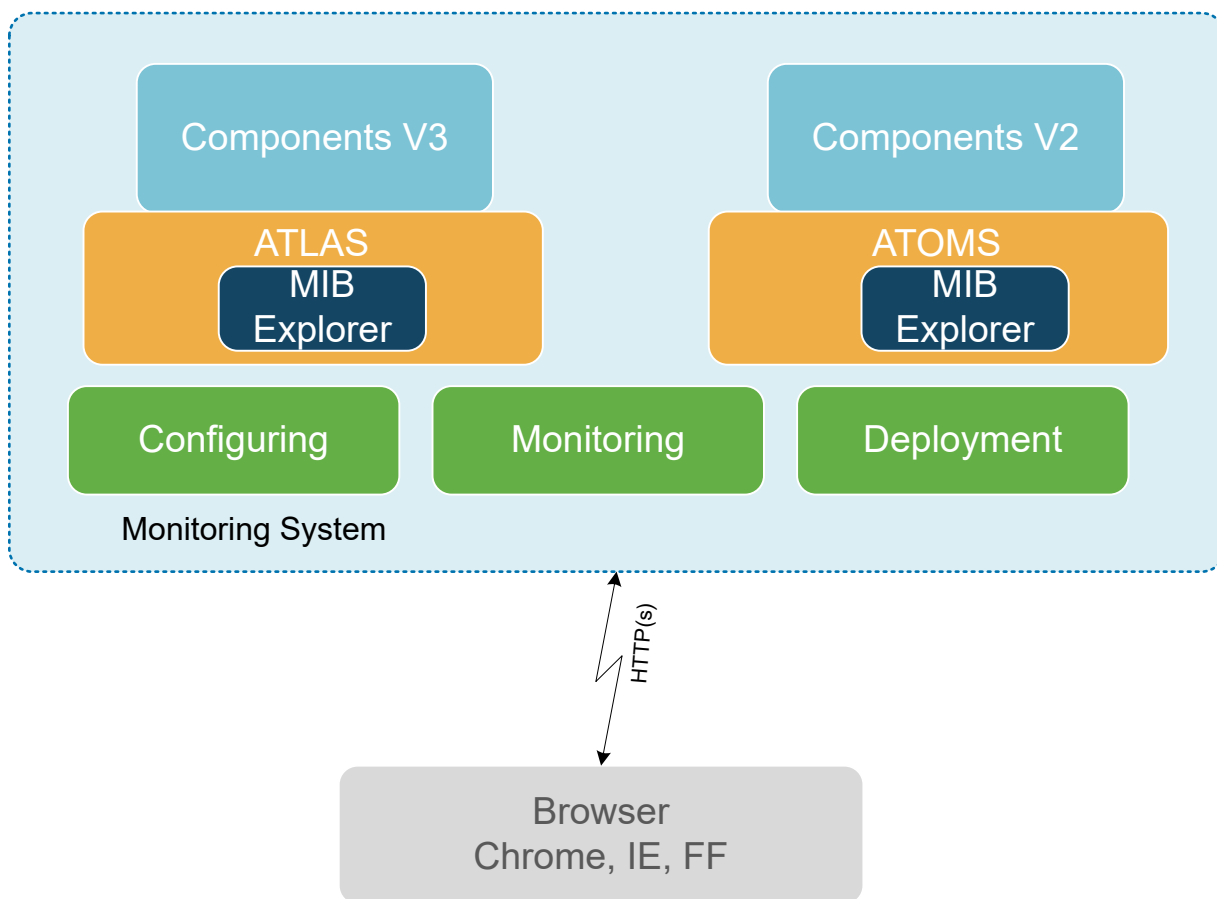
- SimpleDefinition.xsd — словарь простых типов данных. Используется в других артефактах с XSD-типами.
- ComplexDefinition.xsd — словарь составных или комплексных типов данных. Используется в артефактах Platform v3 с XSD-типами.
- ProfileDefinition.xsd — словарь типов данных для профилей. Используется в описаниях профилей сервисов и услуг пользователя Platform v3. Словарь содержит специализированные элементы, дополнительные правила по их формированию, а также пояснения по использованию элементов.
- <имя сервиса>X.xsd — набор файлов, описывающих непосредственно профиль сервиса.

## 2.8. Система управления

*Система управления выполняет мониторинг и конфигурирование компонентов.*

Основные функции системы управления:

- Конфигурирование артефактов и связей Platform v3.
- Конфигурирование компонентов.
- Мониторинг технологических параметров Platform v3.
- Мониторинг всех сущностей, участвующих в решении, с помощью MIB Explorer [1].
- Развертывание артефактов Platform v3.
- Развертывание компонентов решения.
- Визуализация состояния компонентов Platform v3 в виде дерева.



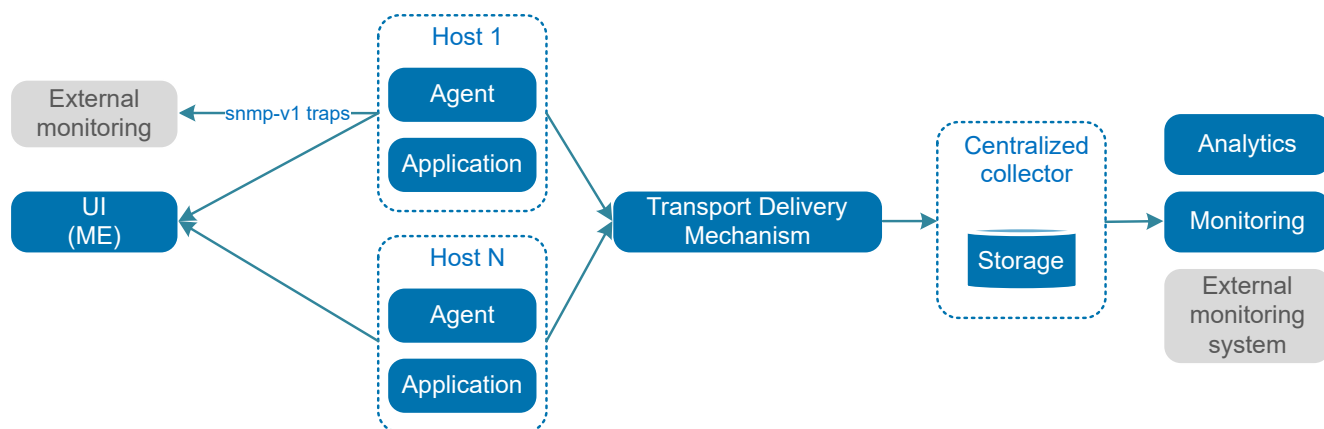
**Рис. 20. Архитектура системы управления**

### 2.8.1. Система Operations&Management

Система *Operations&Management* выполняет мониторинг распределенных приложений.

Мониторинг подразумевает следующие задачи:

- логирование сообщений по принципам построения Unified Logging Layer (ULL).
- трассировка цепочек вызовов сообщений на основе Open Tracing (OT).



**Рис. 21. Принципы Open Tracing и Unified Logging Layer**

1. Агент собирает логи и метрики на своем хосте.
2. Система отправляет данные от агента в централизованной коллектор.
3. Централизованной коллектор передает собранные данные системам аналитики, мониторинга и другим внешним системам.

Функциональные возможности системы Operations&Management:

1. Загрузка лога сервиса с хостов, на которых он развернут. Системе неизвестны адреса хостов, доступ к данным для мониторинга выполняется централизованно из специализированных систем Grafana\Kibana и соответствует сервису (Business Application).
2. Построение цепочки распределенных вызовов действия (трассировка). Цепочки передаются средствами централизованной транспортной системы (Kafka).
3. Построение зависимостей метрик приложения и логов в заданный момент времени.
4. Развертывание сервиса (Business Application) как Virtual Network Feature (VNF). Исключение проблемы хранения данных мониторинга на локальных хостах при отмене функции.
5. Подключение внешних систем мониторинга с помощью топиков системы Kafka. Система Kafka получает данные приложений Bercut в формате JSON.

Логи и метрики хранятся централизованно и привязываются к сервису (Business Application). Приложение MIB Explorer может использовать информацию, которая хранится в агенте.

**i** ***Примечание.** Внешние системы мониторинга можно подключить по SNMP-протоколу к агенту.*

### **Базовый сценарий работы системы**

Сценарий работы включает в себя следующие основные шаги:

1. Сервис отправляет данные о лог-записях и о трейсовых сообщениях в агент.
2. Агент отправляет данные в систему Kafka — в специализированные топика для логов и трейсов.
3. Компонент logDecoder преобразует бинарные логи из системы Kafka в текстовые логи с помощью справочников. Затем отправляет логи в отдельный топик в системе Kafka.
4. Компонент traceDecoder по трейсовым записям из системы Kafka склеивает цепочки вызовов. Получившуюся последовательность запросов и ответов компонент traceDecoder отправляет в отдельный топик в систему Kafka.
5. Компонент logstash сохраняет в хранилище elasticsearch декодированные логи и объединенные цепочки вызовов.
6. Компоненты Grafana или Kibana используют данные из хранилища и отображают их в веб-интерфейсе.

**i** ***Примечание.** Компоненты сервиса могут быть развернуты на разных хостах.*



## Архитектура системы

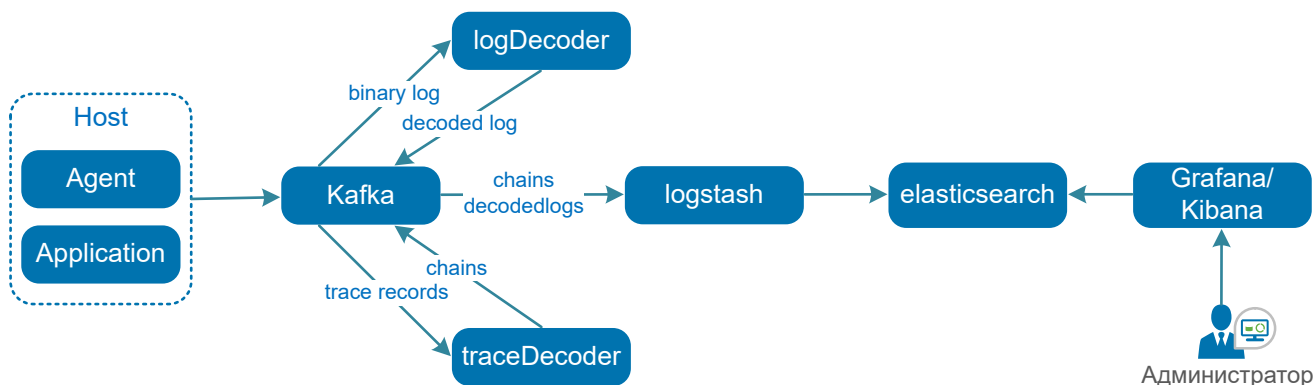


Рис. 22. Архитектура Operations&Management

- Agent — ATLAS Agent [2]. Мониторинг приложений Bercut.

**! Внимание!** Используйте ATLAS Agent версии не ниже 2.0.

### Задачи:

- получение логов с помощью пользовательских Java-библиотек от компонента сервиса и платформенных систем;
- получение трейсовых записей от платформенных систем о событиях в цепочке вызовов;
- создание топиков `log-<BA name>` и `trace-<BA name>` в системе Kafka при первом обращении приложения для логирования или трассировки;
- отправка бинарных данных логов в топик `log-<BA name>` системы Kafka.

**i Примечание.** Если имя *BusinessApplication* неизвестно, название топика в системе Kafka: `log-default`.

- отправка данных трейса в топик `trace-<BA name>` системы Kafka.

**i Примечание.** Если имя *BusinessApplication* неизвестно, название топика в системе Kafka: `trace-default`.

Форматы бинарных и трейсовых записей в топиках приведены в 15.

- Kafka — транспортный узел доставки сообщений в систему мониторинга. Задачи:
  - получение сообщений от агентов с хостов;
  - отправка сообщений в компоненты: `logDecoder`, `traceDecoder`, `logstash`;
  - временное хранение сообщений от агентов, если отсутствуют компоненты, которым нужно отправить сообщения.
- `logDecoder` — декодирует бинарные логи. Задачи:
  - извлечение из топика `log-<BA name>` системы Kafka бинарных логов;
  - декодирование логов по `Id`-справочникам, добавление шаблонных текстов;

**i Примечание.** Загрузите заранее `Id`-справочники и бинарные декодеры в локальную файловую систему.

- отправка декодированного лога в топик `decodedlog-<BA name>` системы Kafka в формате JSON.

**i Примечание.** Если имя *BusinessApplication* неизвестно, то название топика в системе *Kafka* будет следующим: *decodedlog-default*.

- `traceDecoder` — объединяет трейсовые записи в цепочки вызовов: иницирующие, промежуточные и ответные. Задачи:
  - извлечение из топика `trace-<BA name>` системы *Kafka* трейсовых сообщений;
  - склейка в единую цепочку трейсовых сообщений приложения, которые относятся к одному инициализирующему вызову;
  - декодирование данных сообщения, если данные закодированы в формате `FastInfoSet`;
  - отправка цепочки вызовов в топик `chain-<BA name>` системы *Kafka* в формате `JSON`.

**i Примечание.** Если имя *BusinessApplication* неизвестно, название топика в системе *Kafka*: *chain-default*.

- `logstash` — читает из канала метрики, логи, трейсы и передает их в компонент `elasticsearch`. Дополнительно трансформирует и выполняет обфускацию (маскирование) параметров сообщений.
- `elasticsearch` — хранит и индексирует логи, метрики, трейсы. Источник данных для компонентов `Grafana\Kibana`.
- `Grafana\Kibana` — обеспечивает доступ к данным компонента `elasticsearch` и отображение их на `dashboard`. Администратор может выполнить поиск, просмотр данных и настройку аларминга.

Установка и настройка — [15](#).

## 3. Принципы построения бизнес-приложений

*Platform v3 построена по принципам сервисно-ориентированной архитектуры (Service-Oriented Architecture, SOA).*

Концепция SOA использует следующие принципы:

- Каждая внешняя система представляет собой один или несколько сервисов с бизнес-ориентированным интерфейсом (AOI), который с помощью специализированного адаптера публикуется в общую шину (RTSIB).
- Координацию взаимодействия сервисов выполняют бизнес-процессы (BP) и вспомогательные сервисы (SE). BP и SE предоставляют интерфейс (AOI) для взаимодействия с другими бизнес-процессами и сервисами.
- Бизнес-приложение — это совокупность бизнес-процессов и сервисов, выполняющих определенную бизнес-задачу оператора. Бизнес-приложение направлено на оказание услуги или логического набора услуг для пользователей и (или) внешних приложений.
- Абоненты получают доступ к функциональным возможностям бизнес-приложения из мобильной сети оператора с помощью SMS/USSD- и Voice/Video-витрины.
- Конфигурационные параметры сервисов, продуктов и систем хранятся в XSD-профилях системы Service Profile Management (SPM).
- Конфигурационные и текущие параметры сервиса для пользователя описываются в XSD-профилях системы Subscriber Service Profile Subsystem (SSP).
- Доступ для внешних приложений к функциональным возможностям бизнес-приложений реализует Back Office, Front Office или другие веб- и SOA-приложения.

Бизнес-процесс может:

- объединять несколько AOI для выполнения совокупных операций, связанных с разными источниками;
- декомпозировать сложную операцию с обращением к внешним данным на ряд более простых операций.

Platform v3 использует возможности сторонних систем за счет преобразования (адаптации) протоколов по доступу к сторонним системам в бизнес-ориентированные интерфейсы (AOI).

Аутентификация и авторизация пользователей и внешних приложений реализована с помощью единой точки доступа для третьих лиц и (или) приложений к бизнес-ориентированным интерфейсам Platform v3. Компонент Service Gateway выполняет разграничение прав доступа. В Platform v3 используются следующие виды аутентификации:

- HTTP Basic
- HTTP Digest access
- аутентификация по IP-адресу.

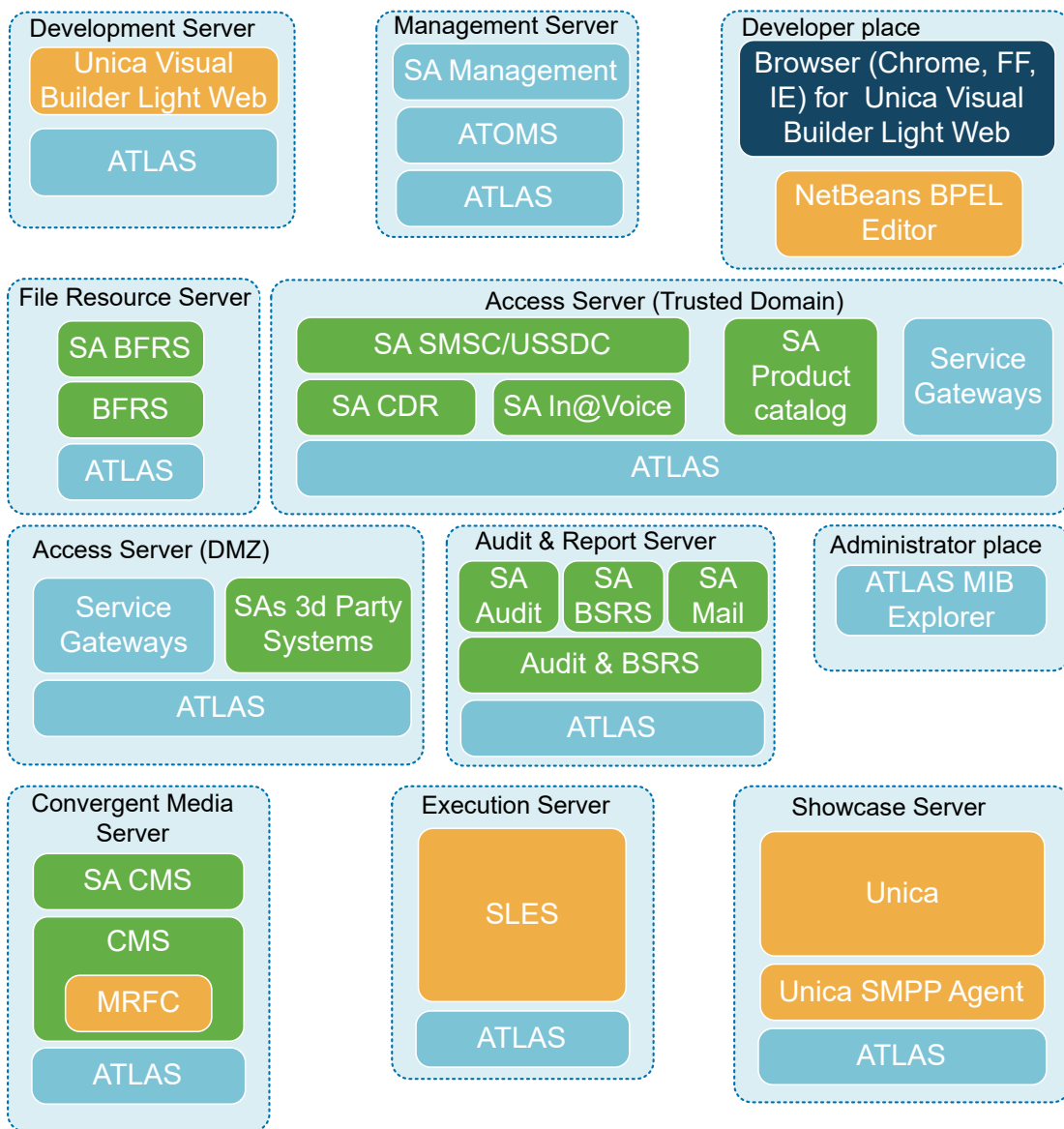
Platform v3 поддерживает учет версионности бизнес-интерфейсов на уровне их WSDL-описаний.

## 4. Требования

### 4.1. Аппаратное обеспечение

Требования к оборудованию, на которое устанавливаются компоненты Platform v3, согласовываются представителями компании Bercut и представителями заказчика.

Компоненты Platform v3 устанавливаются на серверы в соответствии с диаграммой развертывания.



**Рис. 23. Диаграмма развертывания компонентов Platform v3**

Характеристики оборудования, на котором устанавливается Platform v3, зависят от нагрузки, производительности БД и количества экземпляров. Конфигурация оборудования определяется конкретно для каждого решения.

Минимальные требования к серверу **Administrator place**:

- CPU: Intel Xeon 5650 2.66 GHz.
- RAM: 4Gb.
- HDD: 40Gb.

**Минимальные требования к серверу Developer place:**

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 4Gb.
- HDD: 8Gb.

**Минимальные требования к серверу Management Server:**

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 4Gb.
- HDD: 40Gb.

**Минимальные требования к серверу Showcase Server:**

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 3Gb.
- HDD: 40Gb.

Конфигурация этого сервера зависит от производительности — количества сессий, стартующих в секунду — и размера сценария. Данные приведены для производительности 100 сессий/с.

**Минимальные требования к серверу Development Server:**

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 2Gb.
- HDD: 40Gb.

**Минимальные требования к серверу File Resource Server:**

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 1Gb.
- HDD: 70Gb.

Размер HDD этого сервера зависит от объема хранилища BFRS.

Минимальные требования к серверам **Execution Server**, **Convergent Media Server**, **Access Server (DMZ)** и **Access Server (Trusted Domain)**:

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 4Gb.
- HDD: 40Gb.

Минимальные требования к серверу **Audit & Report Server**:

- CPU:
  - OC Linux: 2 vCPU E5-2603v3 2.66 GHz.
  - OC Solaris: 2 vCPU T5 2.66 GHz.
- RAM: 4Gb.
- HDD: 40Gb.

Размер HDD этого сервера зависит от объема статистических данных, хранящихся в БД.

## 4.2. Программное обеспечение

*Для работы системы требуется определенное системное и специальное программное обеспечение.*

Системное и прикладное ПО устанавливаются на серверы в соответствии с [диаграммой развертывания](#).

### Требования к системному программному обеспечению

Platform v3 работает под управлением базовой ОС — Red Hat Enterprise Linux (RHEL) 6.3 или CentOS. Также компоненты Platform v3 могут функционировать под ОС Solaris SPARC версии не ниже 10. На сервер **Administrator place** устанавливается ОС Windows 7, 8 или 10.

Конкретный состав и версия системного программного обеспечения определяются по согласованию с Заказчиком в зависимости от требований, предъявляемых к системе.

Все используемое системное программное обеспечение должно быть лицензировано. Порядок лицензирования определяется поставщиками программного обеспечения. Применение нелицензионного программного обеспечения недопустимо.

На операционные системы должны быть установлены обновления, выпускаемые их разработчиками.

Специализированное программное обеспечение должно удовлетворять спецификациям, заявленным компанией Bercut.

### Требования к прикладному программному обеспечению

На серверах, где устанавливаются компоненты Platform v3, а также на рабочих местах персонала, выполняющего администрирование Platform v3, устанавливается следующее прикладное программное обеспечение:

- Oracle JDK версии 13.0.2 и выше. Устанавливается на все серверы.
- Компонент ATLAS Core версии не ниже 1.6. Устанавливается на все серверы.

- Компонент ATLAS MIB Explorer версии 1.3.5. Устанавливается на сервер **Administrator place**.
- Apache Tomcat версии 7 и выше. Устанавливается на серверы **Development Server** и **Showcase Server**.
- Apache Ant версии 1.8.4 и выше. Устанавливается на серверы **Developer place** и **Execution Server**.
- Веб-браузер (на усмотрение заказчика):
  - InternetExplorer версии 9.0 и выше;
  - Mozilla FireFox версии 15 и выше;
  - GoogleChrome версии 22 и выше.

Устанавливается на сервер **Developer place**.

### 4.3. Персонал

*Для профессиональной установки, настройки и технического обслуживания Platform v3 персонал должен обладать профессиональными навыками.*

Подготовку к работе и ввод в эксплуатацию Platform v3 выполняют представители компании Bercut при участии представителей заказчика.

#### **Требования к численности персонала**

Для обслуживания системы оператору требуется не менее трех штатных единиц:

- Администратор Linux;
- Администратор Solaris;
- Технический администратор.

#### **Требования к квалификации персонала**

Для работы с офисным оборудованием технический персонал оператора должен быть допущен к работе после обучения правилам техники безопасности и соответствующей квалификационной аттестации по электробезопасности.

Обслуживающий технический персонал должен иметь высшее профильное образование. Технический администратор должен иметь сертификаты компании Bercut.

## 5. Подготовка к установке

Проверьте, установлены ли Oracle JDK, система ATLAS, а также переменные `JAVA_HOME`, `ANT_HOME` и `PATH`.

Перед установкой и настройкой компонентов Platform v3 убедитесь, что на соответствующих серверах:

- Установлена программа Oracle JDK версии 7 и выше. Для проверки версии установленной программы Java выполните команду:

```
java -version
```

Система выведет ответное сообщение:

```
java version "1.7.0_71"
```

```
Java(TM) SE Runtime Environment (build 1.7.0_71-b02)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 20.7-b01, mixed mode)
```

- Установлен Apache Tomcat версии не ниже 7.
- Создан пользователь **bercut**.
- Установлен веб-браузер.
- Установлена *система ATLAS Core* версии 1.6 и выше.
- Установлено приложение ATLAS MIB Explorer версии 1.3.5 и выше [1].
- В приложение ATLAS MIB Explorer добавлены *справочники* всех устанавливаемых компонентов, а также их *XML-описания*.
- Для ОС Linux и Solaris: заданы переменные `JAVA_HOME`, `ANT_HOME` и `PATH` в файле настроек `.profile`. Файл настроек располагается в директории `/home/bercut` для ОС Linux и `/export/home/bercut` для ОС Solaris. После внесения изменений в файл настроек `.profile` перезапустите сессию для применения изменений.
- Для ОС Windows: заданы переменные окружения `JAVA_HOME`, `ANT_HOME` и `PATH`.

**Примечание.** Для всех ОС в переменных задайте:

- Путь к Java в переменной `JAVA_HOME`.
- Путь к исполняемому файлу Java в переменной `PATH`.
- Путь к Apache Ant в переменной `ANT_HOME`.

Список и версии ПО приведены в разделе «*Программное обеспечение*».

Установка каждого компонента описана в соответствующем разделе данного руководства.

### 5.1. Установка системы ATLAS Core

Установите компоненты системы ATLAS на каждый сервер, где работают компоненты систем Bercut.

Система ATLAS Core включает в себя компоненты:

- ATLAS Agent — предназначен для хранения параметров конфигурации приложений, мониторинга их состояния и записи в журнал событий, на которые необходимо отреагировать.
- Start Stop Manager — предназначен для выполнения автоматического запуска и останова Bercut-приложений.



- SysInfo — предназначен для решения задачи удаленного контроля нагрузки и работоспособности рабочих станций.

Для установки компонентов системы ATLAS Core:

1. Установите пакет libstdc++ версии 4.1 и выше.
2. Установите приложение ATLAS MIB Explorer [1] на компьютер под управлением ОС Windows.
3. Настройте приложение ATLAS MIB Explorer для администрирования ATLAS Core. Для этого:

3.1. В каталог `$ME\directories` скопируйте файлы `LDSDK-x.ld` и `DECODER_DEFAULT-x.ld`.

3.2. В каталог `$ME\directories` скопируйте файлы справочников трейсовых сообщений (\*.ld) компонентов ATLAS Core. Файлы .ld располагаются в каталоге `$ATLAS_Core\directories\`.

`$ME` — каталог, в котором установлено приложение ATLAS MIB Explorer на рабочем месте администратора;

`$ATLAS_Core` — каталог дистрибутива с установочными файлами ATLAS Core.

3.3. В каталог `$ME\descriptions` скопируйте XML-описание компонента ATLAS Agent. XML-описание компонента ATLAS Agent находится в каталоге `$ATLAS_Core\descriptions\`.

4. Запустите ATLAS Core:

4.1. Для ОС Linux:

Переключитесь на пользователя `root` и выполните команду:

```
# rpm -i atlas-agent-N.N.N-NN.el5.x86_64.rpm
```

По окончании установки система выведет следующие сообщения:

```
Запускается ATLAS Agent: [ OK ]
```

```
Запускается Bercut Applications: [ OK ]
```

4.2. Для ОС Solaris:

1. Переключитесь на пользователя `root`.

2. Создайте каталоги для установки:

```
mkdir /opt/BERCatlas
```

3. Выдайте права для пользователя `bercut` на созданные каталоги:

```
chown bercut /opt/BERCatlas
```

4. Переключитесь на пользователя `bercut`.

5. Распакуйте установочный архив, перейдя в каталог, куда он был скопирован.

```
unzip ./atlas-core-distributive-N.N-solaris10-release.zip -d /opt/BERCatlas/
```

6. Переключитесь на пользователя `root`.

7. Выполните команду:

```
./install
```

5. Проверьте работоспособность ATLAS в приложении ATLAS MIB Explorer [1] или выполните команды:

```
# /sbin/service atlas-agent status
```

```
# /sbin/service bercut-apps status
```

Также вы можете проверить значение переменной *Status* MIB-группы *SMR\_Agent\_v1.6/Core/Statistics/*. При нормальной работе значение переменной – **available**.

Выполните пункты 2-4 на всех серверах, где установлены компоненты систем Bercut.

### 5.1.1. Миграция системы ATLAS Agent 2.0

#### Обновление RPM-пакетов

**i Примечание.** Перед обновлением должны быть соблюдены следующие условия:

1. Предыдущая версия ATLAS Agent должна быть установлена с помощью RPM-дистрибутива:

```
rpm -aq atlas-agent atlas-agent-1.6.21-144.x86_64
```

2. ATLAS Agent и SSM должны быть запущены с помощью сервиса:

```
/sbin/service atlas-agent status
```

Порядок обновления:

1. Остановите сервис atlas-agent:

```
/sbin/service atlas-agent stop
```

2. Обновите пакет:

```
rpm -Uvh atlas-agent-2.0.0-4.x86_64.rpm
```

Результат выполнения:

```
Preparing... ##### [100%]
Updating / installing...
1:atlas-agent-2.0.0-4 ##### [ 50%]
Cleaning up / removing...
2:atlas-agent-1.6.21-144 ##### [100%]
Created symlink from /etc/systemd/system/multi-user.target.wants/atlas-agent.service
to /etc/systemd/system/atlas-agent.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/bercut-apps.service
to /etc/systemd/system/bercut-apps.service.
```

3. Запустите сервис atlas-agent:

```
/sbin/service atlas-agent start
```

**i Примечание.** Все действия выполняйте с правами пользователя *root*.

#### Обновление бинарных файлов

Не используйте обновление бинарных файлов для production-окружений.

**i Примечание.** Перед обновлением должны быть соблюдены следующие условия:

1. Предыдущая версия ATLAS Core должна быть установлена из дистрибутива.
2. ATLAS Agent и SSM должны быть запущены с помощью скриптов:

```
/opt/BERCatlas/bin/agent  
/opt/BERCatlas/bin/bercssm
```

Порядок обновления:

1. Остановите предыдущую версию ATLAS Agent:

```
pkill -f agent.1.6
```

2. Убедитесь, что ATLAS Agent остановлен:

```
ps -ef | grep agent.1.6
```

3. Создайте резервную копию каталога data:

```
cp -R /opt/BERCatlas/data /opt/data
```

4. Удалите предыдущую версию ATLAS Core:

```
rm -rf /opt/BERCatlas
```

5. Разархивируйте дистрибутив ATLAS Core 2.0:

```
unzip /opt/atlas_core_distrib/atlas-core-2.0-rhel-release.zip -d /opt/  
atlas_core_distrib
```

6. Скопируйте новую версию:

```
cp -R /opt/atlas_core_distrib/BERCatlas/ /opt
```

7. Замените каталог data:

```
mv /opt/data /opt/BERCatlas
```

8. Запустите ATLAS Agent:

```
sh /opt/BERCatlas/bin/agent
```

**i Примечание.** Все действия выполняйте с правами пользователя `bercut`.

## 5.2. Общие сведения о настройке компонентов

### 5.2.1. MIB-переменные

Настройку систем и компонентов выполняет технический администратор в базе управляющей информации MIB.

Конфигурационные параметры системы хранятся в базе управляющей информации (MIB). Для доступа к базе используйте систему удаленного администрирования и мониторинга Bercut ATLAS.

**i Примечание.** Рекомендации по настройке компонентов и систем в веб-интерфейсе и с помощью файлов конфигурации указаны в документации, поставляемой вместе с ними.

Настройка систем и компонентов заключается в конфигурировании их параметров в MIB-редакторе приложения ATLAS MIB Explorer.

Каждому экземпляру системы в приложении ATLAS MIB Explorer соответствует своя группа настроек, которая содержит вложенные группы и переменные. Администратор в процессе конфигурирования системы может изменять значения переменных, если переменные доступны для редактирования, а также может добавлять и удалять переменные и группы.

**! Внимание!** *Каталоги и параметры дерева MIB, названия которых начинаются с символа «@», являются служебными, не подлежат редактированию и в данном разделе не описываются. Не изменяйте значения параметров и каталогов, которые начинаются с символа @.*

### 5.2.2. Настройка логирования трейсовых сообщений

*Приложения, компоненты, SE и SA в процессе своей работы генерируют трейсовые сообщения. Администратор системы имеет возможность настраивать объем сообщений, записываемых в трейс.*

В процессе работы приложения генерируют трейсовые сообщения — записи о событиях, которые не влияют на работоспособность компонентов. Каждое трейсовое сообщение относится к определенному уровню детальности событий. Предусмотрено четыре уровня детальности:

- **maximum.** Трейсовые сообщения максимального, среднего и минимального уровней значимости.
- **medium.** Трейсовые сообщения среднего и минимального уровней значимости.
- **minimum.** Наиболее важные трейсовые сообщения.
- **debug.** Сообщения, предназначенные для отладки работы объекта. Данный уровень включает в себя сообщения всех перечисленных уровней.

Для каждого компонента отдельно можно указывать: сообщения каких уровней детальности будут поступать в трейс этого компонента.

Для этого надо задать значение параметра *LogLevel* в группе настроек */Core/Configuration/* или */Configuration/* — для бизнес-процессов.

**! Внимание!** *Рекомендуется задавать уровень *minimum* для регистрации событий каждого компонента системы в штатном режиме. Повышение уровня может привести к снижению производительности системы.*

### 5.2.3. Настройка интервала обновления статистики

*В процессе работы компоненты SE, SA и BP записывают статистические данные, которые характеризуют их работу и отображаются в режиме реального времени в MIB. Администратор может задать настройку обновления числовой статистической информации, которая требует последующей тобработки. Для этого используйте переменную *StatisticsInterval*.*

Значение числовых статистических переменных вычисляется с учетом выбранного типа агрегации — *sum*, *min*, *max* и других — за период обновления статистики приложения. То есть в течение заданного интервала система собирает информацию, затем вычисляется значение статистической переменной и записывается в MIB.

**i Примечание.** Подробная информация о работе со статистикой в системе приведена в руководстве администратора приложения ATLAS MIB Explorer [1].

#### 5.2.4. Загрузка XML-описаний и справочников трейсовых сообщений

Для администрирования систем и компонентов в приложении ATLAS MIB Explorer загрузите XML-описания и справочники трейсовых сообщений.

##### 5.2.4.1. Загрузка XML-описаний

Для загрузки XML-описаний скопируйте файлы в установочный каталог ATLAS MIB Explorer и перезагрузите описания в интерфейсе.

XML-описания компонентов используются для администрирования компонентов в приложении ATLAS MIB Explorer. На основании XML-описаний система формирует дерево с MIB-настройками компонентов.

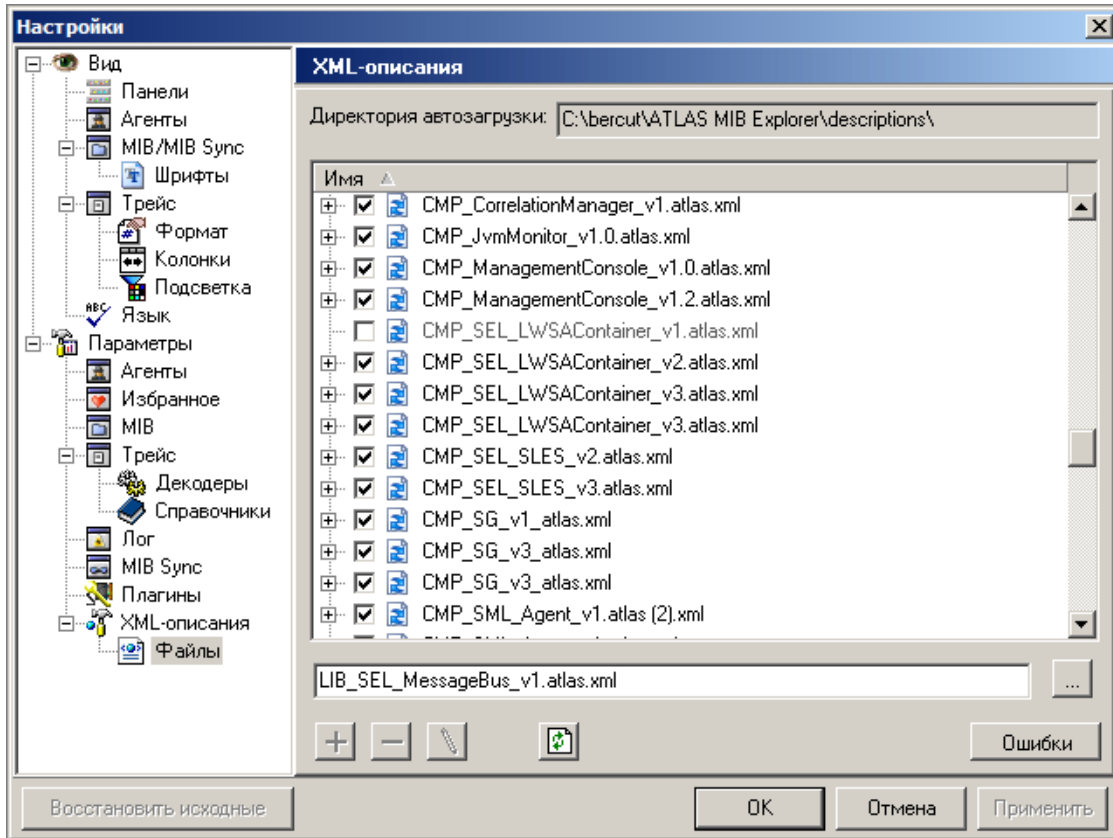
Чтобы добавить XML-описания перед установкой компонента выполните следующие действия:

1. Скопируйте в каталог **\$ME\descriptions\** на рабочем месте администратора файлы с XML-описаниями компонентов. XML-описания находятся в каталоге **descriptions** дистрибутива компонента.


**\$ME** — каталог, в котором установлено приложение ATLAS MIB Explorer.

Система автоматически добавит XML-описания компонентов. Статус загруженного XML-описания можно просмотреть на вкладке **Параметры > XML-описания > Файлы** меню **Опции > Настройки**.

2. Запустите приложение ATLAS MIB Explorer.



**Рис. 24. Загрузка XML-описаний в приложение ATLAS MIB Explorer**

Если приложение ATLAS MIB Explorer уже запущено, нажмите  для перезагрузки описаний вручную.

#### 5.2.4.2. Загрузка справочников сообщений

Справочники трейсовых сообщений (Log Directories) и алармов служат для корректного отображения в ATLAS MIB Explorer тех сообщений, которые поступают от компонентов в процессе работы.

Подробная информация о загрузке справочников сообщений — [1].

## 6. Запуск и останов компонентов системы

Для запуска или останова компонентов Platform v3 используйте SSM (Start Stop Manager).

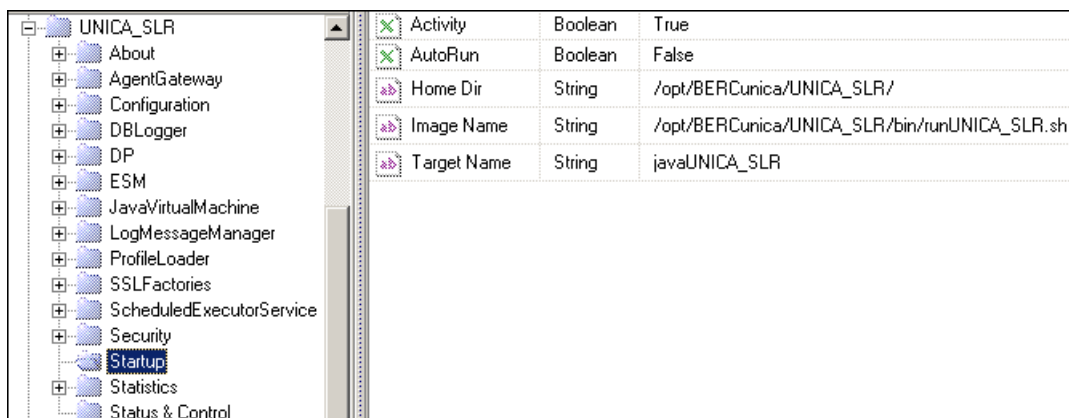
**! Внимание!** Рекомендуется останавливать систему только в случае острой необходимости и только в часы наименьшей нагрузки.

Установка и администрирование компонента SSM — [2].

При установке и запуске компонентов Platform v3 в MIB-дереве каждого компонента система автоматически создает группу Startup с необходимым для SSM набором переменных. В группе Configuration компонента StartStopManager на том же сервере система автоматически создает новую переменную с именем компонента. В качестве значения данной переменной указан путь к папке в MIB, который соответствует компоненту.

Для корректной работы SSM выполните следующие действия:

1. Откройте MIB-дерево компонента, автоматический запуск которого требуется настроить.
2. Создайте в группе требуемого приложения вложенную группу с именем Startup.
3. Создайте в группе Startup следующие обязательные переменные:



Имя переменной	Тип переменной в MIB	Описание
Activity	Boolean	True
AutoRun	Boolean	False
Home Dir	String	/opt/BERCunica/UNICA_SLR/
Image Name	String	/opt/BERCunica/UNICA_SLR/bin/runUNICA_SLR.sh
Target Name	String	javaUNICA_SLR

Рис. 25. MIB-группа Startup

Описание обязательных переменных приведено в таблице.

Имя переменной	Тип переменной в MIB	Описание
Image Name	String	Полный путь к исполняемому файлу приложения.
Activity	Boolean	Флаг удаленной остановки или запуска приложения: <ul style="list-style-type: none"> <li>True — компонент StartStopManager запускает приложение.</li> <li>False — компонент StartStopManager останавливает работу запущенного приложения.</li> </ul>

Имя переменной	Тип переменной в MIB	Описание
		<p><b>i Примечание.</b> Если после запуска приложения с помощью переменной Activity приложение будет остановлено без участия StartStopManager, то приложение будет вновь запущено, независимо от значения переменной AutoRun.</p>
AutoRun	Boolean	<p>Флаг автоматического запуска приложения:</p> <ul style="list-style-type: none"> <li>■ True — в процессе запуска StartStopManager автоматически запускает приложение.</li> <li>■ False — StartStopManager не выполняет автоматический запуск приложения.</li> </ul> <p>Для автоматического запуска в MIB-группе StartStopManager/Configuration должна быть <b>переменная</b>, соответствующая приложению.</p> <p><b>i Примечание.</b> Значение переменной AutoRun не влияет на автоматический запуск и последующую работу приложения. Переменная используется StartStopManager только при запуске приложения, после чего StartStopManager на значение данной переменной не реагирует.</p>

**i Примечание.** Успешная инициализация приложения соответствует появлению переменной Running в группе <название контролируемого приложения>/Status & Control. Тип переменной Boolean, значение True.

4. Создайте новую переменную с произвольным именем типа STRING в группе Configuration компонента StartStopManager на том же сервере. В качестве значения данной переменной укажите путь к MIB-группе, которая соответствует компоненту.



**Рис. 26. Настройка компонента в SSM**

Если перед именем переменной возник знак «-» («минус»), это означает следующее:

- Компонент не настроен для работы с SSM или настройка выполнена неверно.
- Компонент удален.

5. Перезапустите компонент для вступления в силу настроек, то есть применения изменений в дереве MIB настраиваемого компонента.

После выполнения вышеуказанных действий запуск каждого из компонентов можно контролировать с помощью SSM. Подсистема SSM предоставляет также и другие возможности по контролю работы различных служб — [2].



## 7. Настройка Platform v3

Для настройки используйте *MIB Explorer*.

Настройку компонентов Platform v3 выполняйте в следующем порядке:

1. [Запуск компонентов](#).
2. [Конфигурирование портов и настройка взаимодействия между компонентами](#).
3. [Настройка шины RTSIB](#).
4. [Настройка SLES](#).
5. [Настройка источников данных на LW SA Container](#).
6. [Настройка Service Gateway](#).
7. [Настройка условий маршрутизации](#).
8. Настройка конфигурационных параметров [BP](#) и [SA](#).
9. Настройка прочих компонентов Platform v3:
  - [Service Profile Management \[15\]](#).
  - Системы [Subscriber Service Profile Subsystem](#).
  - Системы Correlation System [\[16\]](#).
  - Системы Scheduler Subsystem [\[18\]](#).
  - Системы [Session Management](#).
  - Системы RTSIB Audit [\[19\]](#).
  - Систем [Tarantool RTSIB Driver](#) и [Tarantool SOAP Adapter](#).
  - [SA LDAP](#).

При использовании компонентов платформы Unica дополнительно [настройте](#) Service Gateway для работы с Unica SLR.

**i Примечание.** Когда изменяется первая цифра версии SA, SE и BP (мажорное обновление), при установке система не переносит настройки из предыдущей версии. Но при изменении второй цифры версии (минорное обновление) — система переносит настройки.

## 8. Шина RTSIB

*RTSIB логически объединяет все интерфейсы к сервисам и бизнес-процессам Platform v3, обеспечивая высокую скорость обмена данными.*

*RTSIB* представляет собой набор библиотек, имя которых начинается в префикса `rtsib-`. Библиотеки поставляются в каталоге дистрибутива SLES, LW SA Containen, Service Gateway. Установка RTSIB выполняется при установке соответствующего компонента.

Настройки RTSIB расположены в MIB-группе `@libs`.

ATLAS-контейнер содержит идентификатор сессии (`SessionId`). Идентификатор создан в точке входа запроса в систему Platform v3. `SessionId` вместе с ATLAS-контейнером передается по шине RTSIB в каждом вызывающем сообщении.

### 8.1. Варианты подключения компонентов

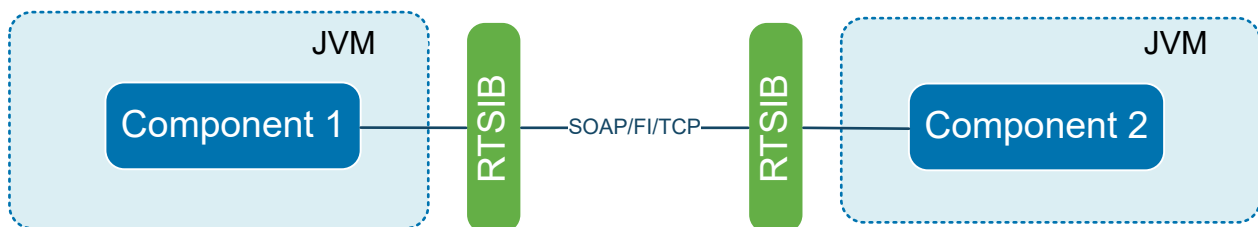
*Взаимодействие с RTSIB — это обмен служебными сообщениям и данными.*

RTSIB использует следующие протоколы:

- SOAP/HTTP
- SOAP/HTTPs
- SOAP/FI/HTTP
- SOAP/FI/HTTPs
- SOAP/TCP
- SOAP/FI/TCP
- SOAP(/FI)/HTTP(s)
- SOAP(/FI)/TCP.

Варианты подключения компонентов с помощью RTSIB:

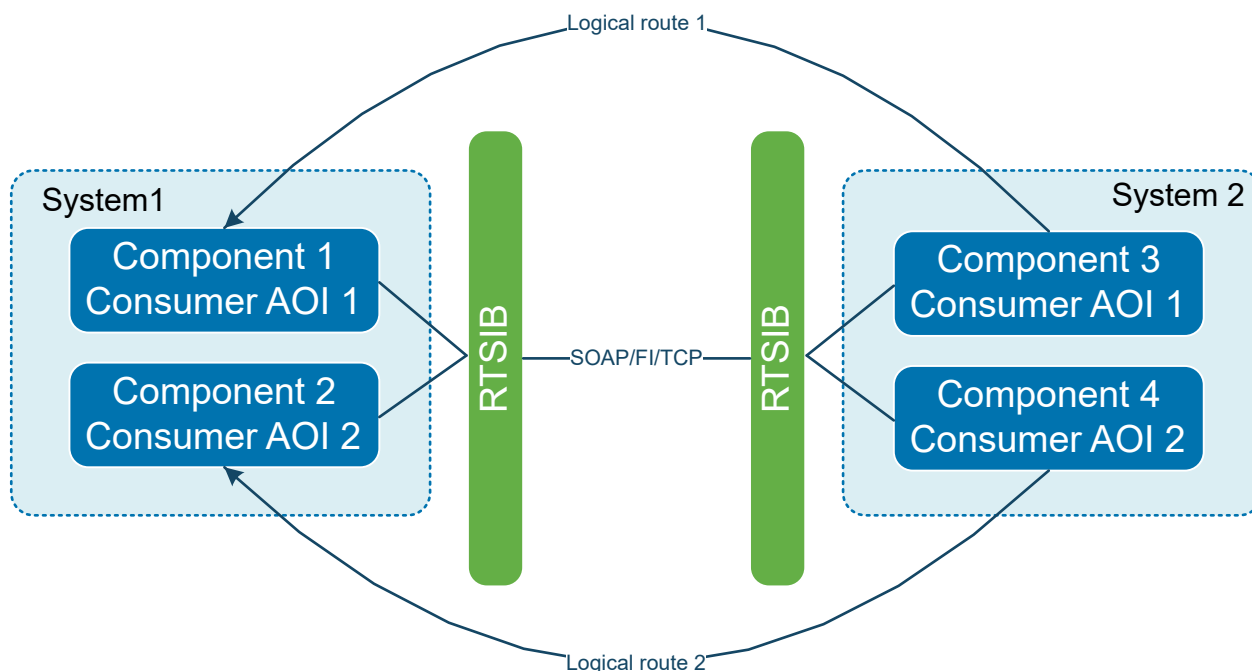
1. Подключение самостоятельного компонента. Выполняет RTSIB-SDK. Для сетевого взаимодействия используется стек SOAP/FI/TCP или SOAP/HTTP.



**Рис. 27. Пример подключения самостоятельного компонента**

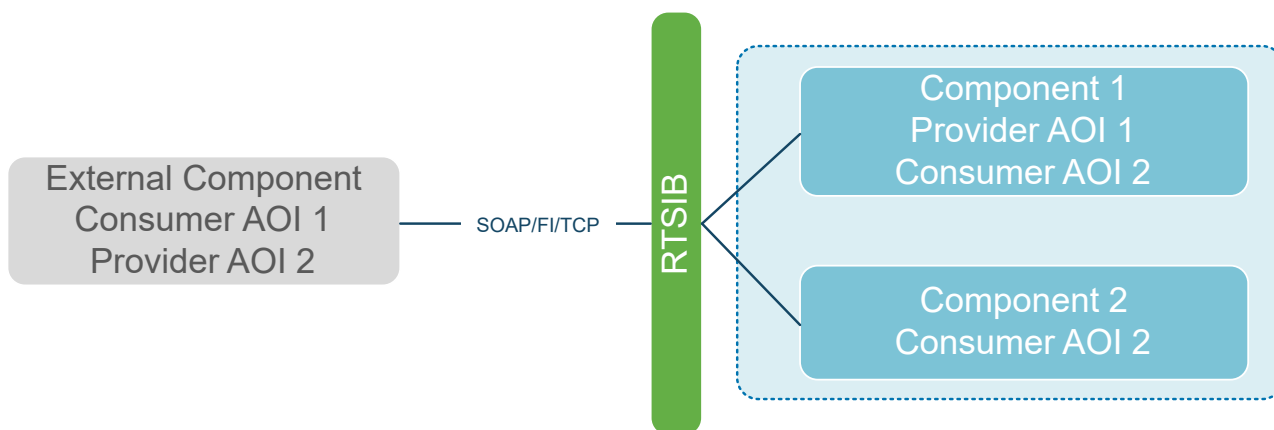
**i Примечание.** При использовании FI (fast infoset) поставьте флаг `UseFastInfoSet` в группе `CMP_SLES_vX.Y/@libs/MessageBus/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp/`. По умолчанию флаг поставлен.

2. Подключение компонентов в среде SLES, UP, Application Server. Для каждого SLES, UP, Application Server используйте один экземпляр RTSIB. Между компонентами двух сред может существовать несколько логических связей. Взаимодействие обеспечивает пул физических соединений, но количество физических соединений не зависит от логических связей.



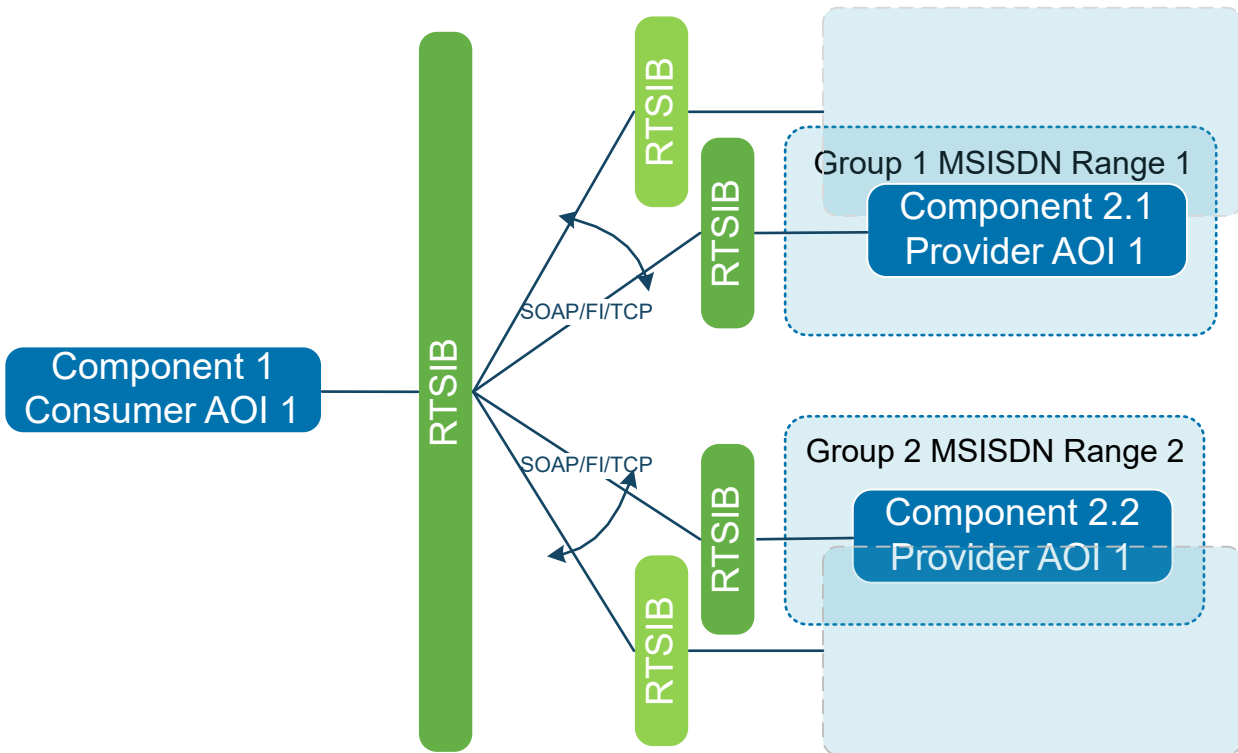
**Рис. 28. Пример подключения компонентов в среде SLES, UP, Application Server**

3. Подключение стороннего компонента, поддерживающего только стек SOAP/HTTP. RTSIB предоставляет точку доступа HTTP. Сторонний компонент может быть как пользователем сервиса, так и предоставлять сервис для компонентов, размещенных в контейнере.



**Рис. 29. Пример подключения стороннего компонента**

Для выбора реализации интерфейса RTSIB обеспечивает маршрутизацию сообщений между контейнерами. При размещении нескольких реализаций одного интерфейса в одном контейнере выбор реализации выполняет контейнер.



**Рис. 30. Маршрутизация сообщений**

Система взаимодействует с RTSIB по проприетарному протоколу по нескольким TCP-соединениям. Для обмена служебными сообщениями система устанавливает одно соединение с ServiceServer AccessPoint. Для обмена данными система устанавливает одно или более соединений с DataServer AccessPoint .

**Серверная часть**

При установке служебного соединения RTSIB отправляет клиенту сообщение с номерами поддерживаемых версий протоколов: служебного и передачи данных. Формат сообщения:

```
<xs:complexType name="ProtocolVersions">
  <xs:sequence>
    <xs:element name="DataProtocolVersion" type="xs:byte"/>
    <xs:element name="ControlProtocolVersion" type="xs:byte"/>
  </xs:sequence>
</xs:complexType>
```

ControlProtocolVersion — версия протокола сообщений для передачи состояния компонентов.

DataProtocolVersion — версия заголовка протокола.

Ответное сообщение от клиента содержит информацию о поддерживаемых клиентом версиях протоколов. Система ожидает ответное сообщение в течение 5 секунд, после окончания которых завершает соединение.

**Клиентская сторона**

При установке соединения с сервером по служебному каналу клиент ожидает 5 секунд сообщение о поддерживаемых сервером версиях протоколов. Когда прошло 5 секунд, система завершает соединение. После получения сообщения клиент передает поддерживаемые им версии протоколов и устанавливает соединения по каналам передачи данных. После обмена версиями протоколов клиент и сервер выбирают наименьшие версии протоколов для передачи данных и передачи служебных

сообщений. Сервер отправляет клиенту информацию о доступных компонентах в виде набора сообщений:

```
<xs:complexType name="ComponentInfo">
  <xs:sequence>
    <xs:element name="ComponentQname" type="xs:QName" minOccurs="0"/>
    <xs:element name="ComponentMessage" type="tns:ComponentMessage"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ComponentMessage">
  <xs:choice>
    <xs:element name="QueueState" type="tns:QueueState"/>
    <xs:element name="ComponentState" type="tns:ComponentState"/>
  </xs:choice>
</xs:complexType>

<xs:simpleType name="QueueState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="StateHwm"/>
    <xs:enumeration value="StateLwm"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ComponentState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Available"/>
    <xs:enumeration value="Not Available"/>
  </xs:restriction>
</xs:simpleType>
```

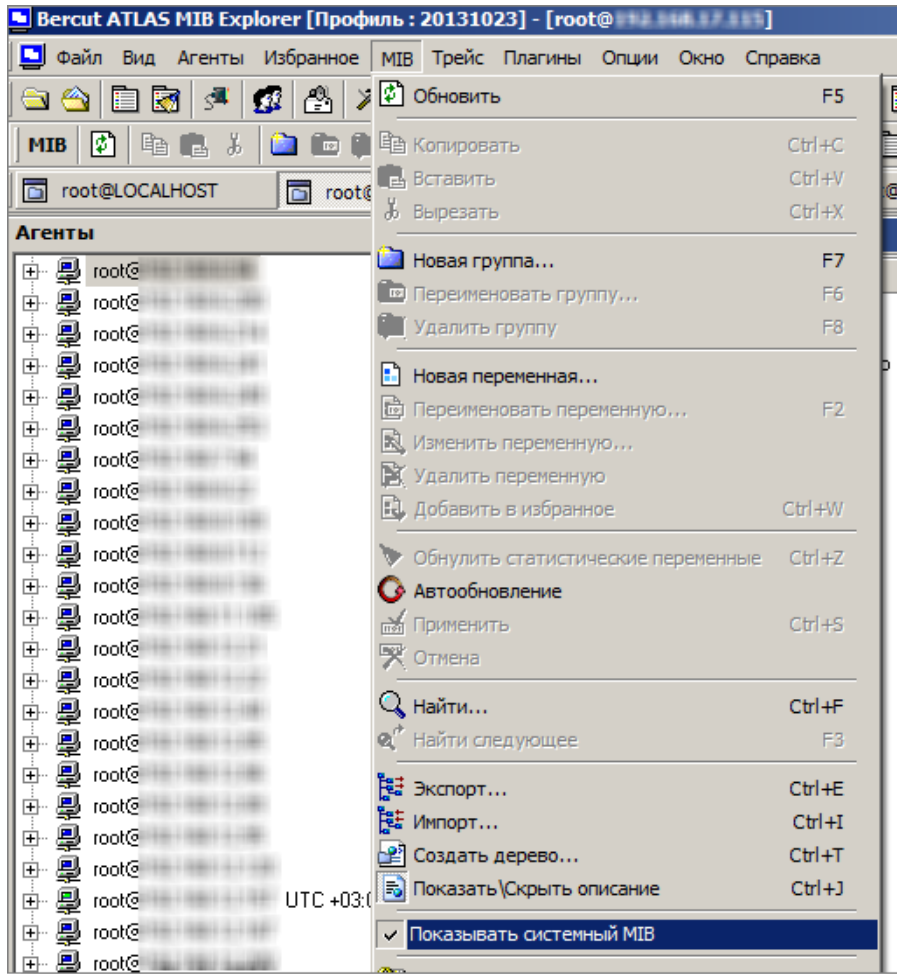
## 8.2. Версионность компонентов в MIB-деревьях

Для поддержки XML-описания артефактов других версий Platform v3 использует миграцию MIB-дерева.

Для работы приложения или артефакта новой версии, у которого уже существует имя в MIB, но осталось XML-описание от другой версии, — используется процедура миграции. Порядок миграции:

- Система копирует существующее MIB-дерево в каталог с названием: <MIB-имя приложения или артефакта>#backup.
- Система создает новое MIB-дерево, в котором:
  - удаляет неиспользуемые параметры;
  - создает новые параметры со значением по умолчанию;
  - переносит существующие параметры с сохранением значений.

**Примечание.** Каталог с названием: <MIB-имя приложения или артефакта>#backup отображается, когда поставлен флаг «Показывать системный MIB» в программе MIB Explorer.



Если компонент Platform v3 имеет развернутые артефакты, при миграции предыдущее имя артефакта добавляется в MIB-переменную *MibName* группы @artifacts/<имя артефакта>/About.

Чтобы удалить созданный резервный файл по окончании миграции, задайте параметр `-Datlas.removeCreatedBackup=true` в скрипт-файле установки для соответствующего компонента (9.1, 10.1 и 12.1).

### 8.3. Общие настройки

В группе @libs/MessageBus/Core/Configuration вы можете задать путь к файлу с лицензией и включить отслеживание входящих сообщений.

Задайте значения следующих параметров группы @libs/MessageBus/Core/Configuration:

- *DumpRequestOnValidationFail*. Флаг отслеживания входящего сообщения при неуспешной проверке активности потока.
- *DeveloperMode*. Флаг включения режима разработчика. Не изменяйте значение параметра без согласования с Bercut.
- *LicenseFilePath*. Путь к файлу с лицензией.

## 8.4. Лицензирование шины RTSIB

*Platform v3 генерирует лицензию на работу компонентов.*

Параметры лицензии:

- Имя приложения.
- Общее количество вызовов в секунду.
- Количество вызовов по определенному интерфейсу или portType.
- Даты начала и окончания действия лицензии.
- MAC-адреса для установки лицензии на аппаратное обеспечение.

Platform v3 проверяет параметры лицензии в следующем порядке:

- при запуске:
  1. имя приложения — элемент <application>;
  2. дату вступления в силу и окончания лицензии — элемент <validity>;
  3. MAC-адрес и IP-адрес.
- во время работы:
  1. количество запросов — элемент <totalInvokes>;
  2. количество запросов по portType — элемент <portTypeInvokes>.

При достижении ограничения система отправляет сообщение LicenseViolation fault.

Platform v3 поддерживает следующие типы лицензирования:

- Без ограничения. Библиотека rtsib-lic-ai.jar.
- С ограничениями, которые задаются в файле лицензии. Библиотека rtsib-lic.jar.

**! Внимание!** В дистрибутив компонента должна входить библиотека только для одного типа лицензирования.

При использовании лицензирования с ограничениями задайте путь к XML-файлу с лицензией в параметре *LicenseFilePath* группы @libs/MessageBus/Core/Configuration. Файл должен иметь цифровую подпись.

Текущий тип лицензирования вы можете посмотреть в переменной *LicenseStatus* группы @libs/MessageBus/Core/Statistics/StateInfo/LicenseInformation.



## 8.5. Настройка портов приложений или взаимодействия между компонентами

*Настройте порты приложений и локальные соединения в MIB-группах SLES.*

После первого запуска компонента выполните настройку портов в следующих MIB-группах:

- @libs/MessageBus/DataServer/ProtocolLayers/Tcp и @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp для **проприетарного протокола**;
- @libs/MessageBus/HttpServer/ProtocolLayers/Tcp для **HTTP-протокола**;
- @libs/MessageBus/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp/ и @libs/LIB\_MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/<MIB-имя компонента> для **взаимодействия между компонентами**;
- @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers для **внешних соединений**.

Если порт, на котором запускается приложение, уже занят, приложение не запускается.

**Локальные соединения** настройте в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<Имя portType>/ScalingTable/local.

**Примечание.** В корневой MIB-группе компонента система отображает все используемые номера портов.



data	String	14812
http	String	14712
service	String	14912
sles mgmt	String	4912

**Рис. 31. Отображение используемых номеров портов**

### 8.5.1. Настройка взаимодействия по проприетарному протоколу

Для успешного взаимодействия по проприетарному протоколу задайте параметры MIB-групп @libs/MessageBus/DataServer/ProtocolLayers/Tcp и @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp.

Для включения или выключения серверов DataServer и ServiceServer используйте MIB-параметр OnOff группы @libs/MessageBus/<DataServer/ServiceServer>/Configuration:

- on — включен;
- off — выключен.



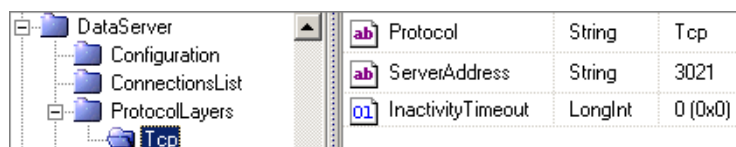
OnOff	String	on
-------	--------	----

**Рис. 32. Включение или выключение серверов**

Порядок настройки серверной части RTSIB:

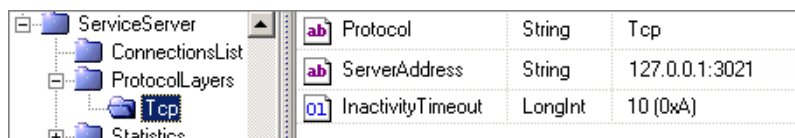
1. Задайте IP-адрес и номер порта точки доступа приема входящих сообщений для передачи данных в MIB-параметре ServerAddress группы @libs/MessageBus/DataServer/ProtocolLayers/Tcp.





**Рис. 33. Группа @libs/MessageBus/DataServer/ProtocolLayers/Tcp**

2. Задайте IP-адрес и номер порта точки доступа приема входящих сообщений для передачи управляющих сообщений в MIB-парамetre *ServerAddress* группы @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp.



**Рис. 34. Группа @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp**

**Примечание.** При изменении IP-адреса измените значения параметров *ServerAddress* для групп @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp, @libs/MessageBus/DataServer/ProtocolLayers/Tcp и @libs/MessageBus/HttpServer/ProtocolLayers/Tcp.

3. Задайте параметры в группах @libs/LIB\_MessageBus/DataServer/ProtocolLayers/Tcp и @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp:

- *AllowedIPs* — список IP-адресов, доступ с которых разрешен. IP-адреса перечисляются через запятую. Можете указать в значении диапазон адресов или маску IP-адреса. Например: 192.168.1.0–192.168.1.55 или 192.168.1.\*.
- *BannedIPs* — список IP-адресов, доступ с которых запрещен. IP-адреса перечисляются через запятую. Можно указать в значении диапазон адресов или маску IP-адреса.
- *InactivityTimeout* — тайм-аут неактивности соединения в минутах. После окончания тайм-аута система разрывает соединение. При значении '0' тайм-аут неограничен.
- *TransportLogLevel* — уровень логирования.
- *AddSourceAddressToContainer* — передача информации о входящем подключении в ATLAS-контейнер.

Точка доступа в MIB-парамetre *ServerAddress* задается в одном из следующих форматов:

- <TcpPort>;
- <IP-address>:<TcpPort>;
- <HostName>:<TcpPort>.

### 8.5.2. Настройка взаимодействия по HTTP-протоколу

Для успешного взаимодействия по HTTP-протоколу задайте параметры MIB-группы @libs/MessageBus/HttpServer/ProtocolLayers/Tcp.

Ограничения при использовании HTTP-протокола:

- Отсутствует стандартное для RTSIB управление потоком.
- Отсутствует стандартное для RTSIB управление доступностью компонентов.

При взаимодействии по HTTP-протоколу используйте следующие HTTP Headers:

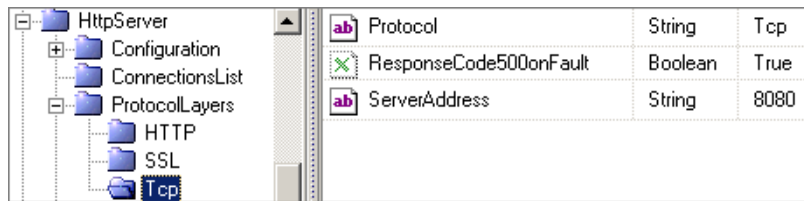
- *Host* — IP-адрес получателя.
- *Content-Type* — укажите значение 'application/fastinfoset' при использовании FastInfoSet, в остальных случаях — 'text/xml'.
- *Connection* — укажите значение 'Keep-Alive' при использовании Persistence-соединения.
- *Content-Length* — размер передаваемых данных пользователя.

Для включения или выключения возможности входящих соединений по HTTP(S)-протоколам используйте MIB-параметр *OnOff* группы @libs/MessageBus/HttpServer/Configuration:

- on — включен;
- off — выключен.

Порядок настройки RTSIB для взаимодействия по HTTP-протоколу:

1. Задайте IP-адрес и номер порта точки доступа в MIB-параметре *ServerAddress* группы @libs/MessageBus/HttpServer/ProtocolLayers/Tcp. При использовании HTTPS-протокола задайте значения переменных группы @libs/MessageBus/HttpServer/ProtocolLayers/SSL.



**Рис. 35. Группа @libs/MessageBus/HttpServer/ProtocolLayers/Tcp**

**Примечание.** При изменении IP-адреса измените значения переменных *ServerAddress* для групп: @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp, @libs/MessageBus/DataServer/ProtocolLayers/Tcp и @libs/MessageBus/HttpServer/ProtocolLayers/Tcp.

2. Задайте URI алиасов в группе @libs/MessageBus/HttpServer/Configuration/Aliases/<название алиаса>.



**Рис. 36. Настройка алиасов**

**Примечание.** Если URI содержит переменные параметры в постоянной части (*UriReplacement*), система записывает этот URI так, как он указан в атрибуте *location* в HTTP Binding. Правила объявления таких псевдонимов:

1. Количество параметров в псевдониме и оригинале должно быть равным.
2. Имена параметров в псевдониме и оригинале должны быть одинаковыми.
3. Порядок параметров в псевдониме может отличаться от порядка в оригинале.

**Пример:**

Для объявления HTTP Binding операции:

```
<definitions targetNamespace="http://www.bercut.com/cc/sa/accountmanagement/http">
  <portType name="AccountManagementHttpGet">
    <binding name="AccountManagementPostPortBinding" type="acwsdl:AccountManagement">
      <http:binding verb="POST"/>
      <operation name="getSubscribers">
        <http:operation location="/subscribers/(msisdn)/search"/>
      </operation>
    </binding>
  </portType>
</definitions>
```

система сформирует следующий оригинальный URI:

```
/cc/sa/accountmanagement/http/AccountManagementHttpGet/subscribers/(msisdn)/search
```

Для этого URI можно объявить псевдоним:

```
/subscribers/(msisdn)/search
```

3. Задайте параметры в группе @libs/MessageBus/HttpServer/ProtocolLayers/Tcp:

- *AllowedIPs* — список IP-адресов, доступ с которых разрешен. IP-адреса перечисляются через запятую. Можете указать в значении диапазон адресов или маску IP-адреса. Например: 192.168.1.0-192.168.1.55 или 192.168.1.\*.
- *BannedIPs* — список IP-адресов, доступ с которых запрещен. IP-адреса перечисляются через запятую. Можно указать в значении диапазон адресов или маску IP-адреса.
- *InactivityTimeout* — тайм-аут неактивности соединения в минутах. После окончания тайм-аута система разрывает соединение. При значении '0' тайм-аут неограничен.
- *TransportLogLevel* — уровень логирования.

При использовании HTTPS-протокола задайте параметры группы @libs/MessageBus/HttpServer/ProtocolLayers/SSL, идентичные параметрам выше. Дополнительно можно задать следующие параметры:

- *AddSourceAddressToContainer* — передача информации о входящем подключении в ATLAS-контейнер.
- *KeyStorePassword* — пароль к хранилищу ключей.
- *KeyStorePath* — путь к хранилищу ключей.

Точка доступа в MIB-параметре *ServerAddress* задается в одном из следующих форматов:

- <TcpPort>;
- <IP-address>:<TcpPort>;
- <HostName>:<TcpPort>.

Чтобы система отправляла в SoapFault код '500', задайте значение True в MIB-параметре *ResponseCode500onFault*. При значении False система отправляет код '200'.

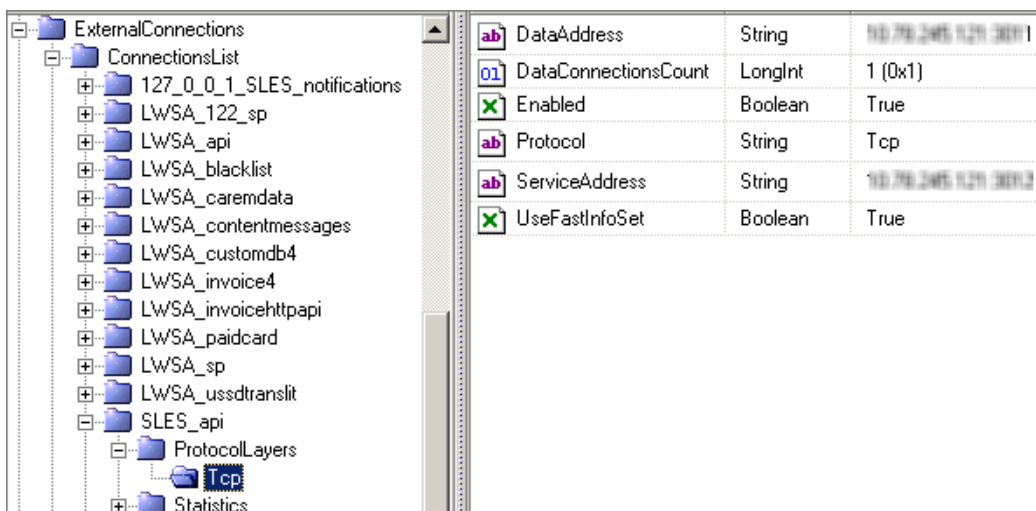
### 8.5.3. Настройка взаимодействия между компонентами

Для настройки взаимодействия с другими компонентами Platform v3 задайте параметры MIB-групп *@libs/MessageBus/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp* и *@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/<MIB-имя компонента>*.

Порядок настройки:

1. Создайте в MIB-группе *@libs/MessageBus/ExternalConnections/ConnectionsList* вложенную группу с MIB-именем компонента.
2. Задайте значения MIB-параметров группы *@libs/MessageBus/ExternalConnections/ConnectionsList/<MIB-имя компонента>*:
  - *DataAddress*. TCP-адрес для передачи данных в формате: IP-address:port.
  - *DataConnectionsCount*. Количество соединений для передачи данных.
  - *Enabled*. Флаг использования подключения:
    - True — подключение используется.
    - False — подключение не используется.
  - *Protocol*. Наименование протокола. Значение: 'Tcp'.
  - *ServiceAddress*. TCP-адрес для передачи управляющих сообщений в формате: IP-address:port.
  - *UseFastInfoSet*. Флаг использования бинарного кодирования (FastInfoSet):
    - True — FastInfoSet используется.
    - False — FastInfoSet не используется.

Значение по умолчанию: True.



**Рис. 37. Группа @libs/MessageBus/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp**

3. Создайте в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable вложенную группу <MIB-имя компонента>.

4. Задайте значения MIB-переменных группы @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/<MIB-имя компонента>:

- *ConnectionName*. Название соединения. Значение: <MIB-имя, указанное в пункте 1>.
- *ConnectionType*. Тип соединения. Значение: 'tcp'.
- *Reserved*. Флаг активности маршрута:
  - True — маршрут активен.
  - False — маршрут находится в резерве.
- *Weight*. Вес маршрута.

ab	ConnectionName	String	SLES_api
ab	ConnectionType	String	tcp
X	Reserved	Boolean	False
01	Weight	LongInt	1 (0x1)

**Рис. 38.** Группа @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/<MIB-имя компонента>

#### 8.5.4. Настройка внешних соединений

При внешнем соединении для настройки RTSIB задайте параметры MIB-группы @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers.

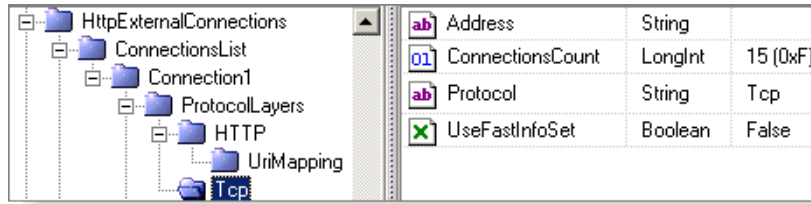
Для взаимодействия с RTSIB используйте следующие внешние соединения:

- По HTTP-протоколу.
- По проприетарному протоколу.

Порядок настройки RTSIB при внешнем соединении по HTTP-протоколу:

1. Задайте значения MIB-переменных группы @libs/MessageBus\_vX.Y/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Tcp:

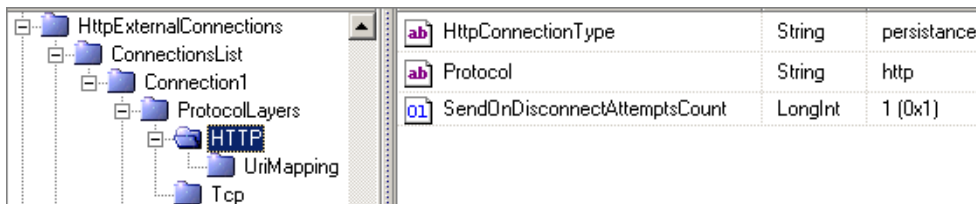
- *Address*. IP-адрес и номер порта физического подключения.
- *ConnectionsCount*. Количество TCP соединений, устанавливаемых с удаленной стороной.
- *UseFastInfoSet*. Флаг включения бинарного кодирования:
  - True — бинарное кодирование используется.
  - False — бинарное кодирование не используется.



**Рис. 39. Группа @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Tcp**

2. Задайте значения МIB-параметров группы @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/HTTP:

- *HttpConnectionType*. Тип подключения:
  - single — соединение под одно сообщение.
  - persistence — постоянное соединение.
  - pipeline — постоянное соединение с механизмом pipeline.
- *SendOnDisconnectAttemptsCount*. Количество попыток отправки сообщения при разрыве соединения до получения ответа. Необязательный параметр.



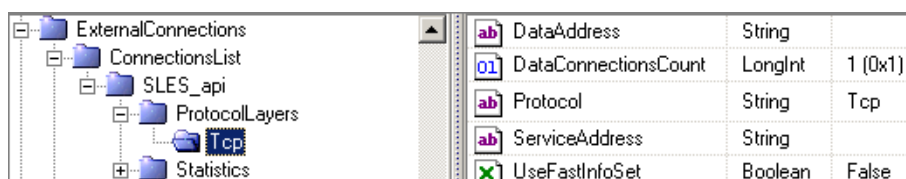
**Рис. 40. Группа @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/HTTP**

3. Задайте таблицу отображения portType на HTTP-URI в группе @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/HTTP/UriMapping. В имени переменной укажите имя portType, в значении — URI.

4. Для настройки RTSIB при внешнем соединении по проприетарному протоколу задайте значения МIB-переменных группы @libs/MessageBus/ExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Tcp:

- *DataAddress*. IP-адрес и номер порта сервера для передачи данных пользователя.
- *ServiceAddress*. IP-адрес и номер порта сервера для передачи служебных данных.
- *DataConnectionsCount*. Количество TCP-соединений, устанавливаемых с удаленной стороной.
- *UseFastInfoSe*. Флаг включения бинарного кодирования сообщений:
  - True — бинарное кодирование используется.
  - False — бинарное кодирование не используется.
- *Enabled*. Флаг использования соединения:
  - True — соединение используется.

- False — соединение не используется.



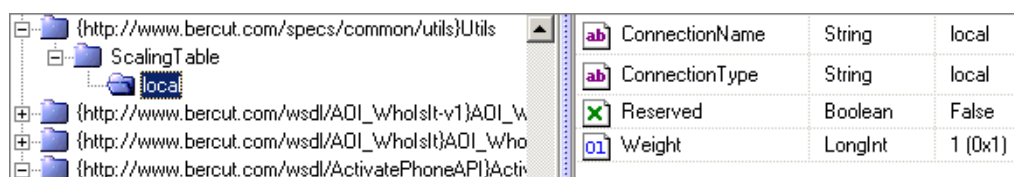
**Рис. 41.** Группа @libs/MessageBus/ExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Тср

### 8.5.5. Настройка локальных соединений

При локальном соединении для настройки RTSIB задайте параметры MIB-группы @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/local.

Порядок настройки:

1. Создайте в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable вложенную группу local.
2. Задайте значения MIB-параметров группы @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/local:
  - *ConnectionName*. Название соединения. Значение: 'local'.
  - *ConnectionType*. Тип соединения. Значение: 'local'.
  - *Reserved*. Флаг активности маршрута:
    - True — маршрут активен.
    - False — маршрут находится в резерве.
  - *Weight*. Вес маршрута.



**Рис. 42.** Группа @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable/local

## 8.6. Настройка маршрутов

RTSIB выполняет маршрутизацию по portType компонента-получателя сообщения. При отсутствии маршрута для portType RTSIB использует маршрут по умолчанию. Настройте маршруты в MIB-группе @libs/MessageBus/Core/Configuration/Routing.

Порядок выбора маршрута:

1. По имени portType получателя и соответствию значений бизнес-параметров.
2. По имени portType получателя.
3. По таблице маршрутизации по умолчанию из MIB-группы @libs/MessageBus/Core/Configuration/Routing/DefaultScaleTable.

При отсутствии маршрута система формирует для сообщения исключение `NoRouteAvailable`.

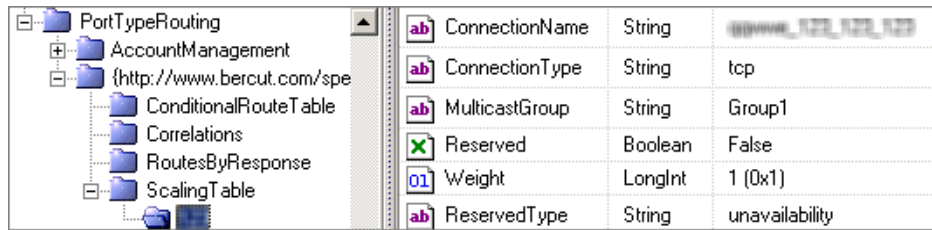
Порядок настройки маршрутизации:

1. Задайте параметры таблицы резервирования и масштабирования в группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType получателя>/ScalingTable/<имя маршрута>:
  - **ConnectionName.** Имя внешнего соединения.
  - **ConnectionType.** Тип используемого внешнего соединения:
    - tcp — внешнее соединение по проприетарному протоколу, MIB-группа @libs/MessageBus/ExternalConnections/ConnectionsList/<имя маршрута>/ProtocolLayers/Tcp.
    - http — внешнее соединение по HTTP-протоколу, MIB-группа @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Tcp.
    - local — локальное соединение по проприетарному протоколу, MIB-группа @libs/MessageBus/LocalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Local.
  - **Reserved.** Признак резервного маршрута:
    - True — маршрут используется при недоступности одного из основных маршрутов.
    - False — резервный маршрут не используется.

При недоступности основного маршрута для балансировки нагрузки система выбирает резервный маршрут с наибольшим весом. При восстановлении доступности основного маршрута система перестает использовать резервный маршрут. Исключение составляют сессии, открытые пока резервный маршрут был активным. Маршрут считается недоступным в следующих ситуациях:

- Для HTTP-соединения с типом Persistence отсутствует хотя бы одно соединение с удаленной стороной.
  - Для проприетарного протокола отсутствует служебное соединение или хотя бы одно соединение для передачи данных.
- **ReservedType.** Тип резервного маршрута:
    - unavailability — включается при недоступности ранее доступного физического соединения по другим маршрутам;
    - overload — включается при получении события HWM на другом маршруте.
  - **Weight.** Вес маршрута. Все исходящие сообщения, кроме продолжающих сессию, система распределяет между элементами таблицы пропорционально весу маршрута.



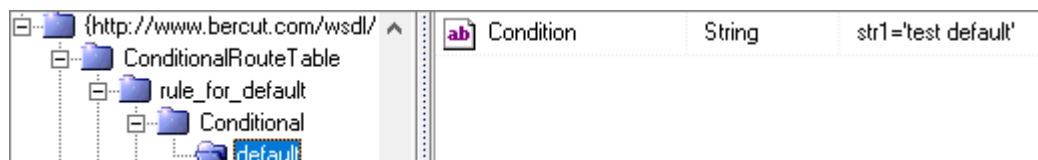


**Рис. 43.** Группа @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType получателя>/ScalingTable/<имя маршрута>

Особенности маршрутизации широковещательных сообщений описаны в разделе 8.9.

2. Задайте параметры маршрутизации по бизнес-параметрам в группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType получателя>/ConditionalRouteTable/<имя условия выбора маршрута>/Conditional/<имя операции>:

**Примечание.** Если требуется настроить одинаковые условия для всех операций, создайте операцию с именем 'default'. В этом случае система проверяет условие для всех операций.



**Рис. 44.** Настройка условия для всех операций

Система выполняет маршрутизацию по группе масштабирования или резервирования при выполнении логических условий. Каждому условию соответствует определенная группа масштабирования или резервирования. Для задания условий используйте синтаксис XPath 1.0:

- Булевы выражения:

- and
- or
- =
- !=
- <
- <=
- >
- >=

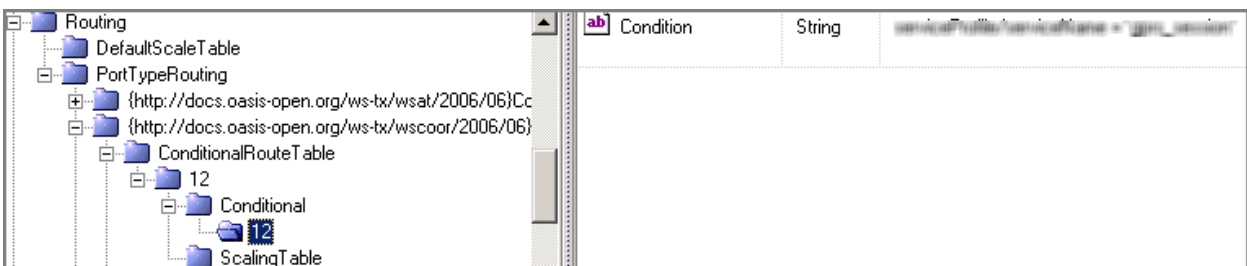
- Булевы функции:

- boolean
- not
- true
- false

- Числовые выражения:

- +

- -
- \*
- mod
- div
- Числовые функции:
  - Number
  - Floor
  - Ceiling
  - Round
- Строковые функции:
  - string
  - concat
  - start-with
  - contains
  - substring-before
  - substring-after
  - substring
  - string-length
  - normalize-space
  - translate
- Специальные числовые значения:
  - NaN
  - +Infinity
  - -Infinity
- Sun XPath Extension:
  - current-date
  - current-time
  - current-dateTime



**Рис. 45. Группа @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType получателя>/ConditionalRouteTable/<имя условия выбора маршрута>/Conditional/<имя операции>**

3. Если для portType нужно подключить все доступные резервные маршруты при выходе из строя всех основных, создайте MIB-параметр *ConnectAllReservedIfAllMainLeft* в группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType получателя>. Задайте значение 'true'

для этого параметра. Если параметр отсутствует или у него значение 'false', система подключает столько резервных маршрутов, сколько вышло из строя основных.

### 8.6.1. Порядок выбора маршрута

Для каждого типа вызова существует свой порядок выбора маршрута.

Типы вызовов при работе с веб-сервисами в рамках RTSIB:

- Транзакционный вызов. Вызов в рамках распределенной транзакции. Транзакция открывается на одном из экземпляров SLES и может быть продолжена на другом экземпляре SLES или на LW SA Container.
- Сессионный вызов. При первом вызове устанавливается сессии «точка-точка». Сессия устанавливается между локальным и удаленным portType и контекстом, который выполняет текущий вызов.
- Широковещательный вызов. Вызов на все доступные маршруты или группы маршрутов.
- Обычный вызов. Простой вызов без признаков транзакционности, сессионности или широкого вещания.

#### Порядок выбора маршрута в рамках одной таблицы маршрутизации

Одна таблица маршрутизации — это настроечный узел ScalingTable. Таблица может использовать безусловную маршрутизацию, маршрутизацию по условию или маршрутизацию по другим принципам.

В каждой таблице маршрутизации вы можете задать любое количество записей маршрутизации — 8.5.3. Система составляет циклическую цепочку маршрутов в соответствии с их весами.

#### Пример.

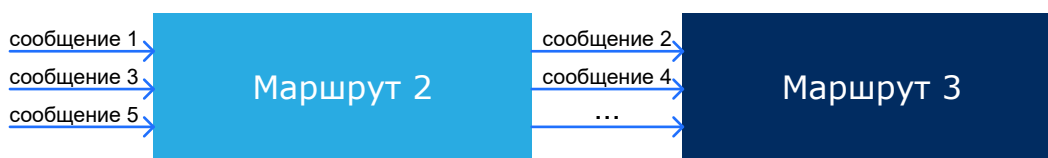
В системе есть три маршрута: *Маршрут 1* с весом 2, *Маршрут 2* с весом 1 и резервный маршрут *Маршрут 3* с весом 1.

Если все маршруты доступны, система составляет следующую цепочку для отправки сообщений:



**Рис. 46. Цепочка маршрутов при доступности всех маршрутов**

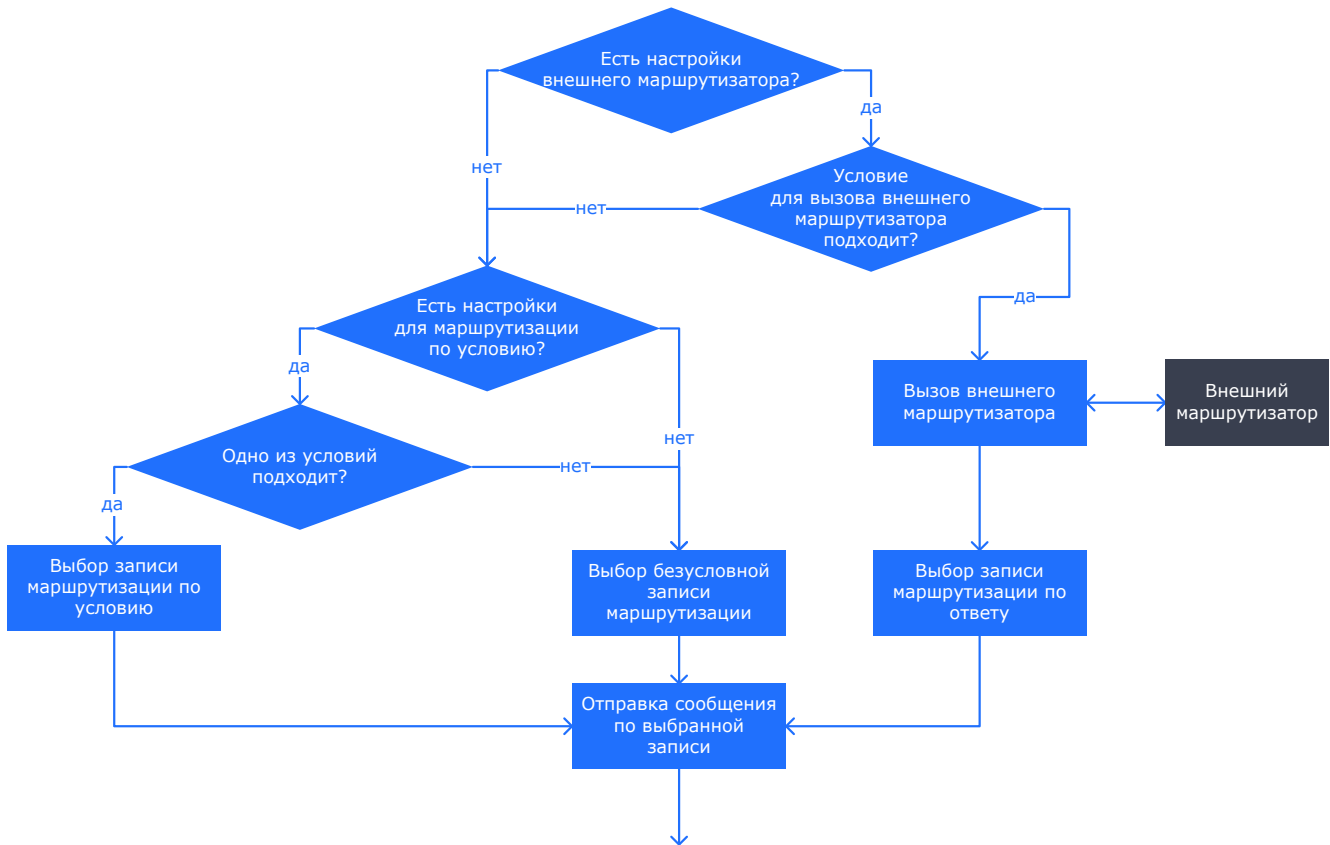
Когда один из маршрутов становится недоступным, система подключает резервный маршрут *Маршрут 3*. При недоступности *Маршрута 1* цепочка будет выглядеть так:



**Рис. 47. Цепочка маршрутов при недоступности одного маршрута**

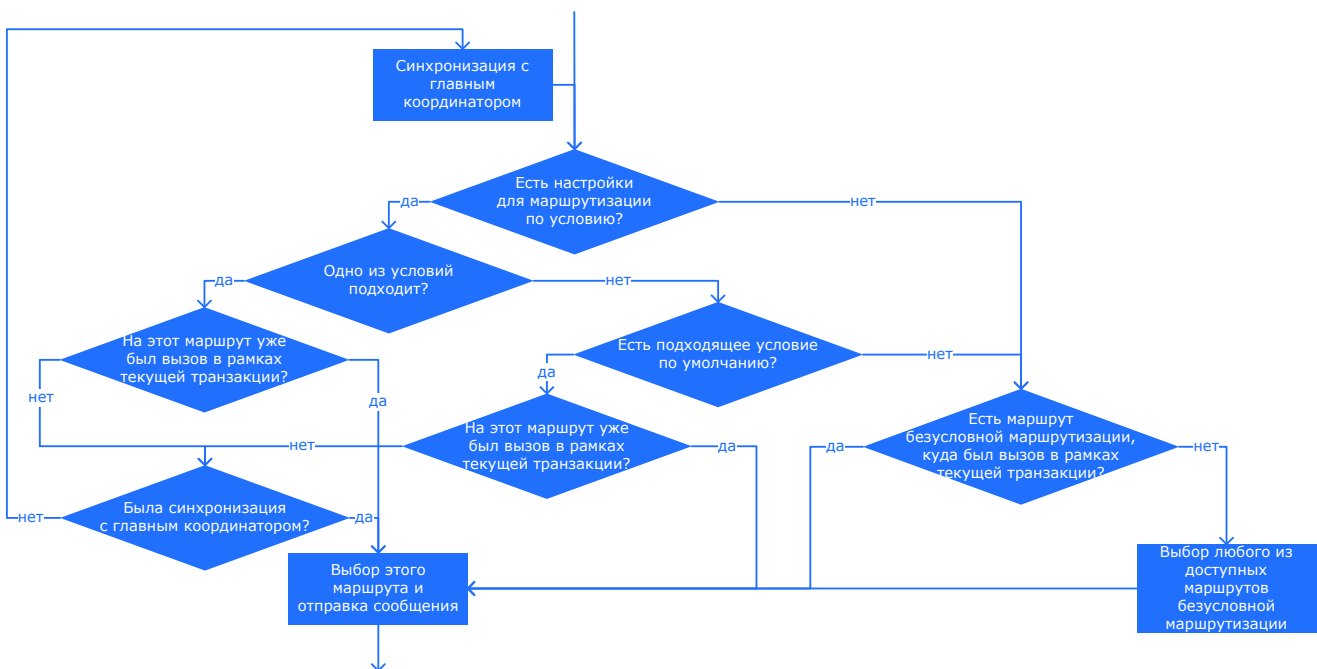
После восстановления *Маршрута 1* цепочка принимает первоначальный вид.

**Порядок выбора маршрута для обычных вызовов**



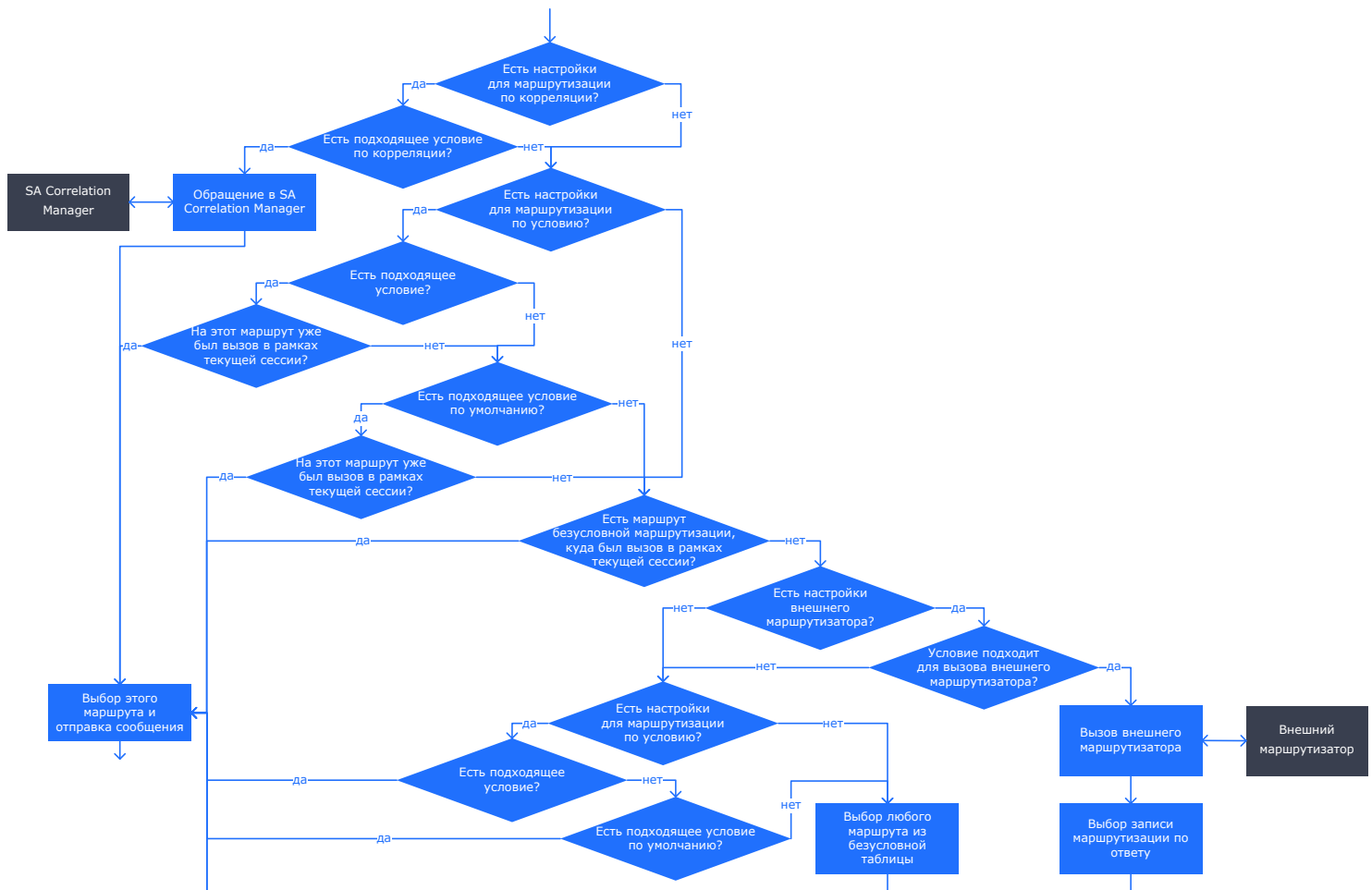
**Рис. 48. Определение маршрута для обычных вызовов**

**Порядок выбора маршрута для транзакционных вызовов**



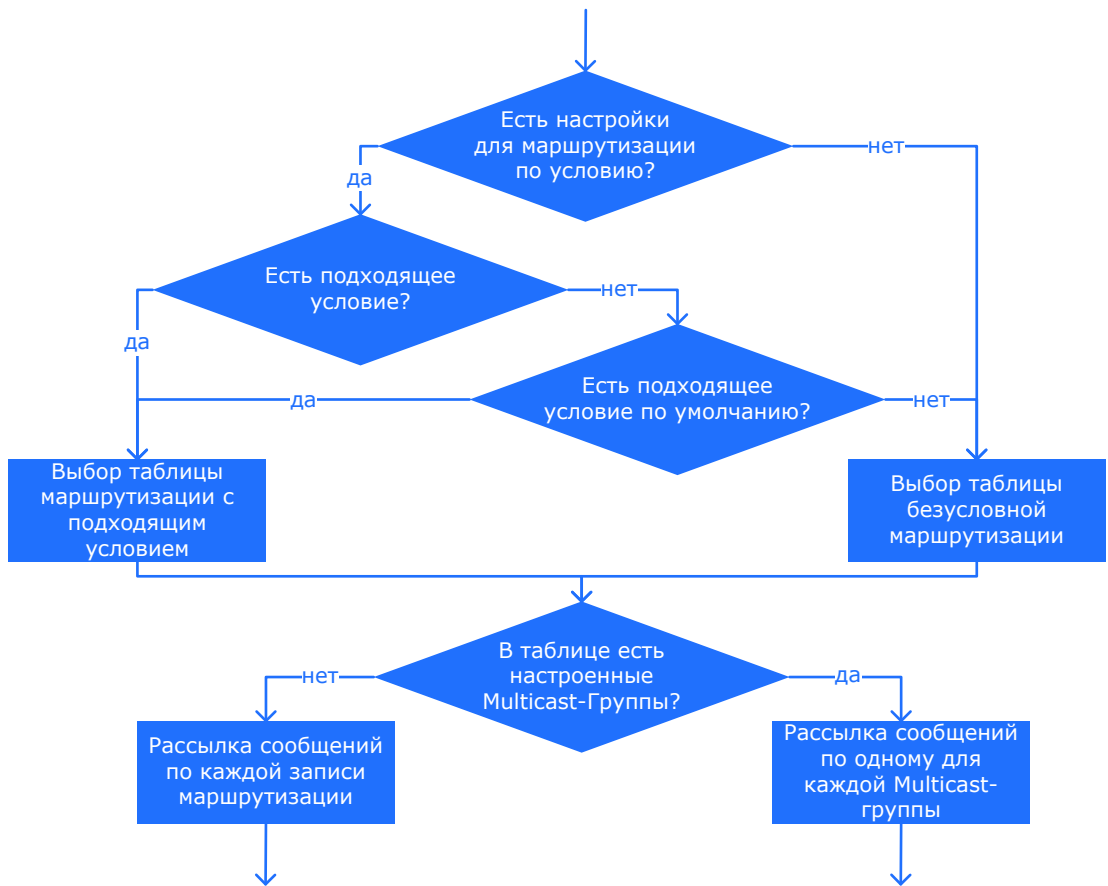
**Рис. 49. Определение маршрута для транзакционных вызовов**

### Порядок выбора маршрута для сессионных вызовов



**Рис. 50. Определение маршрута для сессионных вызовов**

### Порядок выбора маршрута для широковещательных вызовов



**Рис. 51. Определение маршрута для широковещательных вызовов**

## 8.7. Настройка условий маршрутизации

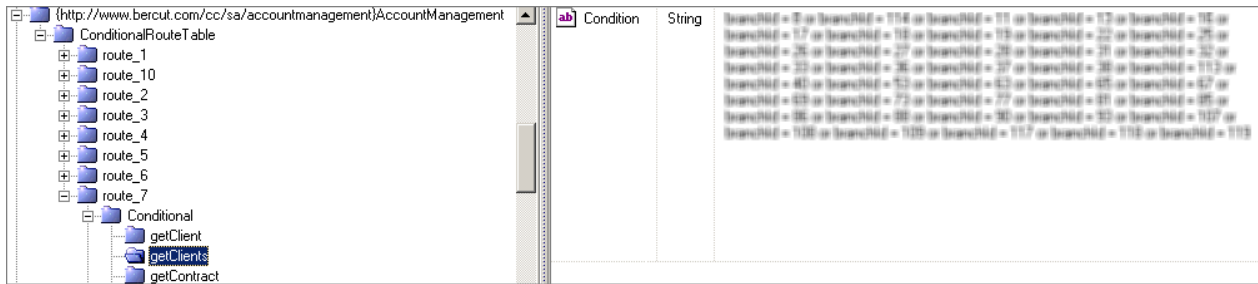
*RTSIB* позволяет задать условие для конкретного *portType*.

Выполните следующие действия:

1. Создайте в группе `CMP_SLES_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>` вложенную группу `ConditionalRouteTable`.
2. В созданную группу добавьте группу с именем маршрута.
3. Создайте в группе `CMP_SLES_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ConditionalRouteTable/<имя маршрута>` вложенную группу с названием условия.
4. Задайте условие в переменной *Condition* в формате:

```

.<operation> : <condition>
...
<operation> : <condition>
.* : <condition>
    
```

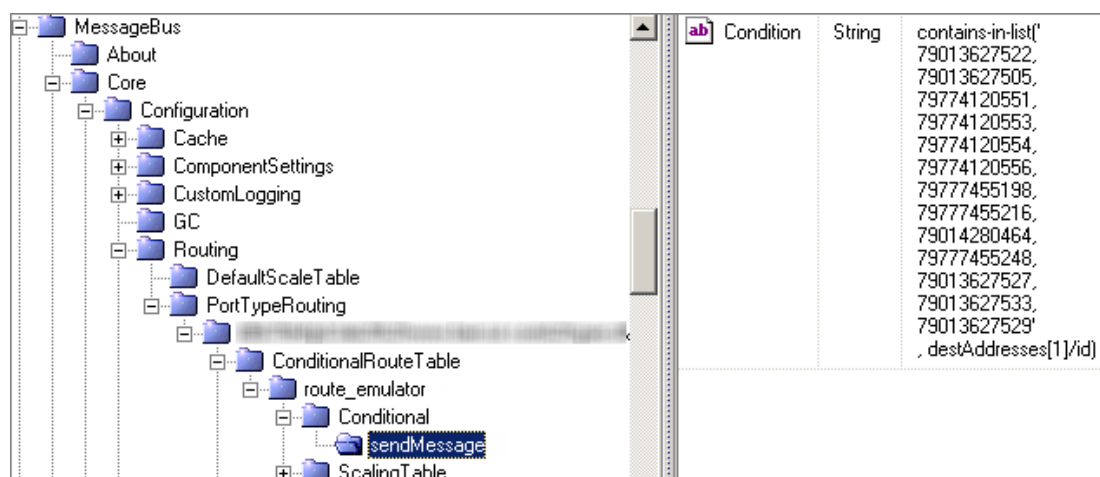


**Рис. 52. Условие маршрутизации для portType**

**Примечание.** Вы можете создать не более 48 условий. Условия могут быть объединены.

### Пример использования функции contains-in-list для операции sendMessage

```
contains-in-list('
79013627522,
79013627505,
79774120551,
79774120553,
79774120554,
79774120556,
79777455198,
79777455216,
79014280464,
79777455248,
79013627527,
79013627533,
79013627529'
, destAddresses[1]/id)
```



**Рис. 53. Пример условия для операции sendMessage**

## 8.8. Маршрутизация по бизнес-параметрам ответа из внешней системы

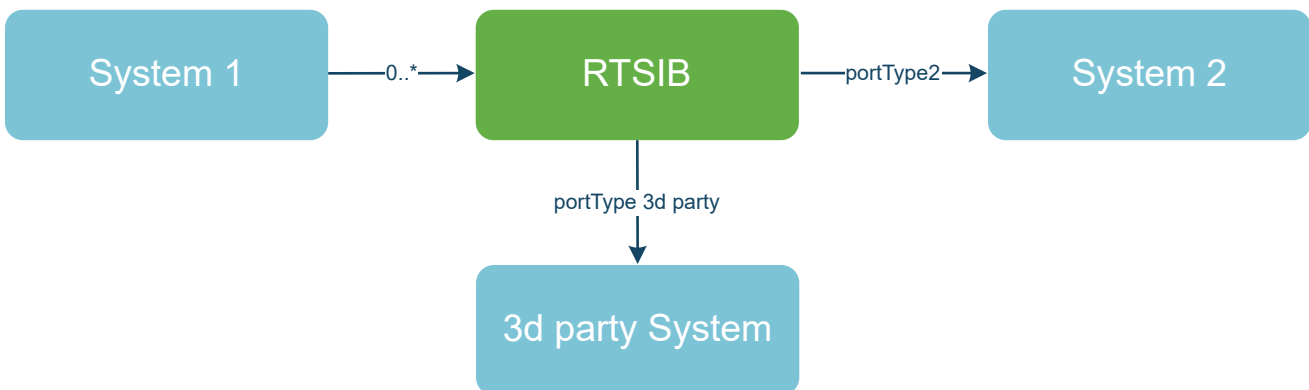
*RTSIB* позволяет маршрутизировать сообщение по бизнес-параметрам ответа из внешней системы.

Алгоритм маршрутизации по бизнес-параметрам ответа из внешней системы:

1. Система 1 вызывает систему 2 по portType 2 с помощью клиентского интерфейса.
2. Маршрутизация по ответу включается, если соблюдены следующие условия:
  - исходящее SOAP-сообщение не продолжает сессию.
  - соблюдено условие XPath-выражения с результатом булева типа. XPath-выражение система вычисляет по XML-модели исходящего SOAP-сообщения на операцию из portType 2.

Если условия не соблюдены, система маршрутизирует сообщение по стандартным правилам RTSIB.

3. Система выполняет маппинг бизнес-параметров исходящего сообщения в параметры сообщения к третьей системе с portType 3d party. Система вызывает portType 3d party.
4. Система последовательно проверяет условия XPath-выражений по ответному SOAP-сообщению от portType 3d party:
  - Если условие подходит, система выбирает маршрут этого условия. Исходное сообщение portType 2 система отправляет по выбранному маршруту.
  - Если условие не подходит, система проверяет следующее условие.



**Рис. 54. Маршрутизация по бизнес-параметрам**

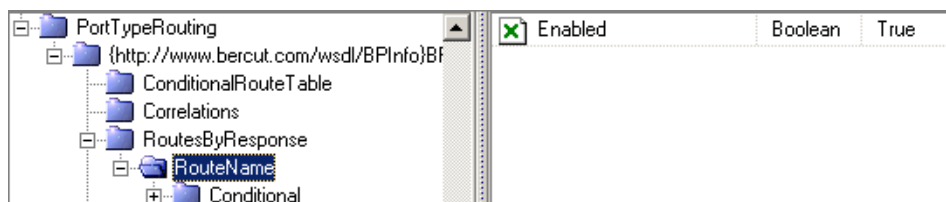
### 8.8.1. Настройка маршрутизации по бизнес-параметрам ответа внешней системы

Параметры маршрутизации по бизнес-параметрам ответа внешней системы хранятся в MIB-группе `CMP_SLES_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse`.

Выполните следующие действия:

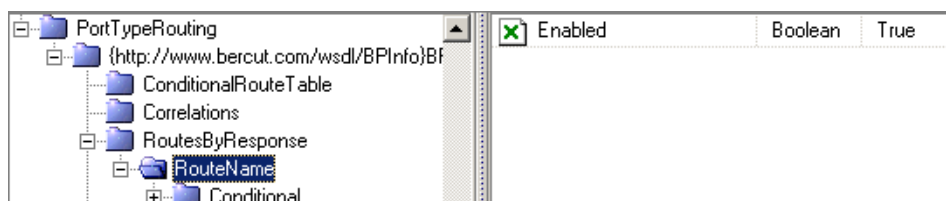


1. Создайте в группе CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse вложенную группу с именем маршрута.
2. Включите маршрут. Для этого поставьте флаг *Enabled*.



**Рис. 55. Включение маршрута**

3. Создайте в группе CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional вложенную группу с названием условия.
4. Задайте следующие параметры:
  - *EnablingCondition*. XPath-выражение, которое задает условие выбора маршрута.
  - *ExternalInterfaceVersion*. Версия интерфейса третьей стороны.
  - *ExternalOperation*. Имя операции для portType третьей стороны (portType 3d party).
  - *ExternalPortType* имя portType третьей стороны (portType 3d party).



**Рис. 56. Параметры условия**

5. Создайте в группе CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional/<название условия>/ConditionalRoutes/ вложенную группу с названием маршрута.
6. Задайте следующие параметры:
  - *ExternalResponseCondition*. XPath-выражение, которое определяет условие выбора маршрута по ответу от третьей стороны.
  - *RequestCondition*. XPath-выражение, которое определяет условие выбора маршрута по начальному запросу маршрута.



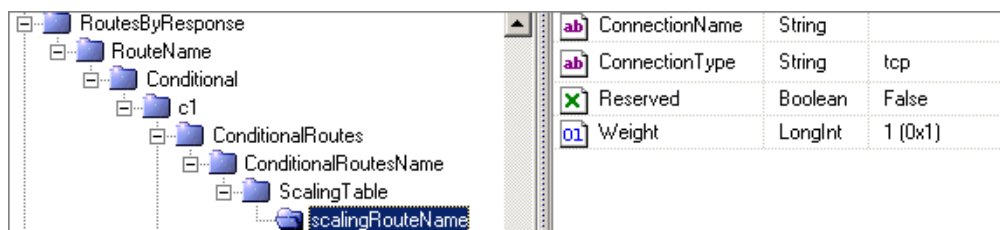
**Рис. 57. Параметры маршрута условия**

7. Задайте параметры таблицы резервирования и масштабирования в группе CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional/<название условия>/ConditionalRoutes/<имя маршрута условия>/ScalingTable/<имя маршрута>:

- **ConnectionName.** Имя внешнего соединения.
- **ConnectionType.** Тип транспорта внешнего соединения:
  - tcp — внешнее соединение по проприетарному протоколу, MIB-группа @libs/MessageBus/ExternalConnections/ConnectionsList/<имя маршрута>/ProtocolLayers/Tcp.
  - http — внешнее соединение по HTTP-протоколу, MIB-группа @libs/MessageBus/HttpExternalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Tcp.
  - local — локальное соединение по проприетарному протоколу, MIB-группа @libs/MessageBus/LocalConnections/ConnectionsList/<Имя соединения>/ProtocolLayers/Local.
- **Reserved.** Признак резервного маршрута:
  - True — маршрут используется при недоступности одного из основных маршрутов.
  - False — резервный маршрут не используется.

При недоступности основного маршрута для балансировки нагрузки система выбирает резервный маршрут с наибольшим весом. При восстановлении доступности основного маршрута система перестает использовать резервный маршрут. Исключение составляют сессии, открытые пока резервный маршрут был активным. Маршрут считается недоступным в следующих ситуациях:

- Для HTTP-соединения с типом Persistence отсутствует хотя бы одно соединение с удаленной стороной.
- Для проприетарного протокола отсутствует служебное соединение или хотя бы одно соединение для передачи данных.
- **Weight.** Вес маршрута. Все исходящие сообщения, кроме продолжающих сессию, система распределяет между элементами таблицы пропорционально весу маршрута.

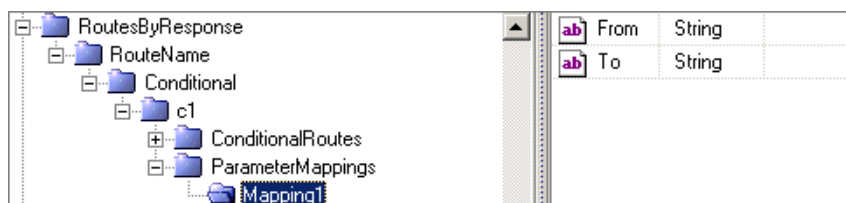


ab	ConnectionName	String	
ab	ConnectionType	String	tcp
X	Reserved	Boolean	False
01	Weight	LongInt	1 (0x1)

**Рис. 58. Параметры таблицы резервирования и масштабирования**

8. Задайте параметры мappинга к третьей стороне в группе CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional/<название условия>/ParameterMappings/<название мappинга>:

- *From*. XPath-выражение по исходному запросу, из которого будет скопировано значение.
- *To*. XPath-путь к элементу, в который будет скопировано значение из параметра *From*.



ab	From	String	
ab	To	String	

**Рис. 59. Параметры мappинга**

Порядок применения параметров:

1. Поиск условия активизации маршрутизации по ответу — параметр *EnablingCondition*.
2. Создание запроса к третьей стороне по portType *ExternalPortType* и операции *ExternalOperation*.
3. Заполнение запроса по всем строкам таблицы CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional/<название условия>/ParameterMappings. Отправка запроса по стандартным правилам RTSIB.
4. Получение ответа от *ExternalPortType*.
5. Последовательное вычисление XPath-условий по ответу третьей стороны из группы CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional/<название условия>/ConditionalRoutes:
  - Если подходящее условие найдено, отправка исходного сообщения по маршруту CMP\_SLES\_vX.Y/@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<название portType>/RoutesByResponse/<имя маршрута>/Conditional/<название условия>/ConditionalRoutes/<имя маршрута условия>/ScalingTable.
  - Если подходящее условие не найдено, отправка сообщения об ошибке 'RouteUnavailable'.

Синтаксис адресации к элементам сообщений в XPath-выражениях описан в разделе [8.8.2](#).

## 8.8.2. Синтаксис адресации к элементам сообщений в XPath-выражениях

*XPath-выражения используются в MIB-настройках RTSIB для описания условий или получения и установки значений в элементах сообщений.*

Для адресации элемента указанием 'part' Platform v3 использует синтаксис с фиктивной переменной \$message:

```
$message.[PART_NAME]/[ELEMENT_NAME]
```

где:

PART\_NAME — имя part.

ELEMENT\_NAME — имя элемента.

При подмене выражения с подстановкой имени part Platform v3 использует следующий синтаксис:

- Выражение `$input/[ELEMENT_NAME]` аналогично записи `$message.[BODY_PART_NAME]/[ELEMENT_NAME]`.
- Выражение `/[ELEMENT_NAME]` аналогично записи `$message.[BODY_PART_NAME]/[ELEMENT_NAME]`.
- Выражение `[ELEMENT_NAME]` аналогично записи `$message.[BODY_PART_NAME]/[ELEMENT_NAME]`.

где: BODY\_PART\_NAME — имя part типа body в сообщении.

Допускается использовать в выражении один вид синтаксиса: или \$message, или \$input, или без фиктивной переменной.

### Примеры:

```
<xsd:complexType name="ManyPartHeaderType">
  <xsd:sequence>
    <xsd:element name="string" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="ManyPartRequestType">
  <xsd:sequence>
    <xsd:element name="list" type="xsd:string"/>
    <xsd:element name="string" type="xsd:string"/>
    <xsd:element name="delimiter" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:element name="ManyPartRequest" type="tns:ManyPartRequestType"/>
<xsd:element name="ManyPartHeader" type="tns:ManyPartHeaderType"/>
```

```
<message name="ManyPartRequestMessage">
  <part name="header" element="aspm:ManyPartHeader"/>
  <part name="parameters" element="aspm:ManyPartRequest"/>
</message>
```

### Адресация элемента string в заголовке:

```
$message.header/string
```

Адресация элемента list в body:

```
$message.parameters/list
```

```
$input/list
```

```
/list
```

```
list
```

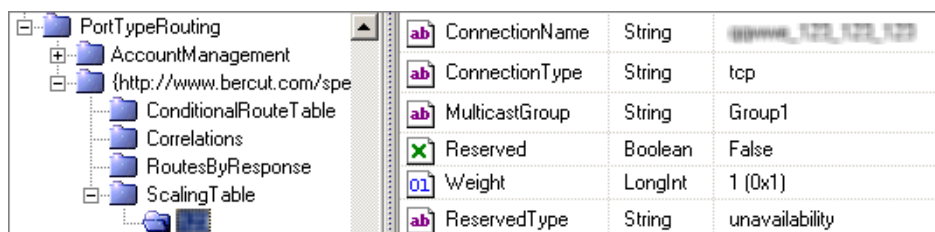
## 8.9. Обработка широковещательных сообщений

В группе `@libs/MessageBus/Core/Configuration` вы можете задать путь к файлу с лицензией и включить отслеживание входящих сообщений.

Задайте признак широковещательных сообщений в теге `PartnerLink` в WSDL-описании.

Широковещательные сообщения должны быть однонаправленными (OneWay).

Система отправляет сообщение каждой широковещательной группе. Имя широковещательной группы задайте в переменной `MulticastGroup`.



**Рис. 60.** Группа `@libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType получателя>/ScalingTable/<имя маршрута>`

Если имя группы не указано, система присваивает маршруту уникальное имя. Доставка сообщений по маршруту выполняется так, как если бы маршрут был в группе один.

Внутри каждой группы система доставляет сообщения стандартным способом — round robin.

## 8.10. Настройка тайм-аутов

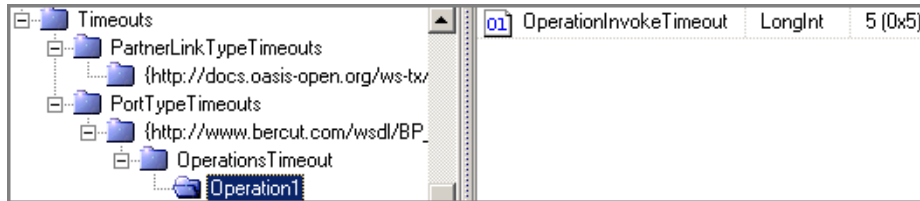
### 8.10.1. Настройка тайм-аутов внешних вызовов

Для любого внешнего вызова на RTSIB задайте тайм-аут. После окончания тайм-аута при отсутствии ответа на вызов система генерирует клиенту событие `onTimeout`.

При внешнем вызове система выбирает тайм-аут по следующему приоритету:

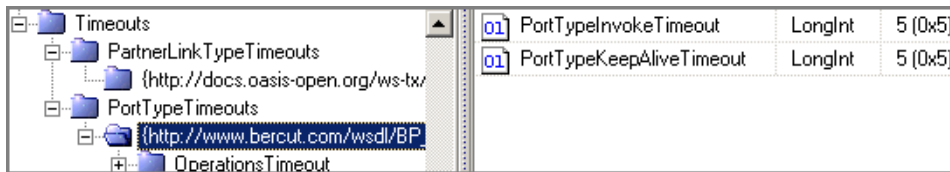
1. Тайм-аут на конкретную операцию для portType.
2. Тайм-аут для portType.
3. Тайм-аут по умолчанию.

1. Для настройки тайм-аута на конкретную операцию для portType используйте MIB-параметр *OperationInvokeTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>/OperationsTimeout/<Имя операции>. Тайм-аут задается в секундах.



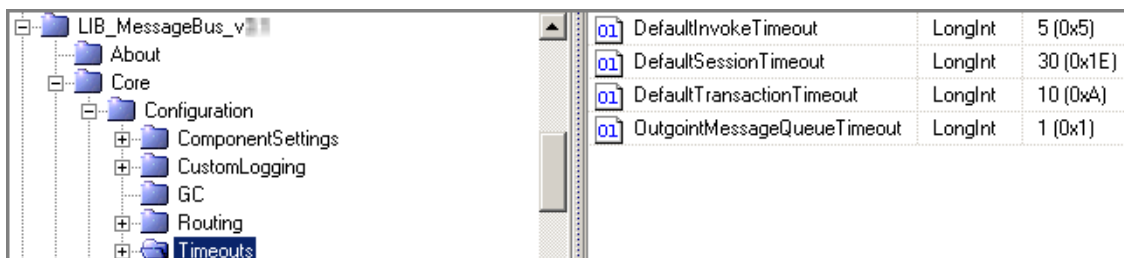
**Рис. 61. Группа @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>/OperationsTimeout/<Имя операции>**

2. Для настройки тайм-аута на все операции для конкретного portType используйте MIB-параметр *PortTypeInvokeTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>. Тайм-аут задается в секундах.



**Рис. 62. Группа @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>**

3. Для настройки тайм-аута по умолчанию используйте MIB-параметр *DefaultInvokeTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts. Тайм-аут задается в секундах.



**Рис. 63. Группа @libs/MessageBus/Core/Configuration/Timeouts**

4. Для настройки тайм-аута нахождения пакета в очереди на отправку используйте MIB-параметр *OutgoingMessageQueueTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts. Тайм-аут задается в секундах.

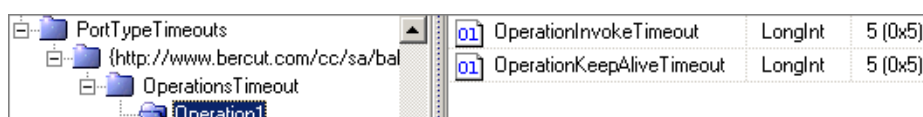
## 8.10.2. Настройка тайм-аутов активности реализации

Для проверки работоспособности сервиса диспетчер потоков анализирует тайм-аут неактивности потока. После окончания тайм-аута при отсутствии активности система генерирует аларм.

При обработке входящего вызова система выбирает тайм-аут активности по следующему приоритету:

1. Тайм-аут на конкретную операцию для portType.
2. Тайм-аут для portType.
3. Тайм-аут по умолчанию.

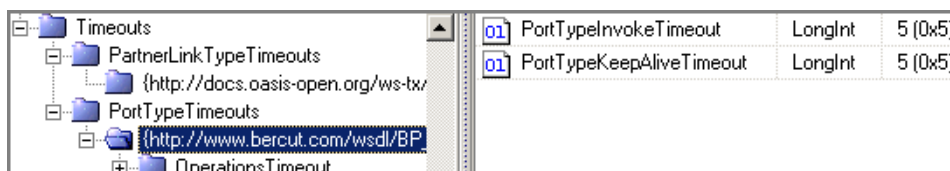
1. Для настройки тайм-аута активности на конкретную операцию для portType используйте MIB-параметр *OperationKeepAliveTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>/OperationsTimeout/<Имя операции>. Тайм-аут задается в секундах.



PortTypeTimeouts	01	OperationInvokeTimeout	LongInt	5 (0x5)
{http://www.bercut.com/cc/sa/bal}	01	OperationKeepAliveTimeout	LongInt	5 (0x5)
OperationsTimeout				
Operation1				

**Рис. 64.** Группа @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>/OperationsTimeout/<Имя операции>

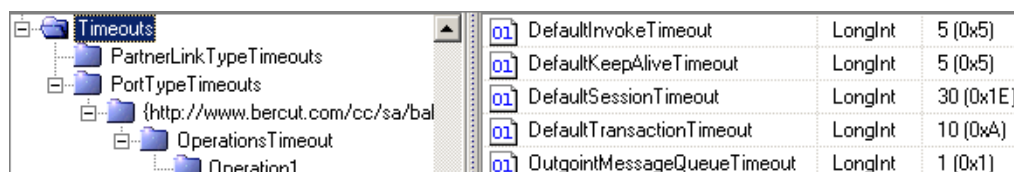
2. Для настройки тайм-аута активности на все операции для конкретного portType используйте MIB-параметр *PortTypeKeepAliveTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>. Тайм-аут задается в секундах.



Timeouts	01	PortTypeInvokeTimeout	LongInt	5 (0x5)
PartnerLinkTypeTimeouts				
{http://docs.oasis-open.org/ws-tx}				
PortTypeTimeouts	01	PortTypeKeepAliveTimeout	LongInt	5 (0x5)
{http://www.bercut.com/wsd/BP}				
OperationsTimeout				

**Рис. 65.** Группа @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>

3. Для настройки тайм-аута активности по умолчанию используйте MIB-переменную *DefaultKeepAliveTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts. Тайм-аут задается в секундах.



Timeouts	01	DefaultInvokeTimeout	LongInt	5 (0x5)
PartnerLinkTypeTimeouts	01	DefaultKeepAliveTimeout	LongInt	5 (0x5)
PortTypeTimeouts	01	DefaultSessionTimeout	LongInt	30 (0x1E)
{http://www.bercut.com/cc/sa/bal}	01	DefaultTransactionTimeout	LongInt	10 (0xA)
OperationsTimeout	01	OutgoingMessageQueueTimeout	LongInt	1 (0x1)
Operation1				

**Рис. 66.** Группа @libs/MessageBus/Core/Configuration/Timeouts

### 8.10.3. Настройка тайм-аутов ожидания ответа

Для проверки реакции сервиса в системе реализован тайм-аут ожидания ответа.

После окончания тайм-аута при отсутствии ответа система увеличивает значение статистической переменной по количеству ответов, обработка которых превысила указанное значение.

При обработке входящего вызова система выбирает тайм-аут ожидания ответа по следующему приоритету:

1. Тайм-аут на конкретную операцию для portType.
2. Тайм-аут для portType.
3. Тайм-аут по умолчанию.

1. Чтобы задать тайм-аут попадания статистики в OverQoSResponses или OverQoSReplys для операции portType используйте MIB-параметр *OperationExpectedQoSTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>/OperationsTimeout/<Имя операции>. Тайм-аут задается в миллисекундах.

01	OperationExpectedQoSTimeout	LongInt	2000 (0x7D0)
01	OperationInvokeTimeout	LongInt	5 (0x5)
01	OperationKeepAliveTimeout	LongInt	5 (0x5)

**Рис. 67. Группа @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>/OperationsTimeout/<Имя операции>**

2. Чтобы задать тайм-аут попадания статистики в OverQoSResponses или OverQoSReplys для всех операций portType используйте MIB-параметр *PortTypeExpectedQoSTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>. Тайм-аут задается в миллисекундах.

01	PortTypeExpectedQoSTimeout	LongInt	2000 (0x7D0)
01	PortTypeInvokeTimeout	LongInt	5 (0x5)
01	PortTypeKeepAliveTimeout	LongInt	5 (0x5)

**Рис. 68. Группа @libs/MessageBus/Core/Configuration/Timeouts/PortTypeTimeouts/<Имя portType>**

3. Для настройки тайм-аута ожидания ответа по умолчанию используйте MIB-переменную *DefaultKeepAliveTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts. Тайм-аут задается в миллисекундах.

01	DefaultExpectedQoSTimeout	LongInt	2000 (0x7D0)
01	DefaultInvokeTimeout	LongInt	5 (0x5)
01	DefaultKeepAliveTimeout	LongInt	5 (0x5)
01	DefaultSessionTimeout	LongInt	30 (0x1E)
01	DefaultTransactionTimeout	LongInt	10 (0xA)
01	OutgoingMessageQueueTimeout	LongInt	1 (0x1)

**Рис. 69. Группа @libs/MessageBus/Core/Configuration/Timeouts**



## 8.11. Настройка адаптивных очередей

Адаптивный механизм управления потоком служит для оптимизации работы в условиях переменной производительности.

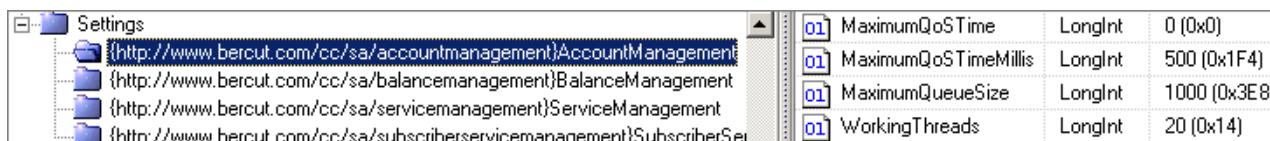
Инструментом управления является динамическое изменение в принимающей очереди сервера уровней HWM/LWM. Смещение уровней выполняется на основании анализа времени нахождения сообщений в очереди. При увеличении времени нахождения сообщений в очереди уровни HWM/LWM снижаются, вследствие чего размер очереди уменьшается. При уменьшении времени нахождения сообщений в очереди уровни повышаются, вследствие чего размер очереди увеличивается. Ограничением использования памяти служит максимальный размер очереди.

**! Внимание!** Адаптивные очереди использует только LW SA Container.

Настройки адаптивных очередей расположены в группе @libs/MessageBus/Core/Configuration/ComponentSettings/Settings/<Имя реализуемого portType>:

- *WorkingThreads*. Количество потоков, используемых для обработки входящей очереди компонента.
- *MaximumQueueSize*. Максимальный размер адаптивной очереди.
- *MaximumQoSTime*. Максимальное время нахождения сообщения во входящей адаптивной очереди, в секундах.
- *MaximumQoSTimeMillis*. Максимальное время нахождения сообщения во входящей адаптивной очереди, в миллисекундах.

**i Примечание.** Итоговое максимальное время рассчитывается как сумма значений переменных *MaximumQoSTime* и *MaximumQoSTimeMillis*. Если *MaximumQoSTime*=0 и *MaximumQoSTimeMillis* меньше 10 миллисекунд, итоговое время устанавливается равным 10 миллисекундам.



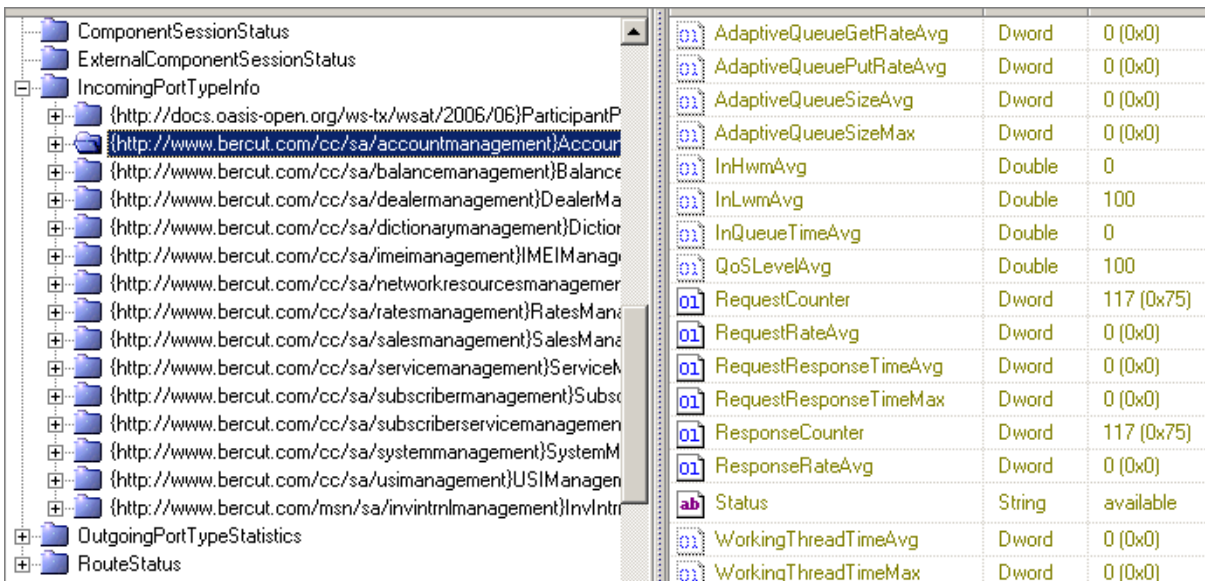
01	MaximumQoSTime	LongInt	0 (0x0)
01	MaximumQoSTimeMillis	LongInt	500 (0x1F4)
01	MaximumQueueSize	LongInt	1000 (0x3E8)
01	WorkingThreads	LongInt	20 (0x14)

**Рис. 70.** Группа @libs/MessageBus/Core/Configuration/ComponentSettings/Settings/<Имя реализуемого portType>

Статистика по каждому portType хранится в переменных группы @libs/MessageBus/Core/Statistics/StateInfo/IncomingPortTypeInfo/<Имя PortType>:

- *AdaptiveQueueSizeAvg*. Усредненный размер адаптивной очереди. Только для LW SA Container.
- *AdaptiveQueueSizeMax*. Максимальный размер адаптивной очереди. Только для LW SA Container.
- *AdaptiveQueueGetRateAvg*. Среднее количество взятых в обработку сообщений в адаптивной очереди. Только для LW SA Container.
- *AdaptiveQueuePutRateAvg*. Среднее количество добавленных в адаптивную очередь сообщений. Только для LW SA Container.
- *InHwmAvg*. Среднее время нахождения компонента в статусе Hwm, в процентах.
- *InLwmAvg*. Среднее время нахождения компонента в статусе Lwm, в процентах.
- *InQueueTimeAvg*. Среднее время нахождения сообщений для компонента в адаптивной очереди, в миллисекундах.
- *QoSLevelAvg*. Среднее значение границы QoS-уровня.
- *RequestCounter*. Общее количество входящих запросов.

- *RequestRateAvg*. Среднее количество запросов в секунду.
- *RequestResponseTimeAvg*. Среднее время ответа на входящий запрос, в миллисекундах.
- *RequestResponseTimeMax*. Максимальное время ответа на входящий запрос, в миллисекундах.
- *ResponseCounter*. Общее количество отправленных ответов.
- *ResponseRateAvg*. Среднее количество ответов в секунду.
- *CantSendReplyAvg*. Среднее количество неотправленных ответов в секунду.
- *CantSendReplyCounter*. Общее количество неотправленных ответов.
- *WorkingThreadTimeAvg*. Среднее время обработки запроса логикой сервиса, в миллисекундах.
- *WorkingThreadTimeMax*. Максимальное время обработки запроса логикой сервиса, в миллисекундах.
- *Status*. Статус доступности компонента.



**Рис. 71. Группа @libs/MessageBus/Core/Statistics/StateInfo/IncomingPortTypeInfo/<Имя PortType>**

## 8.12. Настройка транзакционности

Настройте параметры транзакционности при создании и изменении бизнес-процессов, а также в MIB.

Для транзакционных процессов установите значение **true** атрибута *transaction*. Атрибут расположен в теге `<plink:partnerLinkType>` WSDL-описания интерфейса. Пример:

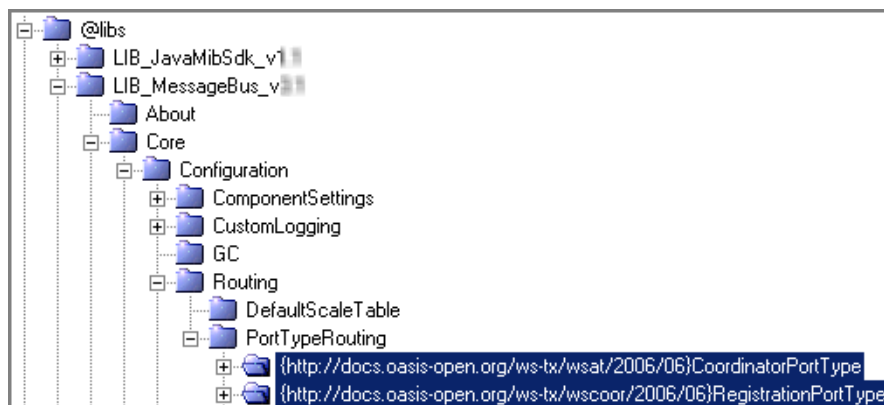
```
<plink:partnerLinkType xmlns:bxw="http://bercut.com/wsd/Extensions"
name="ServiceProfileManagement" bxw:transaction="true">
  <plink:role name="ServiceProfilePortTypeRole" portType="tns:ServiceProfilePortType"/>
</plink:partnerLinkType>
```

При первом вызове в рамках транзакции компонент регистрируется в координаторе транзакций. В качестве зарегистрированных компонентов в координатор передаются

все portType, зарегистрированные на текущем экземпляре RTSIB. Все portType автоматически попадают в список маршрутизации вызывающей стороны.

Задайте параметры маршрутизации для транзакционных процессов на RTSIB в группах:

- {http://docs.oasis-open.org/ws-tx/wsat/2006/06}CoordinatorPortType.
- {http://docs.oasis-open.org/ws-tx/wscoor/2006/06}RegistrationPortType.



**Рис. 72. Группы {http://docs.oasis-open.org/ws-tx/wsat/2006/06}CoordinatorPortType и {http://docs.oasis-open.org/ws-tx/wscoor/2006/06}RegistrationPortType**

При отправке сообщения для транзакционных вызовов выбор маршрута выполняется в следующем порядке:

1. По имени portType получателя и соответствию значений бизнес-параметров, а также по наличию уже зарегистрированного в координаторе данного portType.
2. По имени portType получателя и соответствию значений бизнес-параметров.
3. По имени portType получателя.
4. По таблице маршрутизации по умолчанию.

При отсутствии маршрута система генерирует для сообщения исключение `NoRouteAvailable`.

В бизнес-процессе Transaction Coordinator система хранит таблицу маршрутизации с элементами partnerLinkType и связанными с ними адресами. Система выполняет маршрутизацию любого транзакционного запроса по следующим правилам:

- При заданной для partnerLinkType маршрутизации по бизнес-параметрам система определяет маршрут согласно заданной маршрутизации:
  - при совпадении маршрута с уже имеющимся адресом в таблице маршрутизации система выполняет маршрутизацию по этому маршруту. При этом система передает Coordination Context.
  - при несовпадении маршрута с уже имеющимся адресом в таблице маршрутизации или при использовании нового partnerLinkType система создает новый маршрут. При этом система передает Coordination Context.
  - при наличии нескольких маршрутов система выбирает любой. При этом система не передает Coordination Context.
- При незаданной для partnerLinkType маршрутизации по бизнес-параметрам система определяет маршрут на основе таблицы маршрутизации:

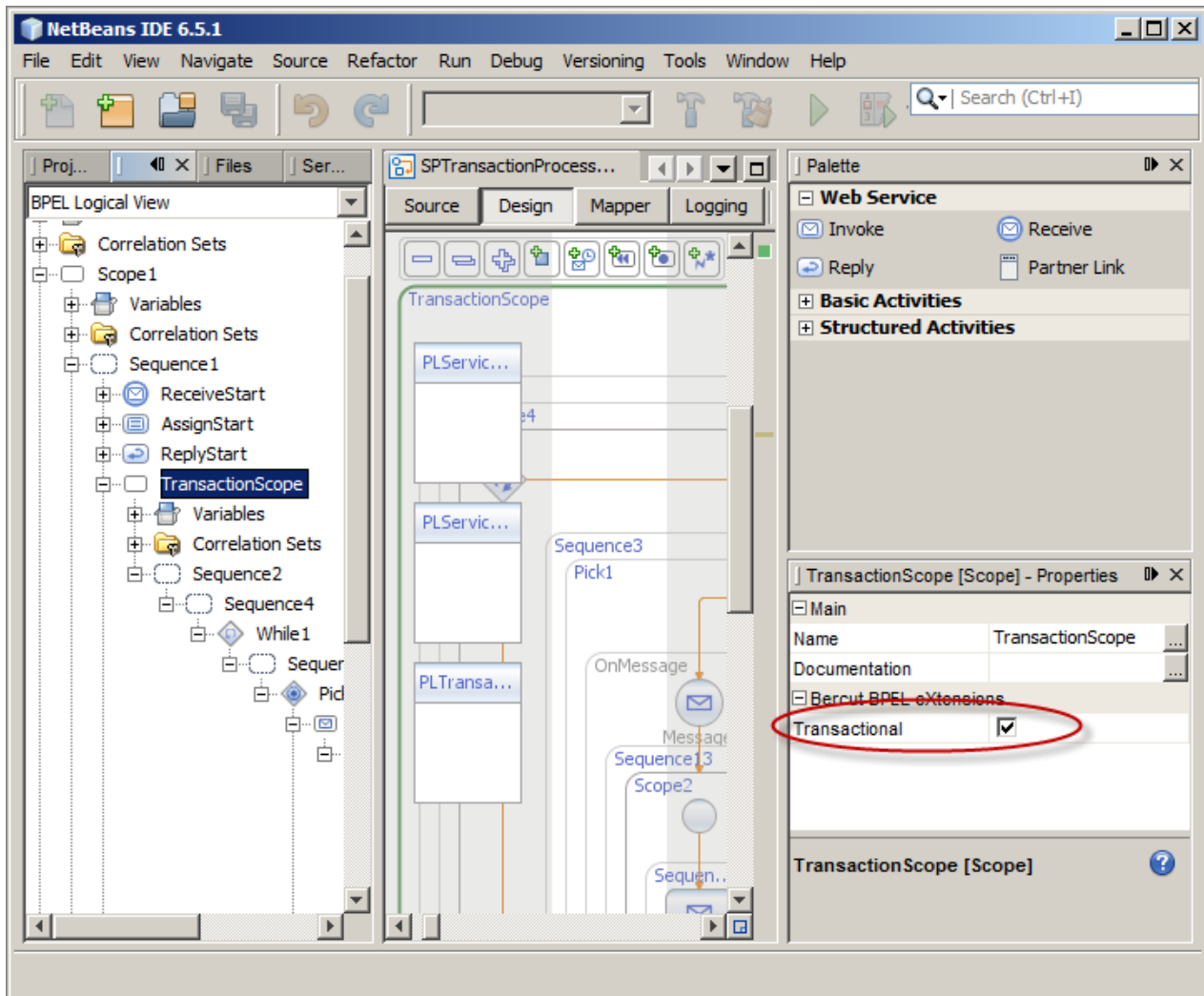
- при использовании нового `partnerLinkType` система создает новый маршрут. При этом система передает `Coordination Context`.
- при наличии одного маршрута для `partnerLinkType` система выполняет маршрутизацию по нему. При этом система не передает `Coordination Context`.
- при наличии нескольких маршрутов для `partnerLinkType` система выполняет маршрутизацию по любому из них. При этом система не передает `Coordination Context`.

**i Примечание.** Система передает `Coordination Context` только новым участникам транзакции. Зарегистрированным ранее участникам система передает идентификатор транзакции.

Для реализации транзакционности в ВР используйте блок `Transaction Scope`. Соблюдайте следующие правила по сравнению с обычным блоком `Scope`:

- вызовы операций PL, реализованных на основе `partnerLinkType`, система выполняет в контексте одной транзакции.
- вызовы операций PL, реализованных на основе не транзакционного `partnerLinkType`, система выполняет как обычные вызовы и не использует `tid`.
- при возникновении исключения или превышении тайм-аута система отменяет все изменения в рамках транзакции.
- при корректном выполнении блока система фиксирует все изменения.
- `Transaction Scope` блок не может быть вложенным для другого блока `Transaction Scope`.

Для установки признака транзакционности поставьте флаг `Transactional` в блоке `Transaction Scope` в среде разработки NetBeans.



**Рис. 73. Установка признака транзакционности**

При поступлении входящего сообщения на транзакционный `partnerLinkType`, не имеющего транзакции внутри себя, система формирует автономную или изолированную транзакцию для этого входящего вызова.

Для транзакционных вызовов в RTSIB задайте тайм-аут по умолчанию, ограничивающий период отсутствия активности в рамках RTSIB-сессии, в MIB-парамetre `DefaultTransactionTimeout` группы `@libs/MessageBus/Core/Configuration/Timeouts`. После окончания тайм-аута система закрывает транзакцию и отправляет клиенту событие `onTimeout`.

LIB_MessageBus_v	01	DefaultInvokeTimeout	LongInt	5 (0x5)
About	01	DefaultSessionTimeout	LongInt	30 (0x1E)
Core	01	DefaultTransactionTimeout	LongInt	10 (0xA)
Configuration	01	OutgoingMessageQueueTimeout	LongInt	1 (0x1)
ComponentSettings				
CustomLogging				
GC				
Routing				
Timeouts				

**Рис. 74. Группа `@libs/MessageBus/Core/Configuration/Timeouts`**

Для развернутых бизнес-процессов система сохраняет статистику транзакционности бизнес-процесса и portType (9.7).

## 8.13. Настройка сессионности

Настройте параметры сессионности при создании и изменении бизнес-процессов, а также в MIB.

Для сессионных процессов установите значение True атрибута *session*. Атрибут расположен в теге `<plnk:partnerLinkType>` WSDL-описания интерфейса. Пример:

```
<plnk:partnerLinkType name="SessionalCMS" xmlns:bwX="http://bercut.com/wsdl/extensions"
  bwX:session="true" bwX:transaction="false">
  <plnk:role name="CMSPortTypeRole" portType="tns:CMSPortType" bwX:roleType="client"/>
  <plnk:role name="ActivationManagerPortTypeRole" portType="tns:ActivationManagerPortType"
    bwX:roleType="server"/>
</plnk:partnerLinkType>
```

partnerLinkType может выполнять одну из трех ролей:

- Серверную.
- Клиентскую.
- Серверную и клиентскую. При этом partnerLinkType обязательно сессионный, взаимодействие обоих ролей происходит в рамках одной сессии.

Для обращения к конечной точке или объекту при сессионном взаимодействии используется механизм Endpoint Reference (EPR). Созданные в рамках сессии EPR передаются в SOAP header. Для однозначной идентификации сессии используйте поля:

- *wsa:To*. IP-адрес назначения и идентификатор удаленного объекта.
- *wsa:From*. IP-адрес отправителя и идентификатор локального объекта.

В иницирующем сессионном запросе полю *wsa:To* соответствует поле *destination* SOAP header, а полю *wsa:From* соответствует поле *source endpoint*. Пример иницирующего сессионного запроса:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.bercut.com/ws/2013/01/addressing">
<S:Header>
  <wsa:To>rtsib://192.0.2.0:3030</wsa:To>
  <wsa:From>
    <wsa:Address>rtsib://198.51.100.0:1020/123456789</wsa:Address>
    <wsa:ReferenceParameters>
      <tns:RequestTo>rtsib://198.51.100.0:3040/123456789 </tns:RequestTo>
    </wsa:ReferenceParameters>
  </wsa:From>
</S:Header>
<S:Body>
  ...
</S:Body>
</S:Envelope>
```

В сессионном ответе значению поля *destination* SOAP header должно соответствовать значение поля *source endpoint* иницирующего сессионного запроса. Пример сессионного ответа:

```
Пример:
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.bercut.com/ws/2013/01/addressing">
<S:Header>
  <wsa:To>rtsib://198.51.100.0:1020/123456789</wsa:To>
```

```

    <wsa:From>
      <wsa:Address>rtsib://192.0.2.0:3030/987654321</wsa:Address>
    </wsa:From>
  </S:Header>
  <S:Body>
    .....
  </S:Body>
</S:Envelope>

```

Запрос, продолжающий сессионное взаимодействие, в поле *wsa:To* содержит адрес назначения и идентификатор удаленного объекта, полученный в ответе или запросе. Поле *wsa:From* содержит информацию об инициаторе запроса. Пример запроса на продолжение сессионного взаимодействия:

```

<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.bercut.com/ws/2013/01/addressing">
  <S:Header>
    <wsa:To>rtsib://192.0.2.0:3030/987654321</wsa:To>
    <wsa:From>
      <wsa:Address>rtsib://198.51.100.0:1020/123456789</wsa:Address>
    </wsa:From>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>

```

Пример ответа на продолжение сессионного взаимодействия:

```

<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.bercut.com/ws/2013/01/addressing">
  <S:Header>
    <wsa:To>rtsib://198.51.100.0:1020/123456789</wsa:To>
    <wsa:From>
      <wsa:Address>rtsib://192.0.2.0:3030/987654321</wsa:Address>
    </wsa:From>
  </S:Header>
  <S:Body>
    .....
  </S:Body>
</S:Envelope>

```

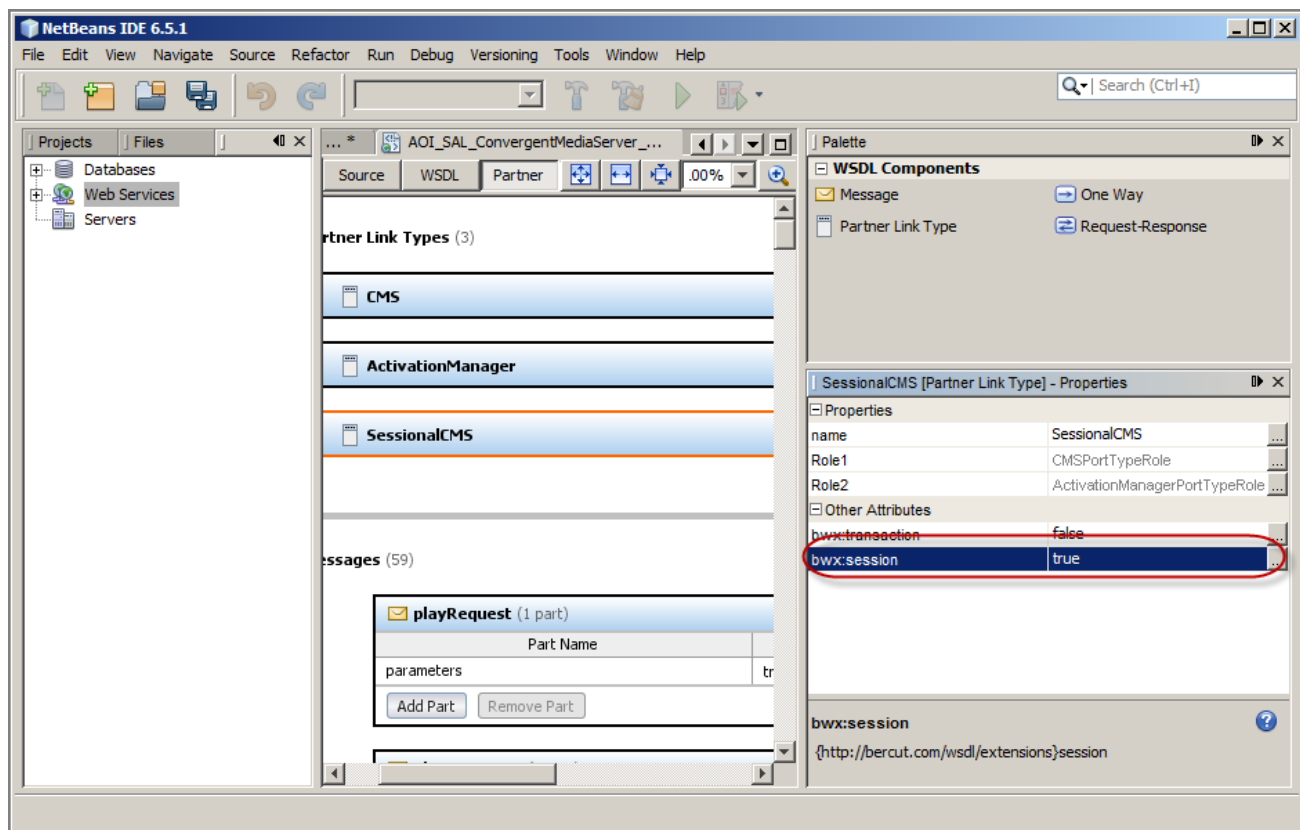
Запрос на завершение сессии содержит признак *Terminate* в поле *wsa:From*. Пример запроса на завершение сессии:

```

<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.bercut.com/ws/2013/01/addressing">
  <S:Header>
    <wsa:To>192.0.2.0:3030</wsa:To>
    <wsa:From>
      <wsa:Address>rtsib://192.0.2.0:3030/123456789</wsa:Address>
    <wsa:ReferenceParameters>
      ...
      <tns:Terminate/>
    </wsa:ReferenceParameters>
    <wsa:From>
  </S:Header>
</S:Envelope>

```

Для установки признака сессионности установите значение **true** параметра *session* для нужного блока в среде разработки NetBeans.



**Рис. 75. Установка признака сессионности**

При отправке сообщения для сессионных вызовов выбор маршрута выполняется в следующем порядке:

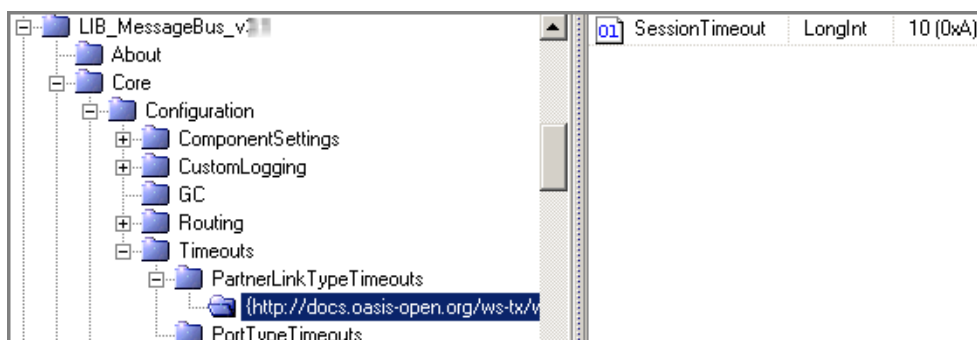
1. По имени portType получателя и соответствию значений бизнес-параметров, а также по наличию уже установленной сессии к данному portType.
2. По имени portType получателя и наличию уже установленной сессии к данному portType.
3. По имени portType получателя и соответствию значений бизнес-параметров.
4. По имени portType получателя.
5. По таблице маршрутизации по умолчанию.

При отсутствии маршрута система генерирует для сообщения исключение `NoRouteAvailable`.

Для сессионных вызовов в RTSIB задайте тайм-аут, ограничивающий период отсутствия активности в рамках RTSIB-сессии. После окончания тайм-аута система удаляет сессию и отправляет клиенту событие `onTimeout`.

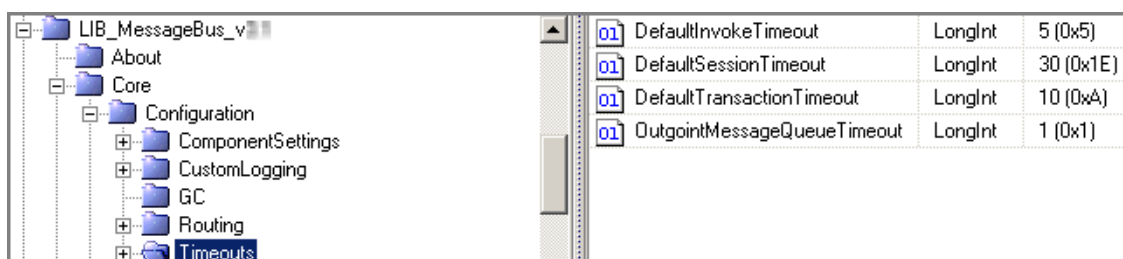
Задайте тайм-аут для всех сессий конкретного PartnerLinkType в MIB-параметре `SessionTimeout` группы `@libs/MessageBus/Core/Configuration/Timeouts/PartnerLinkTypeTimeouts/<Имя PartnerLinkType>`.





**Рис. 76. Группа @libs/MessageBus/Core/Configuration/Timeouts/PartnerLinkTypeTimeouts/<Имя PartnerLinkType>**

Задайте тайм-аут для сессий по умолчанию в MIB-параметре *DefaultSessionTimeout* группы @libs/MessageBus/Core/Configuration/Timeouts.



**Рис. 77. Группа @libs/MessageBus/Core/Configuration/Timeouts**

Тайм-аут, заданный в переменной *SessionTimeout*, более приоритетен. Если он не задан, система использует тайм-аут из переменной *DefaultSessionTimeout*.

Для развернутых бизнес-процессов система фиксирует статистику сессионности бизнес-процесса и portType (9.7).

## 8.14. Совместимость версий WSDL

### 8.14.1. Обратная совместимость (Backward compatibility)

При обратной совместимости новый контракт расширяет возможности старого. Существующие сервисы работают без изменений.

#### Пример:

Исходный контракт:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://bercut.com/schema/po" xmlns="http://actioncon.com/
  schema/po">
  <xsd:complexType name="LineItemType">
    <xsd:sequence>
      <xsd:element name="productID" type="xsd:string"/>
      <xsd:element name="productName" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="LineItem" type="LineItemType"/>
</xsd:schema>
```

```
</xsd:schema>
```

Новый контракт, в котором добавлен элемент `<xsd:element name="available" type="xsd:boolean" minOccurs="0"/>`:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://bercut.com/schema/po" xmlns="http://actioncon.com/
  schema/po">

  <xsd:complexType name="LineItemType">
    <xsd:sequence>
      <xsd:element name="productID" type="xsd:string"/>
      <xsd:element name="productName" type="xsd:string"/>
      <xsd:element name="available" type="xsd:boolean" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>
```

Новый контракт не вносит изменений для существующих сервисов, которые не связаны с элементом `available`.

## 8.14.2. Прямая совместимость (Forward compatibility)

*Прямая совместимость вносит изменения для существующих сервисов.*

Для добавления дополнительного элемента `xsd:any` с существующим элементом `namespace` используйте следующие варианты:

- элемент `namespace="#any"` может быть любым, но должен отличаться от используемых в данной схеме;
- описание промежуточного дополнительного элемента, который содержит элемент `xsd:any`;
- новая версия библиотеки Xerces:
  - скачайте Xerces2 Java 2.11.0 (XML Schema 1.1) (Beta) с сайта <http://xerces.apache.org/mirrors.cgi>;
  - в каталоге `%JAVA_HOME%\jre\lib\jaxp.properties` задайте строку:

```
javax.xml.validation.SchemaFactory\http://www.w3.org/2001/
XMLSchema=org.apache.xerces.jaxp.validation.XMLSchema11Factory
```

### Пример варианта 2:

Исходный контракт с добавлением элемента `<element name="productOther" minOccurs="0"/>`:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://bercut.com/schema/po"
  xmlns="http://bercut.com/schema/po">

  <xsd:complexType name="LineItemType">
    <xsd:sequence>
      <xsd:element name="productID" type="xsd:string"/>
      <xsd:element name="productName" type="xsd:string" minOccurs="0"/>

      <xsd:element name="productOther" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:any minOccurs="0" maxOccurs="unbounded"
              processContents="lax"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
```

```

        </xsd:element>

        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>

```

#### Первое расширение:

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://bercut.com/schema/po"
xmlns="http://bercut.com/schema/po">

    <xsd:complexType name="LineItemType">
        <xsd:sequence>
            <xsd:element name="productID" type="xsd:string"/>
            <xsd:element name="productName" type="xsd:string" minOccurs="0"/>

            <xsd:element name="productOther" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="productID" type="nonNegativeInteger"/>

                        <xsd:element name="productOther" minOccurs="0">
                            <xsd:complexType>
                                <xsd:sequence>
                                    <xsd:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>

```

Последующие расширения формируются аналогично первому.

### 8.14.3. Изменение контракта

*Изменения контракта могут быть совместимыми и несовместимыми.*

Совместимые и несовместимые изменения контракта.

- изменения, после которых существующие сервисы не могут работать, называются *несовместимыми*. При таких изменениях изменяется версия контракта;
- изменения контракта, когда сохраняется работоспособность существующих сервисов, называются *совместимыми*.

Используйте следующие правила:

- при внесении совместимых обратных или прямых изменений меняется минорная версия, при внесении несовместимых изменений — мажорная;
- изменяйте *targetNamespace* так, чтобы XSD-типы данных для несовместимых версий отличались. Например: если пространство имен для первой версии — `http://bercut.com/contract/po/v1`, то для следующей несовместимой версии задайте пространство имен: `http://bercut.com/contract/po/v2`;
- для совместимых изменений пространство имен не изменяется.

## Несовместимые изменения

Несовместимые изменения:

- добавление необязательных XSD-элементов или атрибутов;
- удаление XSD-элементов или атрибутов;
- переименование XSD-элементов или атрибутов;
- уточнение ограничений на XSD-элементы или атрибуты;
- переименование WSDL-операций;
- удаление WSDL-операций;
- изменение MEP (Message Exchange Pattern) WSDL-операций: удаление, добавление, изменение входных или выходных элементов в операции;
- добавление новых ошибок.

## Совместимые изменения

Совместимые изменения:

- расширение ограничений;
- добавление необязательного элемента или атрибута *вниз*;
- добавление WSDL-операции и сообщения, связанного с ней;
- добавление portType;
- добавление новой связи (binding) и сервиса.

## Правила перехода от несовместимых изменений к совместимым

Действия:

- переименование операции. Допустимые варианты расширения контракта:
  - несовместимый. Переименование WSDL-операции и изменение мажорной версии;
  - совместимый. Добавление новой или переименованной операции в существующий контракт.
- изменение MEP. Допустимые варианты расширения контракта:
  - несовместимый. Изменение MEP и мажорной версии;
  - совместимый. Создание новой операции и добавление входных или выходных параметров.
- добавление или удаление ошибок. Допустимые варианты расширения контракта:
  - несовместимый. Изменение ошибки и мажорной версии;
  - совместимый. Создание новой операции и добавление или удаление ошибок.

### Пример несовместимых изменений

Исходная схема с `xsd:element name="productName" type="xsd:string" minOccurs="0"/>`:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://bercut.com/schema/po"
xmlns="http://bercut.com/schema/po">

  <xsd:complexType name="LineItemType">
    <xsd:sequence>
      <xsd:element name="productID" type="xsd:string"/>
      <xsd:element name="productName" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="LineItem" type="LineItemType"/>
</xsd:schema>
```

```
</xsd:schema>
```

**Несовместимые изменения:** уточнение ограничений элемента **productName:**

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://bercut.com/schema/po"
xmlns="http://bercut.com/schema/po">

<xsd:complexType name="LineItemType">
  <xsd:sequence>
    <xsd:element name="productID" type="xsd:string"/>
    <xsd:element name="productName" type="xsd:string" minOccurs="3"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>
```

### Примеры совместимых изменений

**Исходная схема:**

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://bercut.com/schema/po"
xmlns="http://bercut.com/schema/po">

<xsd:complexType name="LineItemType">
  <xsd:sequence>
    <xsd:element name="productID" type="xsd:string"/>
    <xsd:element name="productName" type="xsd:string" maxOccurs="2" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>
```

**Измененная схема, использующая maxOccurs = "3" для элемента productName:**

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://bercut.com/schema/po"
xmlns="http://bercut.com/schema/po">

<xsd:complexType name="LineItemType">
  <xsd:sequence>
    <xsd:element name="productID" type="xsd:string"/>
    <xsd:element name="productName" type="xsd:string" maxOccurs="3" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>
```

В результате старые сервисы работают без изменений, новые сервисы считают добавленный элемент необязательным.

**Измененная схема с необязательным элементом**  
**<xsd:element name="available" type="xsd:boolean" minOccurs="0"/>:**

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://bercut.com/schema/po"
xmlns="http://bercut.com/schema/po">

<xsd:complexType name="LineItemType">
  <xsd:sequence>
    <xsd:element name="productID" type="xsd:string"/>
    <xsd:element name="productName" type="xsd:string" minOccurs="2"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="LineItem" type="LineItemType"/>

</xsd:schema>
```

```

    <xsd:element name="available" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="LineItem" type="LineItemType"/>
</xsd:schema>

```

#### 8.14.4. Правила расширения WSDL-операций

Для расширения WSDL-описаний с помощью элемента **xsd:any** используйте следующие типы:

- xsd:sequence
- xsd:enumeration
- xsd:choice

##### 8.14.4.1. Тип xsd:sequence

Для обеспечения расширения тип *xsd:sequence* использует *<xsd:element name="productOther" minOccurs="0">*.

Если элемент типа *xsd:sequence* получает количество полей больше, чем определяет его описание, то все неизвестные ему поля-элементы пропускаются и помещаются в *any* ?. Исключения не формируются.

#### Пример:

##### ProductType(1):

```

<xsd:complexType name="ProductType">
  <xsd:sequence>
    <xsd:element name="productID" type="xsd:string"/>

    <xsd:element name="productOther" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any minOccurs="0" maxOccurs="unbounded"
            processContents="lax"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

##### ProductType(2):

```

<xsd:complexType name="ProductType">
  <xsd:sequence>
    <xsd:element name="productID" type="xsd:string"/>

    <xsd:element name="productOther" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="productName" type="xsd:string" />

          <xsd:element name="productOther" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:any minOccurs="0" maxOccurs="unbounded"
                  processContents="lax"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>

</xsd:sequence>
</xsd:complexType>

```

ProductType(1) backward-forward совместим с ProductType(2). ProductType(2) расширен элементом `<xsd:element name="productName" type="xsd:string" />`

Сервис на основе ProductType(1) при получении использует элемент `<productID>` и не использует элемент `<productName>`. При декодировании элемент `<productName>` будет пропущен и помещен в any.

```

<....>
  <productID>Books</productID>
  <productOther>
    <productName>Name_of_the_book</productName>
  </productOther>
</....>

```

#### 8.14.4.2. Тип `xsd:enumeration`

Для обеспечения расширения для типа `xsd:enumeration` может быть определено значение 'unknown' для явного использования.

Если элемент получает значение не описанное в `enumeration`, то ему присваивается значение: 'unknown-enumeration'. Исключения не формируются.

##### Пример:

##### Money(1):

```

<xs:simpleType name="Money">
  <xs:restriction base="xs:string">
    <xs:enumeration value="dollar"/>
    <xs:enumeration value="euro"/>

    <xs:enumeration value="__unknown_enumeration__"/>
  </xs:restriction>
</xs:simpleType>

```

##### Money(2):

```

<xs:simpleType name="Money">
  <xs:restriction base="xs:string">
    <xs:enumeration value="dollar"/>
    <xs:enumeration value="euro"/>
    <xs:enumeration value="rub"/>

    <xs:enumeration value="__unknown_enumeration__"/>
  </xs:restriction>
</xs:simpleType>

```

Money(1) backward-forward совместим с Money(2). Money(2) расширен элементом `<xs:enumeration value="rub"/>`.

### 8.14.4.3. Тип `xsd:choice`

Для обеспечения расширения для типа `xsd:choice` должен быть определен элемент `unknown`.

Если элемент получает значение не описанное в `choice`, то ему присваивается значение: 'unknown'. Исключения не формируются.

#### Пример:

##### Money(1):

```
<xs:complexType name="Money">
  <xs:choice>
    <xs:element name="dollar" type="xs:string"/>
    <xs:element name="euro" type="xs:string"/>

    <xs:element name="__unknown_choice__" type="xs:anyType" />
  </xs:choice>
</xs:complexType>
```

##### Money(2):

```
<xs:complexType name="Money">
  <xs:choice>
    <xs:element name="dollar" type="xs:string"/>
    <xs:element name="euro" type="xs:string"/>
    <xs:element name="rub" type="xs:string"/>

    <xs:element name="__unknown_choice__" type="xs:anyType" />
  </xs:choice>
</xs:complexType>
```

Money(1) backward-forward совместим с Money(2). Money(2) расширен элементом `<xs:element name="rub" type="xs:string"/>`.

## 8.15. Кеширование ответов на внешние вызовы

*RTSIB* позволяет кешировать ответы для уменьшения объема сетевого трафика.

Система проверяет критерий допустимости кеширования запроса.

- Если критерий не выполняется, система отправляет запрос по маршруту без кеширования.
- Если критерий выполнен, система хеширует параметры запроса операции. Хеш используется как ключ для кеширования данных ответа на запрос.

Параметры кеширования хранятся в следующих MIB-группах:

- `@libs/MessageBus/Core/Configuration/Cache` — для всех `portType`;
- `@libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>` — для `portType`;
- `@libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/Operations/<имя операции>` — для операции `portType`;

Группы содержат следующие параметры, определяющие политику обновления кеша:

- *Enabled*. Флаг включения кеширования:
  - True — кеширование включено;



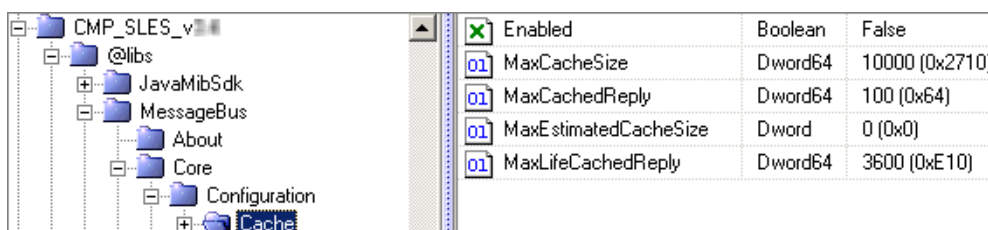
- False — кеширование выключено.

Значение по умолчанию: 'False'.

- *MaxCacheSize*. Максимальное количество кешированных ответов. При значении '0' количество не ограничено. Значение по умолчанию: '10000'.
- *MaxCachedReply*. Максимальное количество ответов до обнуления кеша. При значении '0' количество не ограничено. Значение по умолчанию: '100'.
- *MaxLifeCachedReply*. Максимальное время жизни ответов до обнуления кеша, в секундах. При значении '0' время не ограничено. Значение по умолчанию: '3600'.
- *MaxEstimatedCacheSize*. Максимальное количество кешированных ответов, в мегабайтах. При значении '0' количество ответов неограничено.

### **Примечание.**

1. Чтобы кеширование работало для операции, поставьте флаги *Enabled* для операции, *portType* и в корневой группе. Если один из флагов не поставлен, кеширование работать не будет.
2. Значения параметров *MaxCacheSize*, *MaxCachedReply*, *MaxLifeCachedReply* и *MaxEstimatedCacheSize* более приоритетны для корневой группы. Например: если заданы значения *MaxCacheSize* для операции, *portType* и в корневой группе, система использует значение из корневой группы. Если заданы значения для операции и *portType*, система использует значение для *portType*.



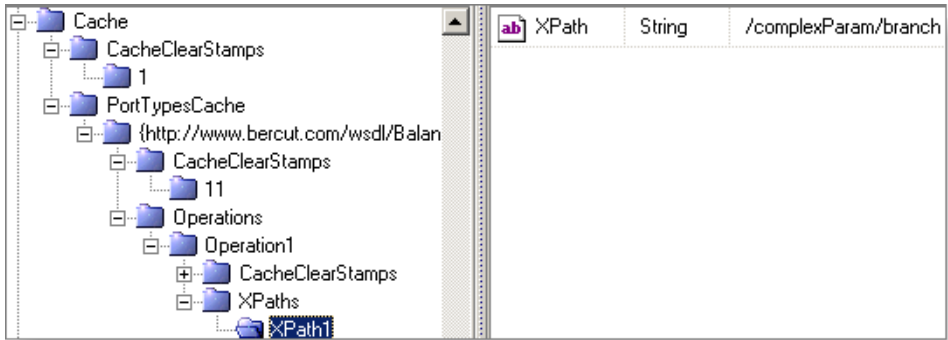
<input checked="" type="checkbox"/>	Enabled	Boolean	False
<input type="checkbox"/>	MaxCacheSize	Dword64	10000 (0x2710)
<input type="checkbox"/>	MaxCachedReply	Dword64	100 (0x64)
<input type="checkbox"/>	MaxEstimatedCacheSize	Dword	0 (0x0)
<input type="checkbox"/>	MaxLifeCachedReply	Dword64	3600 (0xE10)

**Рис. 78. Параметры кеширования**

Для всех *portType* поставьте флаг *CacheBusinessFaults* для включения кеширования бизнес-ошибок.

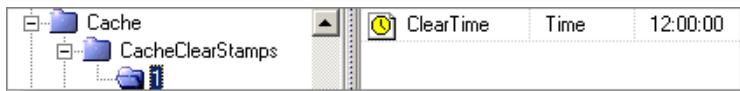
Для операции задайте:

- значение переменной *HashType* — тип подсчета хеша:
  - include — хеш считается по указанным полям, заданным в XPath-выражениях;
  - exclude — хеш считается по всем полям запроса, кроме указанных в XPath.
- XPath-пути для подсчета хеша в переменных *XPath* групп *@libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/Operations/<имя операции>/XPaths/<название узла XPath>*.



**Рис. 79. Группа @libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/Operations/<имя операции>/XPaths/<название узла XPath>**

Задайте время выполнения очистки хеша в формате: HH:MM:SS. Время задается в переменной *ClearTime* групп @libs/MessageBus/Core/Configuration/Cache/CacheClearStamps/<тег времени очистки кеша>, @libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/CacheClearStamps/<тег времени очистки кеша> и @libs/MessageBus/Core/Configuration/Cache/PortTypesCache/<имя portType>/Operations/<имя операции>/CacheClearStamps/<тег времени очистки кеша>.



**Рис. 80. Задание времени выполнения очистки кеша**

## 8.16. Настройка уровня логирования по бизнес-параметрам

Уровень логирования запроса задается в MIB-группе @libs/MessageBus/Core/Configuration/CustomLogging.

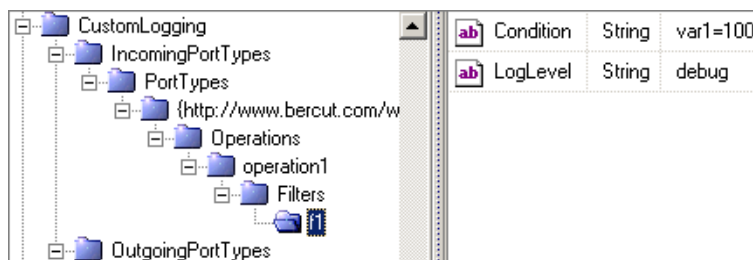
В текущей версии можно устанавливать уровень логирования для конкретного запроса. Значение является более приоритетным, чем локальный уровень трассировки.

Для настройки входящих сообщений используйте MIB-группу @libs/MessageBus/Core/Configuration/CustomLogging/IncomingPortTypes. Для настройки исходящих сообщений используйте MIB-группу @libs/MessageBus/Core/Configuration/CustomLogging/OutgoingPortTypes.

Порядок настройки:

1. Создайте вложенную группу с именем portType в соответствующей MIB-группе.
2. В группе @libs/MessageBus/Core/Configuration/CustomLogging/<Incoming или Outgoing>PortTypes/PortTypes/<имя portType>/Operations создайте вложенную группу с именем операции.
3. В группе @libs/MessageBus/Core/Configuration/CustomLogging/<Incoming или Outgoing>PortTypes/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра> задайте значения MIB-параметров:
  - *Condition*. Условие срабатывания операции.

- *LogLevel*. Уровень логирования запроса.



**Рис. 81.** Группа `@libs/MessageBus/Core/Configuration/CustomLogging/<Incoming или Outgoing>PortTypes/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>`

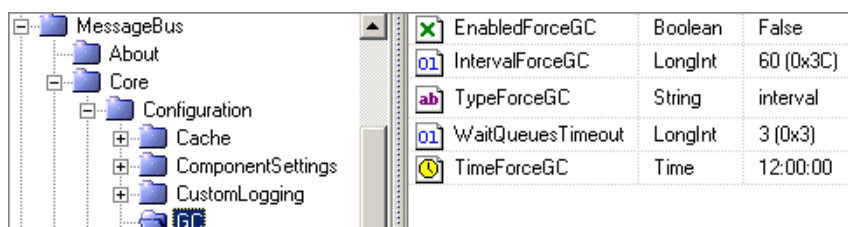
На рисунке приведен пример настройки уровня логирования debug для входящих вызовов операции operation1, для параметра var1=100.

## 8.17. Настройки программы очистки памяти

Настройки программы очистки памяти (*Garbage Collector, GC*) расположены в группе `@libs/MessageBus/Core/Configuration/GC`.

Задайте MIB-параметры группы `@libs/MessageBus/Core/Configuration/GC`:

- *EnabledForceGC*. Флаг принудительного запуска программы очистки памяти.
- *IntervalForceGC*. Интервал принудительного запуска программы очистки памяти. Интервал задается в минутах.
- *TypeForceGC*. Тип принудительного запуска программы для очистки памяти:
  - interval — GC запускается с интервалом *IntervalForceGC*;
  - time — GC запускается в указанное во время *TimeForceGC*.
- *WaitQueuesTimeout*. Тайм-аут ожидания очистки очередей перед вызовом GC, в минутах.
- *TimeForceGC*. Время принудительного запуска программы для очистки памяти. Формат: HH:MM:SS.



**Рис. 82.** Группа `@libs/MessageBus/Core/Configuration/GC`

## 8.18. REST-представление с использованием HTTP Binding

*HTTP Binding* позволяет создать REST-представление для сервиса и выполнять REST-вызовы из сервисов. HTTP Binding может быть описан в WSDL-описании или создан автоматически.

**JSON** — способ передачи параметров POST-запроса в теле сообщения, закодированных в JSON. Пример:

```
POST /subscribers/payment
{
  "msisdn": "1234567890",
  "value": "100"
}
```

**UriReplacement** — способ передачи параметров GET/POST-запроса как части URL. Например: GET /subscribers/1234567890/balance.

```
POST /subscribers/1234567890/payment
```

**UriEncoded** — способ передачи параметров GET-запроса в переменной части URL. Пример:

```
GET /subscribers/balance?msisdn=1234567890
```

**UriEncoded+UriReplacement** — способ передачи параметров GET-запроса как части URL и переменной части URL. При этом способе бизнес-параметры передают в URL, а технические параметры (фильтрация) — в переменной части. Пример:

```
GET /subscribers/1234567890/payments?page=1&count=10
```

**JSON+UriReplacement** — способ передачи параметров POST-запроса как части URL и в теле сообщения. Параметры закодированы в JSON. Пример:

```
POST /subscribers/1234567890/payment
{
  "value": "100"
}
```

**RPC/Encoded** — стиль представления данных операции в виде набора part примитивного типа. Для каждой part указывается передаваемый тип. Пример:

```
<soapenv:Body soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <teswsdl:TestWsdJsonGetUrlEncodedOperation>
    <tesxsd:singleStringElement xsi:type="xsd:string">1</tesxsd:singleStringElement>
    <tesxsd:singleIntElement xsi:type="xsd:int">4</tesxsd:singleIntElement >
    ...
  </teswsdl:TestWsdJsonGetUrlEncodedOperation>
</soapenv:Body>
```

Вызываемая операция идентифицируется корневым XML-элементом, который содержит имя операции. Тип и состав передаваемых параметров может отличаться, поэтому каждый раз передается тип элемента. Данный тип HTTP Binding не используется при автоформировании SOAP Binding. Разработчик может объявить его самостоятельно.

**RPC/Literal** — способ представления данных операции в виде набора part разного типа, который заранее известен на противоположной стороне. В этом случае нет необходимости каждый раз передавать тип данных. Пример:

```
<soapenv:Body>
  <tes:TestJsonUriReplacementOperation>
    <singleStringPart>25</singleStringPart>
    <json>
      <singleIntElement>4</singleIntElement>
```

```

        <singleBooleanElement>true</singleBooleanElement>
        ...
    </json>
    <singleDateTimePart>2008-09-25T21:10:36</singleDateTimePart>
</tes:TestJsonUrlReplacementOperation>
</soapenv:Body>

```

Данный тип HTTP Binding используется при автоформировании SOAP Binding.

**Document/Literal** — стиль представления данных операции в виде набора part разного типа, который заранее известен на противоположной стороне. Этот стиль не используется из-за сложной идентификация операции. Операция идентифицируется по уникальному набору элементов в теле сообщения, обертки с именем операции как в RPC, нет. Вместо этого стиля используется Document/Literal Wrapped.

```

<soapenv:Body>
  <tes:singleStringElement>1</tes:singleStringElement>
  <tes:arrayStringElement>3</tes:arrayStringElement>
  <tes:singleDateElement>2002-09-24+06:00</tes:singleDateElement>
  ...
</soapenv:Body>

```

**Document/Literal Wrapped** — стиль представления данных операции в одной part, которая ссылается на XML-элемент комплексного типа. Этот XML-элемент идентифицирует вызываемую операцию. Пример:

```

<soapenv:Body>
  <tes:testWsdLJsonRequest>
    <tes:singleStringElement>1</tes:singleStringElement>
    <tes:arrayStringElement>3</tes:arrayStringElement>
    ...
  </tes:testWsdLJsonRequest>
</soapenv:Body>

```

### 8.18.1. Создание HTTP Binding для существующего SOAP-сервиса

Когда вы создаете описание HTTP Binding в WSDL-файле самостоятельно, соблюдайте рекомендации, описанные ниже.

#### Использование соответствующего типа HTTP Binding в зависимости от операции

Существуют следующие ограничения:

- действия для операции в зависимости от типа HTTP Binding:
  - для операции объявлен Document/Literal Binding:
    - для POST и PUT операция определяется в HTTP Binding соответствующего типа;
    - Для GET и DELETE объявите новые типы входящих сообщений и новый portType. Затем определите операцию в HTTP Binding соответствующего типа для нового portType.
  - для операции объявлен Rpc или Rpc/Literal Binding. Операция определяется в HTTP Binding соответствующего типа.
- одна операция может быть заявлена только в одном HTTP Binding.

#### Создание HTTP Binding

Если нужно определить особенный location для операций или применить для операций разные типы HTTP-запросов или типы, отличные от HTTP POST и GET, объявите HTTP Binding вручную.

Для объявления дополнительных HTTP Binding размещайте их в дополнительных WSDL-файлах. Используйте тег `wsdl:import` и не изменяйте основной WSDL-файл. В BPEL и при использовании утилиты WSDLGEN используйте один WSDL-файл, который импортирует все другие.

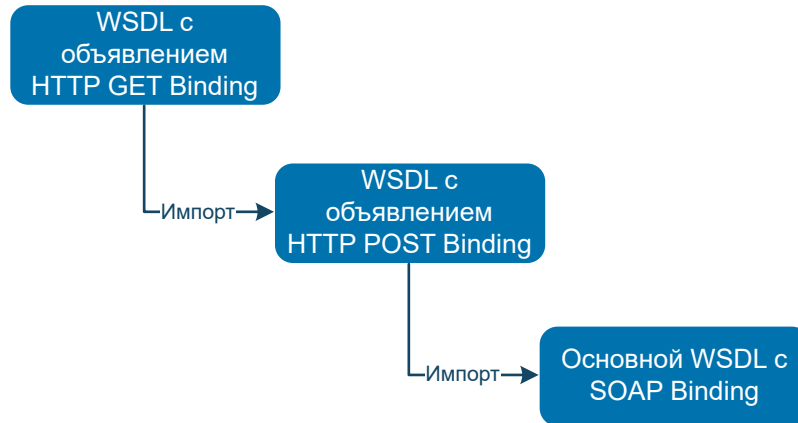


Рис. 83. Изменение WSDL-файла

### Преобразование интерфейса для типов GET и DELETE

Различие POST/PUT и GET/DELETE при объявлении HTTP Binding:

- POST/PUT позволяют передавать любые типы комплексных данных в JSON-кодировании;
- GET/DELETE позволяют передавать одноуровневые структуры с помощью параметров URL.

Если для операции используется входящий комплексный тип, выполните следующие действия для передачи этого запроса с помощью GET/DELETE:

- объявите новый тип WSDL-сообщения, в котором все элементы комплексного типа и используемые типы будут описаны как *part*. Имена *part* могут отличаться от имен исходных элементов.
- объявите новый portType с операциями, которые надо перевести в GET/DELETE. Не изменяйте имена операций. В операциях используйте новый тип сообщения в качестве входящего сообщения. Все остальные сообщения и faults оставьте без изменения.
- объявите HTTP Binding типа GET или DELETE для нового portType и операций.

Объявление новых WSDL-сообщений, portType и Binding выполняйте в новом WSDL-файле. Редактирование и проверку WSDL-файлов выполняйте в SCS [20].

### Примеры:

SOAP:

```

<xsd:complexType name="getSubscriber">
  <xsd:sequence>
    <xsd:element name="branchId" type="ns0:BranchId" minOccurs="1"/>
    <xsd:element name="contractName" type="tns:getSubscriberContractName" minOccurs="1"/>
    <xsd:element name="subscriber" type="ns0:subscriber" minOccurs="1"/>
    <xsd:element name="targetDate" type="tns:targetDate" minOccurs="0" maxOccurs="1"/>
    <xsd:sequence minOccurs="0">
      <xsd:element name="notUseCache" type="xsd:boolean" minOccurs="0"/>
    </xsd:sequence>
    <xsd:element name="clientId" type="ns0:clientId" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<message name="getSubscriber">
  <part name="parameters" element="tns:getSubscriber"/>
</message>
<portType name="AccountManagement">
  
```

```

<operation name="getSubscriber">
  <input message="tns:getSubscriber"/>
  <output message="tns:getSubscriberResponse"/>
  <fault message="tns:SAConnectionException" name="SAConnectionException"/>
  <fault message="tns:SASQLCustomException" name="SASQLCustomException"/>
  <fault message="tns:SASQLException" name="SASQLException"/>
</operation>
</portType>

```

## HTTP GET

```

<message name="getSubscriberHttpRequestMessage">
  <part name="branchId" type="cia:BranchId"/>
  <part name="contractName" type="xsd:string"/>
  <part name="msisdn" type="cia:msisdn"/>
  <part name="subscriberId" type="cia:subscriberId"/>
  <part name="phoneId" type="cia:phoneId"/>
  <part name="servPhoneId" type="cia:phoneId"/>
  <part name="targetDate" type="cia:targetDate"/>
  <part name="notUseCache" type="xsd:boolean"/>
  <part name="clientId" type="cia:clientId"/>
</message>

<portType name="AccountManagementHttpGet">
  <operation name="getSubscriber">
    <input message="tns:getSubscriberHttpRequestMessage"/>
    <output message="acwsdl:getSubscriberResponse"/>
    <fault message="acwsdl:SAConnectionException" name="SAConnectionException"/>
    <fault message="acwsdl:SASQLCustomException" name="SASQLCustomException"/>
    <fault message="acwsdl:SASQLException" name="SASQLException"/>
  </operation>
</portType>
<binding name="AccountManagementGetPortBinding" type="tns:AccountManagementHttpGet">
  <http:binding verb="GET"/>
  <operation name="getSubscriber">
    <http:operation location="/subscribers"/>
    <input>
      <http:urlEncoded/>
    </input>
    <output>
      <mime:content type="application/json" part="parameters"/>
    </output>
    <fault name="SAConnectionException" http:code="501"/>
    <fault name="SASQLCustomException" http:code="502"/>
    <fault name="SASQLException" http:code="503"/>
  </operation>
</binding>

```

## Преобразование реализации интерфейса для типов GET и DELETE

Для реализации интерфейса в основном portType выполняются операции:

- маппинг данных из входящего одноуровневого сообщения в комплексный тип;
- вызов операции основного portType;
- прокидывание ответа;
- прокидывание fault.

**! Внимание!** Ответ и fault не следует преобразовывать в другие структуры — при объявлении новой операции использовались сообщения из основной операции.

## Пример для SA

```

package com.bercut.cc.sa.operations.accountmanagement.http.mb
public class AccountManagementHttpGetEventsImpl
extends AccountManagementEventsImpl
implements AccountManagementHttpGet {

public AccountManagementHttpGetEventsImpl(AtlasProvider atlas) {
  super(atlas);
}

```

```

@Override
public SubscriberProfile getSubscriber(
    BigDecimal branchId,
    String contractName,
    String msisdn,
    BigDecimal subscriberId,
    BigDecimal phoneId,
    BigDecimal servPhoneId,
    XMLGregorianCalendar targetDate,
    Boolean notUseCache,
    BigDecimal clientId,
    ServerRequestParameters requestParameters
) throws SaConnectionErrorException, SasqlCustomExceptionException,
    SasqlExceptionException {

    try {
        Subscriber subscriber = createSubscriber(msisdn, subscriberId, phoneId, servPhoneId);

        com.bercut.cc.sa.accountmanagement.GetSubscriber request =
        new com.bercut.cc.sa.accountmanagement.GetSubscriber(
            branchId,
            contractName,
            subscriber,
            targetDate,
            new com.bercut.cc.sa.accountmanagement.GetSubscriber.Anonymous(notUseCache),
            clientId);

        return com.bercut.cc.sa.operations.accountmanagement.mb.GetSubscriber.getSubscriber(
            this,
            request,
            requestParameters);
    } catch (com.bercut.cc.sa.accountmanagement.SasqlExceptionException e) {
        throw new SasqlExceptionException(e);
    }
}

```

## 8.18.2. Создание HTTP Binding из существующего SOAP Binding

Чтобы создать HTTP Binding из существующего SOAP Binding, используйте следующие способы:

- выполните автоматическое формирование HTTP Binding путем развертывания компонента в SLES;
- скопируйте опубликованный HTTP Binding в новый WSDL-файл;
- добавьте импорт основного WSDL-файла;
- измените значение location в HTTP Binding.

Редактирование и проверку WSDL-файлов выполняйте в SCS [20].

### Пример:

Автоматически сформированный HTTP Binding. Binding и portType находятся в одном WSDL-файле и одном пространстве имен, portType указывается с префиксом tns.

```

<binding name="AutoGeneratedAccountManagementHttpBinding" type="tns:AccountManagement">
  <http:binding verb="POST"/>
  <operation name="getSubscribers">
    <http:operation location="/getSubscribers"/>
    <input name="getSubscribersRequest">
      <mime:content type="application/json" part="parameters"/>
    </input>
    <output name="getSubscribersResponse">
      <mime:content type="application/json" part="parameters"/>
    </output>
    <fault name="SAConnectionErrorException" http:code="452"/>
    <fault name="SASQLCustomException" http:code="453"/>
    <fault name="SASQLException" http:code="454"/>
  </operation>

```



```
</binding>
```

Измененный HTTP Binding находится в отдельном WSDL-файле со своим пространством имен. Имя `portType` указывается с префиксом другого пространства имен — `acwsdl`.

```
<binding name="AccountManagementPostPortBinding" type="acwsdl:AccountManagement">
  <http:binding verb="POST"/>
  <operation name="getSubscribers">
    <http:operation location="/subscribers/search"/>
    <input name="getSubscribersRequest">
      <mime:content type="application/json" part="parameters"/>
    </input>
    <output name="getSubscribersResponse">
      <mime:content type="application/json" part="parameters"/>
    </output>
    <fault name="SAConnectionException" http:code="452"/>
    <fault name="SASQLCustomException" http:code="453"/>
    <fault name="SASQLException" http:code="454"/>
  </operation>
</binding>
```

### 8.18.3. Автоматическое создание HTTP Binding

Механизм автоматического создания HTTP Binding зависит от типа артефакта:

- для BP и SE — HTTP Binding автоматически создается при развертывании (deploy);
- для SA — HTTP Binding автоматически создается при генерации кодов java-заглушек.

#### 8.18.3.1. Автоматическое создание HTTP Binding для BP и SE

Автоматическое формирование HTTP Binding для BP и SE возможно после его включения в настройках и перезапуска SLES или развертывания (удаления) BP и SE.

#### Включение автоматического формирования

Система настроек автоматического формирования HTTP Binding управляет автоматическим формированием при разборе WSDL-интерфейсов или развертывании BP и SE. После развертывания BP и SE настройки не влияют на сформированные HTTP Binding.

Публикация HTTP Binding выполняется независимо от того, были ли они описаны в WSDL-интерфейсе или сформированы при развертывании.

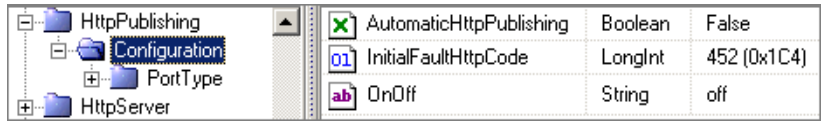
#### Автоматическое формирование HTTP Binding

Для быстрого предоставления HTTP-интерфейса RTSIB может автоматически формировать HTTP Binding.

Для настройки автоматического формирования HTTP Binding задайте MIB-параметры группы `@libs/MessageBus/HttpPublishing/Configuration`:

- *OnOff* — флаг включения формирования HTTP Binding:
  - on — HTTP Binding в соответствии со значением параметра *AutomaticHttpPublishing*;
  - off — HTTP Binding не формируются.
- *AutomaticHttpPublishing* — флаг автоматического формирования HTTP Binding:
  - True — система автоматически формирует HTTP Binding для всех portType, кроме тех, для которых заданы индивидуальные настройки;
  - False — система автоматически не формирует HTTP Binding.

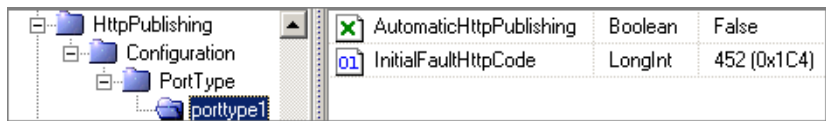
- *InitialFaultHttpCode* — начальное значение HTTP-кода атрибута 'http:code' в ответе fault, которое назначается при формировании HTTP Binding для всех portType.



**Рис. 84. Настройки HTTP Binding**

Чтобы задать индивидуальные настройки для portType, выполните следующие действия:

1. Создайте вложенную группу в группе @libs/MessageBus/HttpPublishing/Configuration/PortType с названием portType.
2. Задайте значения MIB-параметров в группе @libs/MessageBus/HttpPublishing/Configuration/PortType/<имя portType>:
  - *AutomaticHttpPublishing* — флаг автоматического формирования HTTP Binding:
    - True — система автоматически формирует HTTP Binding для этого portType;
    - False — система автоматически не формирует HTTP Binding для этого portType.
  - *InitialFaultHttpCode* — начальное значение HTTP-кода атрибута 'http:code' в ответе fault, которое назначается при формировании HTTP Binding для этого portType.



**Рис. 85. Индивидуальные настройки portType**

Особенности работы настроек:

- В SLES настройки применяются при развертывании BP или SE. Для включения или отключения автоматического формирования необходимо выполнить undeploy или deploy, соответственно.
- В LWSA настройки применяются только для выключения автоматического формирования. Чтобы HTTP Binding присутствовал в публикации интерфейса SA, необходимо использовать опцию автоматического формирования при разработке этой публикации. Во время выполнения можно только отключить публикацию.
- В SG настройки применяются во время работы. При их изменении SG перечитывает WSDL с включенным или отключенным автоматическим формированием. Поэтому для представления REST-интерфейса рекомендуется использовать SG.
- Описание для API Gateway — [13].

### Ошибки при автоматическом формировании

HTTP Binding может быть сформирован не для каждого portType.

Для всех операций portType формируется один HTTP Binding POST или GET типа. Если в portType есть операции с singlepart и multipart сообщениями, система создает HTTP GET Binding. В этом случае для singlepart операций система формирует неверную WSDL-структуру. Система формирует ошибку автоматического формирования.

Если для portType уже описан HTTP Binding с типом, который отличается от определенного для формирования, система формирует ошибку.

При возникновении ошибки система прерывает развертывание. Для развертываемого артефакта можно выполнить следующие действия:

1. Отключить автоматическое формирование HTTP Binding для указанного или для всех portType и включить автоматическое формирование для необходимых portType.
2. Доработать артефакт, добавив описание HTTP Binding.

### 8.18.3.2. Автоматическое создание HTTP Binding для SA

#### Автоматическое формирование HTTP Binding

Автоматическое формирование HTTP Binding для SA может быть выполнено с помощью утилиты WSDLGEN. Утилита формирует класс, реализующий веб-сервис или класс для вызова удаленного веб-сервиса.

Формат:

```
wsdngen {-asynch | -synch} [options] <wsdl path>,
```

где:

asynch | synch — тип реализации синхронная или асинхронная;  
wsdl path – путь к WSDL, который описывает веб-сервис.

Параметры утилиты:

- -interfacepath — каталог, в котором помещаются созданные java-классы. Значение по умолчанию: текущий каталог;
- -sourcerackage — полное имя класса реализации веб-сервиса. Формируется при типе реализации: 'server';
- -sourcerpath — каталог, в который помещается реализация веб-сервиса. Используется при типе реализации: 'server';
- -typepath — каталог, в который помещаются java-классы для работы с типами данных. Для формирования используется утилита xsd-gen;
- -plt <PLT name> — имя элемента PartnerLinkType, по которому выполняется формирование, если их несколько;
- -pgt {asynch | synch} — тип формирования партнерской роли: синхронный или асинхронный. Используется при наличии двух ролей в PartnerLinkType, когда нужно выбрать тип формирования партнера отличный от типа формирования собственного серверного интерфейса;
- -swaproles — переопределение собственной (сервер) и партнерской (клиент) ролей в PartnerLinkType;
- catalogXml — путь к файлу, который содержит переопределение XSD-импортов. Необязательный параметр;
- -noextract — генератор кода, не распаковывает значения элементов из *part* операции. По умолчанию параметр включен. Параметр используется для интерфейсов, у которых сообщения input, output или fault содержат одну часть. Существует возможность распаковать все элементы сложного типа;
- -autoGenerateHttpBinding — признак формирования HTTP Binding для portType. Имя portType определяется автоматически по PartnerLinkType. Если признак включен, HTTP Binding формируется. Если признак выключен, HTTP Binding не формируется.
- -httpBindingInitialFaultHttpCode <http code> — начальное значение HTTP-кода из атрибута 'http:code' в сообщении fault, которое используется при формировании HTTP Binding. Значение по умолчанию: '452'.
- -v — «болтливый» режим вывода.

Пример интерфейса с использованием параметра noextract:

```
public interface DocLiteralSamplePortType {
    public NewComplexType docLiteralSampleOperation(NewComplexType part1, Extension[]
        extensions);
}
```

Тип данных NewComplexType приведен ниже:

```
<xsd:complexType name="newComplexType">
  <xsd:sequence>
    <xsd:element name="field1" type="xsd:int"/>
    <xsd:element name="field2" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

Пример интерфейса без использования параметра noextract:

```
public interface DocLiteralSamplePortType {
  public NewComplexType docLiteralSampleOperation(Integer field1, String field2, Extension[]
  extensions);
}
```

Для ранее созданных SA, которые содержат только SOAP Binding, автоматическое формирование HTTP Binding недоступно.

## 8.18.4. Публикация

### 8.18.4.1. Публикация серверного интерфейса

Для portType с HTTP Binding система публикует 6 URL:

- HTML-версия исходного WSDL;
- исходный WSDL;
- WSDL, который содержит SOAP Binding. Может быть один для portType;
- WSDL, который содержит HTTP Binding. Может быть несколько HTTP Binding для portType, но операция должна быть реализована в одном HTTP Binding;
- JSON-схема используемых структур данных. URL находится в комментариях, в WSDL-интерфейсе с HTTP Binding;
- Описание OpenAPI.

**Пример:**

```
<xsd:schema xmlns:ns2="http://www.bercut.com/test/schema/TestWsdlJson"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified">
  <xsd:import namespace="http://www.bercut.com/test/schema/TestWsdlJson"
    schemaLocation="http://198.51.100.12:7070/test/wsd1/TestWsdlJson/
    TestUrlEncodedAndReplacementPortType?xsd1"/>
  <!-- JsonSchema for XSD "http://www.bercut.com/test/schema/TestWsdlJson"
    http://198.51.100.12:7070/test/wsd1/TestWsdlJson/TestUrlEncodedAndReplacementPortType?
    json1 -->
</xsd:schema>
```

Published services:

<http://www.bercut.com/complex/ComplexQueryPortType> ( [wsdl](#) ) ( [wsdl soap binding](#) ) ( [wsdl http bindings](#) ) ( [open api](#) )

**Рис. 86. Опубликованный интерфейс**

## Публикация сервисов

Для каждого сервиса HTTP на странице публикации сервер отображает ссылки на публикации. Если HTTP-сервер запущен, то по адресу ServerAddress доступна страница публикации. Страница состоит из следующих разделов:

1. Заголовок с именем приложения и версией.
2. Список опубликованных интерфейсов приложений.
3. Список опубликованных системных интерфейсов.

#### 4. Перечень опубликованных REST-операций.

SLES 3.6.18.527		
Published services:		
<a href="http://www.bercut.com/complex/ComplexQueryPortType">http://www.bercut.com/complex/ComplexQueryPortType</a> ( wsdl ) ( wsdl soap binding ) ( wsdl http bindings ) ( open api ) <a href="http://www.bercut.com/openapi/HeadersFaultsServer/HeadersFaultsServerPortType">http://www.bercut.com/openapi/HeadersFaultsServer/HeadersFaultsServerPortType</a> ( wsdl ) ( wsdl soap binding ) ( wsdl http bindings ) ( open api ) <a href="http://www.bercut.com/openapi/wsdl/msgames/MSGamesPortType">http://www.bercut.com/openapi/wsdl/msgames/MSGamesPortType</a> ( wsdl ) ( wsdl soap binding ) ( wsdl http bindings ) ( open api ) <a href="http://www.bercut.com/AllofOneOf/ServerPortType">http://www.bercut.com/AllofOneOf/ServerPortType</a> ( wsdl ) ( wsdl soap binding ) ( wsdl http bindings ) ( open api )		
Published system services:		
<a href="http://docs.oasis-open.org/ws-tx/wsat/2006/06/CoordinatorPortType">http://docs.oasis-open.org/ws-tx/wsat/2006/06/CoordinatorPortType</a> ( wsdl ) <a href="http://docs.oasis-open.org/ws-bw-2/NotificationExpireRecipient">http://docs.oasis-open.org/ws-bw-2/NotificationExpireRecipient</a> ( wsdl ) <a href="http://docs.oasis-open.org/ws-tx/wsat/2006/06/ParticipantPortType">http://docs.oasis-open.org/ws-tx/wsat/2006/06/ParticipantPortType</a> ( wsdl ) <a href="http://docs.oasis-open.org/ws-tx/wscor/2006/06/RegistrationPortType">http://docs.oasis-open.org/ws-tx/wscor/2006/06/RegistrationPortType</a> ( wsdl )		
Published uri:		
Method	Uri	Operation
DELETE	/complex/ComplexQueryPortType/oneComplexSchemasQuery	deleteOneComplexSchemasQuery {http://www.bercut.com/complex/ComplexQueryPortType}
GET	/openapi/wsdl/msgames/MSGamesPortType/games/MDActivate	mdActivateUsingGET {http://www.bercut.com/openapi/wsdl/msgames/MSGamesPortType}
© 2012-2022 Bercut Ltd.		

**Рис. 87. Страница публикации**

#### Формирование URL для HTTP-запроса

Каждый portType имеет единственный SOAP Binding, для которого сформирован URL. На основании этого URL система формирует URL для операций в HTTP Binding с использованием значения обязательного атрибута location.

#### Пример:

portType:

```
{http://www.bercut.com/test/wsdl/TestWsdLJson}TestWsdLJsonPostJsonEncodedPortType
```

SOAP URL:

```
/test/wsdl/TestWsdLJson/TestWsdLJsonPostJsonEncodedPortType
```

Операция:

```
TestWsdLJsonGetUrlEncodedOperation
```

Атрибут location операции:

```
TestWsdLJsonGetUrlEncodedOperation
```

URL операции:

```
/test/wsdl/TestWsdLJson/TestWsdLJsonPostJsonEncodedPortType/  
TestWsdLJsonGetUrlEncodedOperation
```

Операция:

```
TestWsdLJsonPostJsonEncodedOperation
```

Атрибут location операции:

```
TestWsdLJsonPostJsonEncodedOperation
```

URL операции:

```
/test/wsdl/TestWsdlJson/TestWsdlJsonPostJsonEncodedPortType/  
TestWsdlJsonPostJsonEncodedOperation
```

**8.18.4.2. Публикация клиентского интерфейса**

HTTP Binding используются для вызова внешних HTTP-сервисов. Такие интерфейсы не публикуются. Клиентские вызовы с помощью HTTP Binding выполняются в группе [@libs/MessageBus/HttpExternalConnections](#).

**8.18.5. Прием и отправка REST-сообщений**

Перечень опубликованных REST-операций представляет собой таблицу. Колонки Method и Uri образуют уникальную пару, которая сопоставляется операции в WSDL. Одной операции соответствуют одна пара Method и Uri, а одному Uri могут соответствовать разные операции с разными методами.

Published uri:

Method	Uri	Operation
DELETE	/complex/ComplexQueryPortType/oneComplexSchemasQuery	deleteOneComplexSchemasQuery {http://www.bercut.com/complex/}ComplexQueryPortType

**Рис. 88. Перечень опубликованных REST-операций**

**Таблица 1. Пример опубликованных REST-операций**

Метод	URI	Операция
GET	/complex/ComplexQueryPortType/oneComplexSchemasQuery	getOneComplexSchemasQuery {http://www.bercut.com/complex/}ComplexQueryPortType
PUT	/complex/ComplexQueryPortType/oneComplexSchemasQuery	putOneComplexSchemasQuery {http://www.bercut.com/complex/}ComplexQueryPortType
DELETE	/complex/ComplexQueryPortType/oneComplexSchemasQuery	deleteOneComplexSchemasQuery {http://www.bercut.com/complex/}ComplexQueryPortType

**8.18.5.1. Прием REST-сообщений**

При получении входящего HTTP запроса система выполняет следующие действия:

1. Из URL запроса система выбирает базовый SOAP URL и по нему определяет запрашиваемый сервис.
2. По HTTP-заголовку Content-Type и названию HTTP-метода система определяет тип кодирования сообщения и выбирает декодер:
  - SOAP/XML
  - URL ENCODED
  - JSON.

3. Декодер определяет операцию из входящего сообщения.

### 8.18.5.2. Отправка REST-сообщений

#### Выбор Binding для отправки исходящего сообщения

Если portType имеет HTTP Binding (SOAP Binding есть всегда), исходящее сообщение может быть отправлено с помощью SOAP или HTTP Binding.

Чтобы отправить исходящее сообщение с помощью HTTP Binding, выполните следующие действия:

1. Создайте внешнее подключение в группе @libs/MessageBus/HttpExternalConnections/ConnectionsList/porttype/ProtocolLayers/HTTP/UriMapping.
2. Задайте URL доступа к внешней системе в параметре Uri в группе @libs/MessageBus/HttpExternalConnections/ConnectionsList/porttype/ProtocolLayers/HTTP/UriMapping/<имя portType>.



Рис. 89. Задание URL доступа к внешней системе

3. Задайте тип используемого кодирования в параметре MessageEncodingType группы @libs/MessageBus/HttpExternalConnections/ConnectionsList/porttype/ProtocolLayers/Tcp.

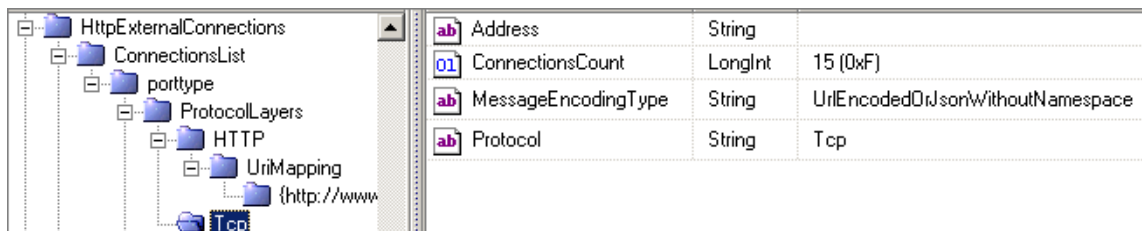


Рис. 90. Задание типа используемого кодирования

Внешняя система и Binding для отправки выбирается из таблицы маршрутизации portType @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable.

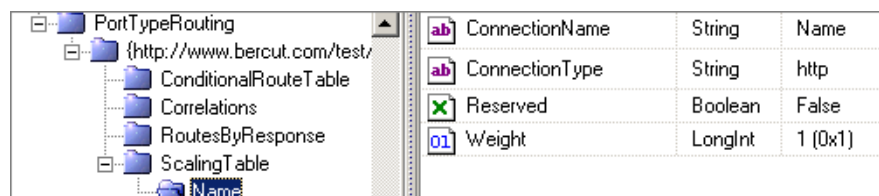


Рис. 91. Выбор внешней системы и Binding

Таблица 2. Варианты внешних систем и типов кодирования исходящего сообщения

Тип внешней системы	Тип кодирования сообщения	Тип Binding
ExternalConnections	SOAP/XML/TCP	SOAP

Тип внешней системы	Тип кодирования сообщения	Тип Binding
	(UseFastInfoSet = False)	
	SOAP/FI/TCP (UseFastInfoSet = True)	SOAP
HttpExternalConnections	SOAP/XML/HTTP (MessageEncodingType=SoapXml)	SOAP
	SOAP/FI/HTTP (MessageEncodingType=SoapFastInfoSet)	SOAP
	JSON/HTTP without namespace (MessageEncodingType=UrlEncodedOrJson WithoutNamespace)	HTTP
	JSON/HTTP with namespace (MessageEncodingType=UrlEncodedOrJson WithNamespace)	HTTP
	UrlEncoded/HTTP (MessageEncodingType=UrlEncodedOrJson WithoutNamespace) или MessageEncodingType=UrlEncodedOrJson WithNamespace)	HTTP

По таблице маршрутизации portType система выбирает внешнюю систему, в настройках которой указан тип кодирования сообщения. Если для этого portType был описан HTTP Binding, сообщение кодируется и отправляется внешней системе. Если HTTP Binding не описан, система отправляет ошибку.

**! Внимание!** Если в `HttpExternalConnections` указан тип кодирования `Json` или `UrlEncoded` и для операции не описан HTTP Binding, то попытка отправить сообщение приведет к ошибке отправки.

### 8.18.6. Структуры данных

При объявлении частей в процессе составления сообщений можно указать на выбор:

- тип части;
- XSD-элемент.

Выбор зависит от типа SOAP Binding, в котором используется сообщение. Каждый portType имеет SOAP Binding. Если он не объявлен в WSDL-описании, система автоматически формирует его при развертывании артефакта или подключении SA.

Правило отображения HTTP-описания в SOAP:

- Если в POST-, PUT- или PATCH-операциях в качестве BODY во входящем сообщении используется элемент, у каждой операции в одном portType должны быть разные элементы. При отображении этих операций в SOAP Document/Literal SOAP-идентификация операции выполняется по имени этого элемента.



- Если в сообщении в качестве BODY указывается тип, сообщение может использоваться в разных операциях. При отображении этих операций в SOAP RPC/Literal SOAP-идентификация операции выполняется по имени операции, которая закодирована в сообщении. Имя part и имя типа при этом не важны.

### 8.18.6.1. Тип данных для HTTP/UrlEncoded-, HTTP/UrlEncoded+UriReplacement-, SOAP/RPC/Encoded- и SOAP/RPC/Literal-сообщений

Для UrlEncoded параметры запроса с их наименованиями передаются в URL после символа '?':

Пример: /subscribers/rests?number={number},

где {number} — параметр.

Все параметры должны быть объявлены во входящем сообщении как отдельные части со следующими особенностями:

- Используйте примитивные типы данных. Комплексные типы не передаются в URL.
- Имя каждой части уникально. В URL невозможно передать массив значений.
- Порядок объявления частей не важен. В URL они могут следовать в любом порядке.
- Обязательность или необязательность части зависит от типа Binding:
  - Для SOAP Binding все части — обязательные. Необязательность возможна для части с указанием конкретного элемента: для данного типа указываются неидентифицируемые типы.
  - Для HTTP Binding все части — необязательные. В URL передаются только нужные данные.

В общем случае все части — обязательные.

Пример WSDL:

```
<message name="TestWsdLJsonGetUrlEncodedOperationRequestMessage">
  <part name="singleStringElement" type="xsd:string"/>
  <part name="singleStringNillableElement" type="xsd:string"/>
  <part name="arrayStringElement" type="xsd:string"/>
  <part name="singleIntElement" type="xsd:int"/>
  <part name="singleBooleanElement" type="xsd:boolean"/>
  <part name="singleDateElement" type="xsd:date"/>
  <part name="singleDateTimeElement" type="xsd:dateTime"/>
  <part name="throwFault" type="xsd:string"/>
</message>
```

### Тип данных для HTTP/UriReplacement-сообщений

UriReplacement передает данные запроса как часть URL.

Пример: /subscribers/{number}/rests,

где {number} — переменная часть URL.

Тип данных аналогичен UrlEncoded, но с дополнительными ограничениями:

- Часть URL не может быть пустой.
- URL — это адрес сервиса, его изменение инициирует вызов другого сервиса.

Пример WSDL:

```
<message name="TestWsdLJsonGetUriReplacementOperationRequestMessage">
  <part name="singleStringElement" type="xsd:string"/>
  <part name="arrayStringElement" type="xsd:string"/>
  <part name="singleIntElement" type="xsd:int"/>
  <part name="singleBooleanElement" type="xsd:boolean"/>
  <part name="singleDateElement" type="xsd:date"/>
  <part name="singleDateTimeElement" type="xsd:dateTime"/>
</message>
```

```
<part name="throwFault" type="xsd:string"/>
</message>
```

### 8.18.6.2. Тип данных для HTTP/UriReplacement-сообщений

UriReplacement передает данные запроса как часть URL.

Пример: /subscribers/{number}/rests,

где {number} — переменная часть URL.

Тип данных аналогичен UriEncoded, но с дополнительными ограничениями:

- Часть URL не может быть пустой.
- URL — это адрес сервиса. Его изменение инициирует вызов другого сервиса.

Пример WSDL:

```
<message name="TestWsdJsonGetUriReplacementOperationRequestMessage">
  <part name="singleStringElement" type="xsd:string"/>
  <part name="arrayStringElement" type="xsd:string"/>
  <part name="singleIntElement" type="xsd:int"/>
  <part name="singleBooleanElement" type="xsd:boolean"/>
  <part name="singleDateElement" type="xsd:date"/>
  <part name="singleDateTimeElement" type="xsd:dateTime"/>
  <part name="throwFault" type="xsd:string"/>
</message>
```

### 8.18.6.3. Тип данных для HTTP/JSON- и SOAP/Document/Literal-сообщений

Пример WSDL:

```
<xsd:complexType name="TestWsdJsonType">
  <xsd:sequence>
    <xsd:element name="singleStringElement" type="xsd:string"/>
    <xsd:element name="singleStringNillableElement" type="xsd:string" nillable="true"/>
    <xsd:element name="arrayStringElement" type="xsd:string" maxOccurs="unbounded"/>
    <xsd:element name="singleIntElement" type="xsd:int"/>
    <xsd:element name="singleBooleanElement" type="xsd:boolean"/>
    <xsd:element name="singleDateElement" type="xsd:date"/>
    <xsd:element name="singleDateTimeElement" type="xsd:dateTime"/>
    <xsd:element name="throwFault" type="xsd:string" minOccurs="0"/>
    <xsd:element name="callNext" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

#### Элементы

```
<xsd:element name="testWsdJsonRequest" type="tns:TestWsdJsonType"/>
<xsd:element name="testWsdJsonFault" type="tns:TestWsdJsonType"/>
```

#### Сообщения

Используется стиль Document/Literal Wrapped, а не чистый SOAP/Document/Literal стиль. Для стиля Document/Literal Wrapped в сообщении объявляется один part, который ссылается на XML-элемент, идентифицирующий вызываемую операцию.

```
<message name="TestWsdJsonPostJsonEncodedOperationRequestMessage">
  <part name="part" element="ns0:testWsdJsonPostJsonEncodedRequest"/>
</message>
<message name="TestWsdJsonOperationFaultMessage">
  <part name="part" element="ns0:testWsdJsonFault"/>
</message>
```

#### 8.18.6.4. Тип данных для HTTP/JSON+UriReplacement- и SOAP/RPC/Literal-сообщений

Для запроса JSON+UriReplacement требуется описание единой структуры данных, которые передаются при JSON-кодировании, и структуры данных, кодируемых в URL. Тип данных, которые передаются при JSON-кодировании, описан как комплексный тип. Части, которые кодируются в URL, описаны как части сообщения.

Пример WSDL:

```
<xsd:complexType name="TestWsdlJsonType">
  <xsd:sequence>
    <xsd:element name="singleStringElement" type="xsd:string"/>
    <xsd:element name="singleStringNillableElement" type="xsd:string" nillable="true"/>
  >
  <xsd:element name="arrayStringElement" type="xsd:string" maxOccurs="unbounded"/>
  <xsd:element name="singleIntElement" type="xsd:int"/>
  <xsd:element name="singleBooleanElement" type="xsd:boolean"/>
  <xsd:element name="singleDateElement" type="xsd:date"/>
  <xsd:element name="singleDateTimeElement" type="xsd:dateTime"/>
  <xsd:element name="throwFault" type="xsd:string" minOccurs="0"/>
  <xsd:element name="callNext" type="xsd:string" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
```

#### Сообщение

Для отправки запроса JSON+UriReplacement требуется multipart сообщение. Оно содержит:

- одну part комплексного типа, которая кодируется в JSON;
- произвольное количество part примитивного типа, которые используются в параметрах URL.

```
<message name="TestJsonUriReplacementRequestMessage">
  <part name="singleStringElement" type="xsd:string"/>
  <part name="json" type="tns:TestWsdlJsonType"/>
  <part name="throwFault" type="xsd:string"/>
</message>
```

#### 8.18.6.5. Числовые имена элементов для типа данных HTTP/JSON

В JSON разрешены имена элементов, которые начинаются с числа или являются числами. В XML-описании такие имена недопустимы из-за несоответствия NCName. При создании описания для существующего стороннего сервиса с такими элементами нет стандартизированного способа преодолеть это ограничение. Для описания таких элементов в XSD-описании применяется префикс «digit-element-name-». Этот префикс добавляется в начало имени каждого элемента, который начинается с числа.

Пример JSON-сообщения:

```
{
  "space": "FreeMsisdns",
  "key": "79028991265",
  "6": "7",
  "7": "1574999999"
}
```

Пример XSD-описания типа данных для сообщения:

```
<xsd:complexType name="CacheUpdateRequestType">
  <xsd:sequence>
    <xsd:element name="space" type="xsd:string"/>
    <xsd:element name="key" type="xsd:string"/>
    <xsd:element name="digit-element-name-6" type="xsd:string"/>
    <xsd:element name="digit-element-name-7" type="xsd:string"/>
  </xsd:sequence>
```

```
</xsd:complexType>
```

Для изменения префикса сервиса с конкретной реализацией определено системное свойство `com.bercut.encoder.digit.element.name.prefix`, которое задается в скрипте запуска. Изменение префикса может потребоваться, если, например, имя префикса совпало с именем существующего элемента.

Пример использования:

```
-Dcom.bercut.encoder.digit.element.name.prefix=<<json-element-name>>
```

### 8.18.6.6. Типы данных при использовании HTTP Headers

Для использования HTTP Headers в HTTP Binding необходимо описать их в сообщениях операций.

В сообщениях HTTP Headers объявляются как части примитивного типа. В сообщениях добавляются части. Если в WSDL-описании не описан SOAP Binding, система формирует его автоматически. Его тип может поменяться с 'Document/Literal' на 'RPC'. Вы можете вручную создать соответствующий Document/Literal Binding с SOAP Headers.

#### Сообщение с HTTP Headers для HTTP/UrlEncoded и HTTP/UrlEncoded +UrlReplacement

```
<message name="TestWsdLJsonGetUrlEncodedOperationRequestMessage">
  <part name="stringHeader" type="xsd:string"/>

  <part name="singleStringElement" type="xsd:string"/>
  <part name="singleStringNillableElement" type="xsd:string"/>
  <part name="arrayStringElement" type="xsd:string"/>
  <part name="singleIntElement" type="xsd:int"/>
  <part name="singleBooleanElement" type="xsd:boolean"/>
  <part name="singleDateElement" type="xsd:date"/>
  <part name="singleDateTimeElement" type="xsd:dateTime"/>
  <part name="throwFault" type="xsd:string"/>
</message>
```

#### Сообщение с HTTP Headers для HTTP/UrlReplacement

```
<message name="TestWsdLJsonGetUrlReplacementOperationRequestMessage">
  <part name="stringHeader" type="xsd:string"/>

  <part name="singleStringElement" type="xsd:string"/>
  <part name="arrayStringElement" type="xsd:string"/>
  <part name="singleIntElement" type="xsd:int"/>
  <part name="singleBooleanElement" type="xsd:boolean"/>
  <part name="singleDateElement" type="xsd:date"/>
  <part name="singleDateTimeElement" type="xsd:dateTime"/>
  <part name="throwFault" type="xsd:string"/>
</message>
```

#### Сообщение с HTTP Headers для HTTP/JSON

```
<message name="TestWsdLJsonPostJsonEncodedOperationRequestMessage">
  <part name="stringHeader" element="ns0:stringHeader"/>

  <part name="part" element="ns0:testWsdLJsonPostJsonEncodedRequest"/>
</message>
```

#### Сообщение с HTTP Headers для HTTP/JSON+UrlReplacement

```
<message name="TestJsonUrlReplacementRequestMessage">
  <part name="stringHeader" type="xsd:string"/>
```

```
<part name="singleStringElement" type="xsd:string"/>
<part name="json" type="tns:TestWsdJsonType"/>
<part name="throwFault" type="xsd:string"/>
</message>
```

### 8.18.6.7. Неструктурированный текст AnyData

В SLES можно передавать:

- произвольные JSON-данные в запросах и ответах;
- неструктурированный текст в ответах.

#### Произвольные JSON-данные в запросах и ответах

Для передачи произвольных данных в sequence необходимо добавить элемент:

```
<xsd:any processContents="lax" namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
```

**i Примечание.** Этот элемент можно добавлять только в пустые sequence.

#### Неструктурированный текст в ответах

Неструктурированный текст можно передавать только в ответах и faults. Весь ответ состоит из этого текста, другой информации тело не содержит. Для неструктурированного текста используйте в message part следующий тип данных:

```
<xsd:complexType name="AutogeneratedEmptySequenceType">
  <xsd:sequence>
    <xsd:any processContents="lax" namespace="##any" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation xml:lang="en">any value</xsd:documentation>
      </xsd:annotation>
    </xsd:any>
  </xsd:sequence>
  <xsd:attribute use="optional" name="contentType" type="xsd:string"/>
</xsd:complexType>
```

Пример объявления исходящего сообщения:

```
<wsdl:message name="YourOutputMessage">
  <wsdl:part name="emptySequencePart" type="ns1:AutogeneratedEmptySequenceType"/>
</wsdl:message>
```

## 8.18.7. Примеры сообщений

### HTTP GET UriEncoded

```
GET http://198.51.100.21:7070/test/wsdl/TestWsdJson/TestWsdJsonGetUriEncodedPortType/
TestWsdJsonGetUriEncodedOperation?singleStringElement=1&singleStringNillableElement=
&arrayStringElement=2%2C3&singleIntElement=4&singleBooleanElement=true&
singleDateElement=2002-09-24%2B06%3A00&singleDateTimeElement=2008-09-04T21%3A10%3A36&
throwFault=false HTTP/1.1
Accept-Encoding: gzip,deflate
Host: 198.51.100.21:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

### HTTP GET UriReplacement

```
GET http://198.51.100.21:7070/test/wsdl/TestWsdJson/
TestWsdJsonGetUriReplacementPortType/TestWsdJsonGetUriReplacementOperation/1/2%2C3/4/
true/2002-09-24%2B06%3A00/2008-09-04T21%3A10%3A36/false HTTP/1.1
Accept-Encoding: gzip,deflate
Host: 192.168.11.105:7070
Connection: Keep-Alive
```

```
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

## HTTP GET UriEncoded+UriReplacement

```
GET /test/wsdl/TestWsdJson/TestUriEncodedAndReplacementPortType/
TestUriEncodedAndReplacementOperation/string/25/throwFault/
false?arrayStringElement=2,3&singleIntElement=
4&singleBooleanElement=true&singleDateElement=2002-09-24&singleDateTimeElement=2008-09-04T21:10:36
Connection: Keep-Alive
Host: 198.51.100.11:7070
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Часть URL, которая соответствует location, объявленному в HTTP Binding: TestUriEncodedAndReplacementOperation/string/(singleStringElement)/throwFault/(throwFault).

## HTTP POST without Namespace

```
POST http://198.51.100.11:7070/test/wsdl/TestWsdJson/
TestWsdJsonPostJsonEncodedPortType/TestWsdJsonPostJsonEncodedOperation HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/json
Content-Length: 291
Host: 198.51.100.21:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

```
{
  "singleStringElement" : "1",
  "singleStringNillableElement" : "",
  "arrayStringElement" : [
    "2",
    "3"
  ],
  "singleIntElement" : "4",
  "singleBooleanElement" : "true",
  "singleDateElement" : "2002-09-24+06:00",
  "singleDateTimeElement" : "2008-09-04T21:10:36",
  "throwFault" : "false"
}
```

## HTTP POST with Namespace

```
POST http://198.51.100.11:7070/test/wsdl/TestWsdJson/
TestWsdJsonPostJsonEncodedPortType/TestWsdJsonPostJsonEncodedOperation HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/json
Content-Length: 400
Host: 198.51.100.21:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

```
{
  "@context": {
    "tes": "http://www.bercut.com/test/schema/TestWsdJson"
  },
  "tes:singleStringElement" : "1",
  "tes:singleStringNillableElement" : "",
  "tes:arrayStringElement" : [
    "2",
    "3"
  ],
  "tes:singleIntElement" : "4",
  "tes:singleBooleanElement" : "true",
  "tes:singleDateElement" : "2002-09-24+06:00",
  "tes:singleDateTimeElement" : "2008-09-04T21:10:36",
  "tes:throwFault" : "false"
}
```

## JSON с числовыми именами элементов

**RTSIB-представление заполненного сообщения:**

```
parameters [CacheUpdateRequestType] =
  space [string] = FreeMsisdns
  key [string] = 79028991265
  digit-element-name-6 [string] = 7
  digit-element-name-7 [string] = 1574999999'
```

**Сообщение, закодированное в JSON:**

```
POST http://198.51.100.21:7070 /v1/cache/update HTTP/1.1
Connection: Keep-Alive
Host: 198.51.100.21:8088
Content-Length: 95
Content-Type: application/json
SOAPAction: http://www.bercut.com/wsdl/t2msisdncacherestapi/T2MsisdnCache/Update
MessageId: 8

{
  "space": "FreeMsisdns",
  "key": "79028991265",
  "6": "7",
  "7": "1574999999"
}
```

**JSON+UrlReplacement**

```
POST /test/wsdl/TestWsdJson/TestJsonUrlReplacementPortType/onepart/string/25/
datetime/2008-09-25T21:10:36
Connection: Keep-Alive
Host: 198.51.100.21:7070
Content-Length: 376
Content-Type: application/json
{
  "singleStringElement" : "1",
  "singleStringNillableElement" : "",
  "arrayStringElement" : [
    "2",
    "3"
  ],
  "singleIntElement" : "4",
  "singleBooleanElement" : "true",
  "singleDateElement" : "2002-09-24+06:00",
  "singleDateTimeElement" : "2008-09-04T21:10:36",
  "throwFault" : "false"
}
```

Часть URL, которая соответствует location, объявленному в HTTP Binding: onepart/string/(singleStringPart)/datetime/(singleDateTimePart).

**HTTP OK RESPONSE**

```
HTTP/1.1 200 OK
Content-Length: 445
Content-Type: application/json
MessageId: 2

{
  "@context": {
    "ns0": "http://www.bercut.com/test/schema/TestWsdJson"
  },
  "ns0:singleStringElement": "1",
  "ns0:singleStringNillableElement": "",
  "ns0:arrayStringElement": [
    "2",
    "3"
  ],
  "ns0:singleIntElement": "4",
  "ns0:singleBooleanElement": "true",
  "ns0:singleDateElement": "2002-09-24+06:00",
  "ns0:singleDateTimeElement": "2008-09-04T21:10:36",
  "ns0:throwFault": "false"
}
```

}

## HTTP USER FAULT RESPONSE

```

HTTP/1.1 501 FAULT
Content-Length: 313
Content-Type: application/json
MessageId: 3

{
  "singleStringElement": "1",
  "singleStringNillableElement": "",
  "arrayStringElement": [
    "2,3"
  ],
  "singleIntElement": "4",
  "singleBooleanElement": "true",
  "singleDateElement": "2002-09-24+06:00",
  "singleDateTimeElement": "2008-09-04T21:10:36",
  "throwFault": "Fault"
}

```

## HTTP SYSTEM FAULT RESPONSE

```

HTTP/1.1 500 FAULT
Content-Length: 15880
Content-Type: application/json
MessageId: 4

{
  "@context": {
    "ns0": "http://schemas.xmlsoap.org/soap/envelope/",
    "ns1": "http://jax-ws.dev.java.net/"
  },
  "ns1:faultcode": "SOAP-ENV:Server",
  "ns1:faultstring": null,
  "ns1:detail": {
    "ns1:exception": {
      "@class": "com.bercut.sdep.sles.bpelengine.exceptions.standart.MissingReply",
      "ns1:message": null,
      "ns1:stackTrace": {
        "ns1:frame": [
          {
            "@class": "com.bercut.sdep.sles.bpelengine.core.ServerPartnerLink
$OperationInfo",
            "@file": "ServerPartnerLink.java",
            "@method": "clear",
            "@line": "195"
          },
          {
            "@class": "com.bercut.sdep.sles.bpelengine.core.ServerPartnerLink
$OperationInfo",
            "@file": "ServerPartnerLink.java",
            "@method": "clear",
            "@line": "186"
          },
          ...
          {
            "@class": "com.bercut.sdep.sles.bpelengine.core.BPELWorkingThread",
            "@file": "BPELWorkingThread.java",
            "@method": "run",
            "@line": "119"
          }
        ]
      }
    }
  }
}

```

## HTTP EMPTY RESPONSE



Для любого нормального ответа или fault может быть объявлен пустой ответ:

```
<xsd:complexType name="EmptyType">
  <xsd:sequence/>
</xsd:complexType>

<message name="EmptyFaultMessage">
  <part name="part" element="ns0:emptyFault"/>
</message>
```

В данной реализации HTTP все конструкции имеют SOAP-представление, а SOAP-представление не может иметь пустого ответа. Ответы кодируются в JSON, в ответ возвращается пустой JSON «{}».

```
503 FAULT
Content-Length: 4
Content-Type: application/json

{}
```

## 8.19. REST-представление с использованием OpenAPI

Поддержка запросов и ответов REST\JSON в RTSIB с использованием реализации OpenAPI на языке программирования C++ выполнена с помощью набора следующих компонентов:

- OpenAPI Runtime — поддерживает время выполнения REST/JSON: кодирование и декодирование параметров, передаваемых в URI, query, Headers, Body.
- OpenAPI Parser — осуществляет разбор OpenAPI-описаний и заполняет модель типов данных и протоколов.
- OpenAPI Repository — управляет OpenAPI-описаниями:
  - загружает описания;
  - перезагружает описания;
  - сравнивает совместимость различных версий описаний;
  - генерирует *протокол* для взаимодействия в рамках RTSIB.
- *RTSIB C++ SDK* — взаимодействует с REST-реализациями. Вызывает внешние REST-сервисы и публикует реализации REST-сервисов.

Для работы репозитория OpenAPI Repository выполните необходимые *настройки*.

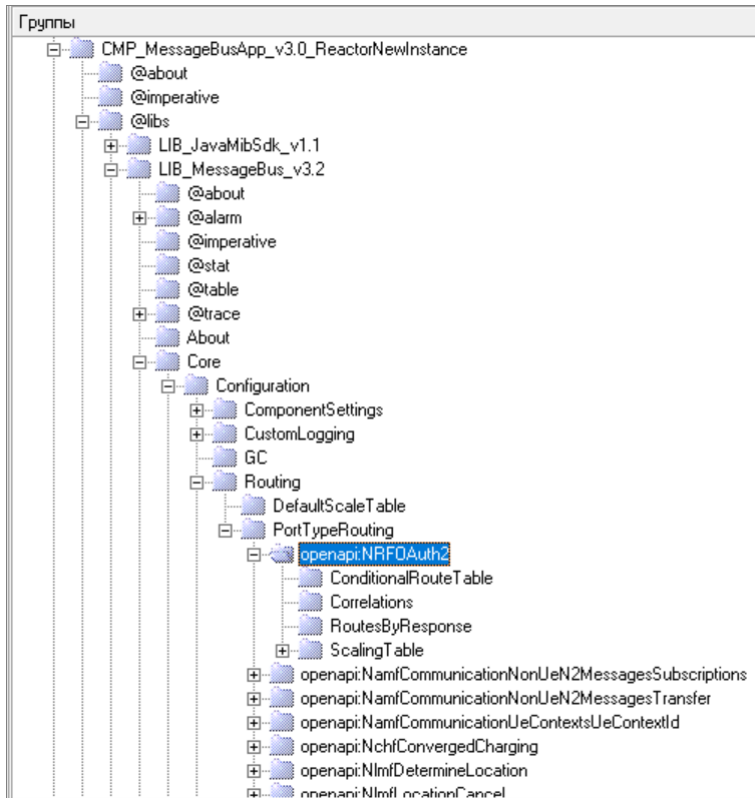
### 8.19.1. Интеграция RTSIB C++ SDK с OpenAPI Repository

RTSIB C++ SDK является стандартной библиотекой с интерфейсом C++, которая динамически подгружается. Для взаимодействия с сервисами RTSIB C++ SDK интегрирован с OpenAPI Repository. Для использования клиентских и серверных компонентов OpenAPI в RTSIB имя PortType формируется следующим образом:

*орепари*: «Имя протокола OpenAPI».

*Задайте имя протокола OpenAPI* в группе <Имя компонента>/Configuration/Repositories/OpenApiRepository/Configuration/Protocols/<Имя\_протокола>.

*Настройте маршрутизацию* вызовов клиентских сервисов в RTSIB C++ SDK в группе @libs/MessageBus/Core/Configuration/Router/PortTypeRouting/.



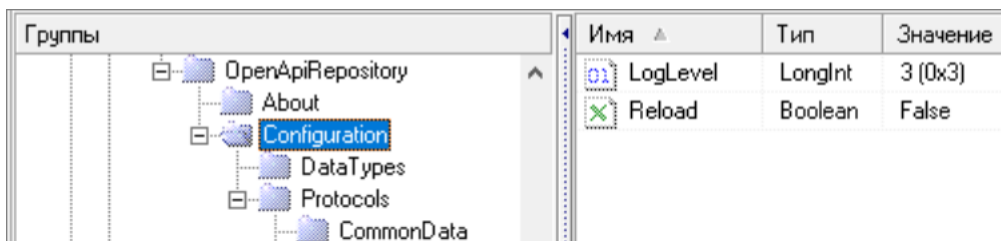
**Рис. 92. Пример настройки маршрутизации для вызовов клиентских сервисов**

**Примечание.** Для OpenAPI ключом в таблице маршрутизации является не имя PortType, а значение openapi: «Имя протокола OpenAPI».

### 8.19.2. Настройки OpenAPI Repository

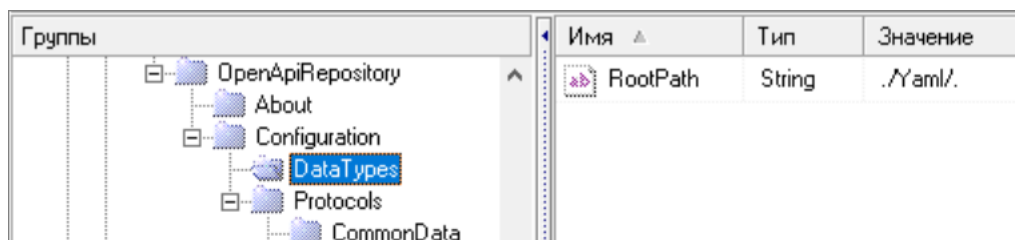
Настройки репозитория OpenAPI Repository расположены в группе <Имя компонента>/Configuration/Repositories/OpenApiRepository/. Задайте значения следующих параметров:

1. В группе <Имя компонента>/Configuration/Repositories/OpenApiRepository/Configuration/ задайте MIB-параметр *Reload* — признак перечитывания всех файлов OpenAPI-описаний и пересоздания всех протоколов. Значение по умолчанию: 'false'.



**Рис. 93. Вариант настройки MIB-параметра Reload**

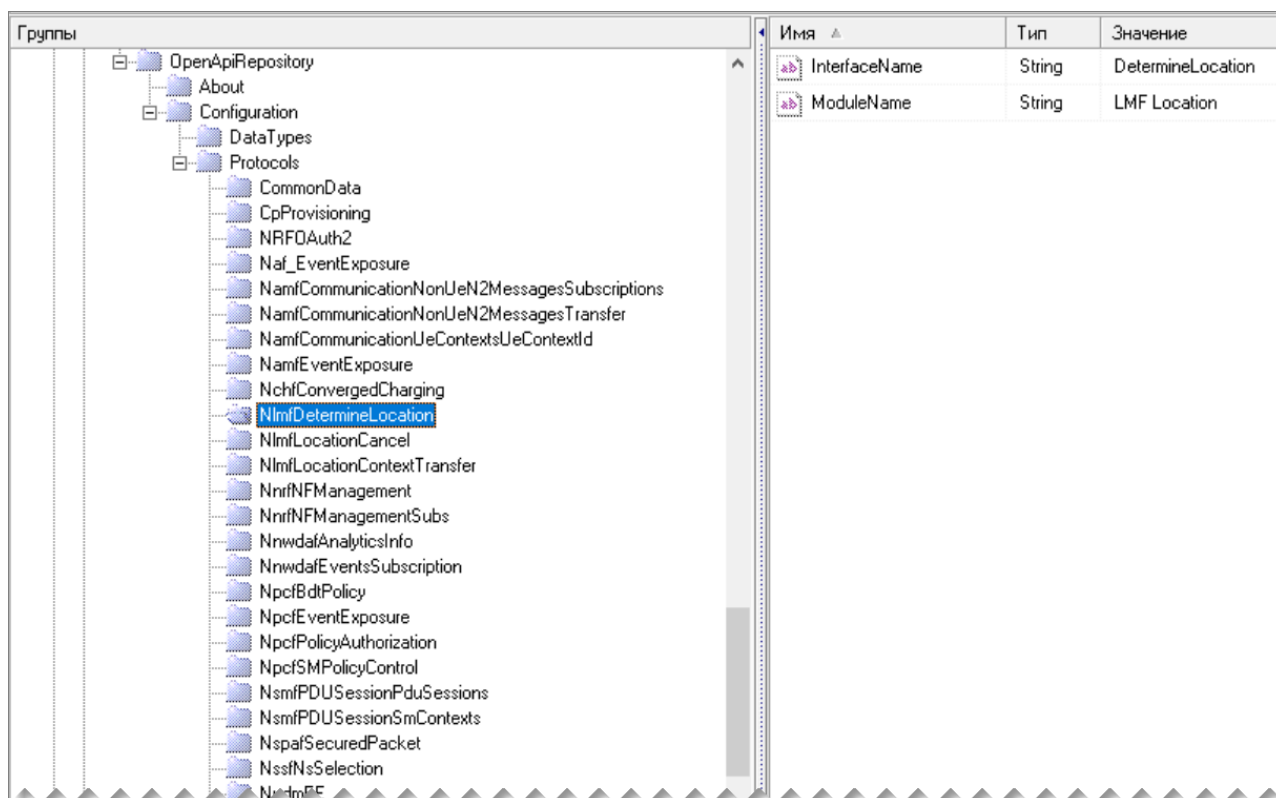
2. В группе <Имя компонента>/Configuration/Repositories/OpenApiRepository/Configuration/DataTypes/ задайте MIB-параметр *RootPath* — путь на диске, где находятся OpenAPI-описания. Описания определяются по расширению для OpenAPI описаний — YAML. Система считывает описания в указанной папке и во всех подпапках. Значение по умолчанию: пустая строка.



**Рис. 94. Вариант настройки MIB-параметра RootPath**

3. В группе <Имя компонента>/Configuration/Repositories/OpenApiRepository/Configuration/Protocols/ для перечисленных в группе протоколов:

- 3.1. MIB-параметр *ModuleName* — имя модуля OpenAPI-описаний, на основе которого строится протокол. Имя модуля для OpenAPI-описания — параметр info/title. Значение по умолчанию: пустая строка.
- 3.2. MIB-параметр *InterfaceName* — имя интерфейса из модуля OpenAPI-описаний, на основе которого строится протокол. Необязательный параметр — используется, когда в модуле описано более одного интерфейса. Каждый параметр Path в OpenAPI-описании является интерфейсом. Значение по умолчанию: пустая строка.



**Рис. 95. Вариант настройки MIB-параметров для протокола NlmfDetermineLocation**

### 8.19.3. Описание протокола для взаимодействия с RTSIB

Для каждого интерфейса генерируется протокол следующего вида:

- Interface {SEQUENCE} (Path)
  - extensions {SEQUENCE} (Системные хидеры)
    - ReplyTo {SEQUENCE}
    - From {SEQUENCE}
    - FaultTo {SEQUENCE}
    - To {STRING}
    - Action {STRING}
    - Container {STRING}
  - data {SEQUENCE} (Пользовательские данные)
    - Operation1 (OperationId или Path+Method)
      - request {SEQUENCE} (Параметры запроса)
        - URI parameter1
        - URI parameterN
        - Query parameter1
        - Query parameterN
        - Header parameter1
        - Header parameterN
        - Body
      - response1 {SEQUENCE} (Параметры ответа 2XX(1))
        - Header parameter1
        - Header parameterN
        - Body
      - responseN {SEQUENCE} (Параметры ответа 2XX(N))
        - Header parameter1
        - Header parameterN
        - Body
      - systemFault {SEQUENCE} (Системная ошибка)
      - fault1 (responses с кодом ответа отличные от 2XX)
      - faultN (responses с кодом ответа отличные от 2XX)
    - OperationN (OperationId или Path+Method)
      - request {SEQUENCE} (Параметры запроса)
        - URI parameter1
        - URI parameterN
        - Query parameter1
        - Query parameterN
        - Header parameter1
        - Header parameterN
        - Body
      - response1 {SEQUENCE} (Параметры ответа 2XX(1))
        - Header parameter1
        - Header parameterN
        - Body
      - responseN {SEQUENCE} (Параметры ответа 2XX(N))
        - Header parameter1
        - Header parameterN
        - Body
      - systemFault {SEQUENCE} (Системная ошибка)
      - fault1 (responses с кодом ответа отличные от 2XX)
      - faultN (responses с кодом ответа отличные от 2XX)

**Рис. 96. Протокол взаимодействия**

## 8.20. Статистические параметры

Система ATLAS собирает статистические данные по работе RTSIB.

Система формирует статистику:

- по входящим сообщениям;
- по исходящим сообщениям.

### Статистика по входящим сообщениям

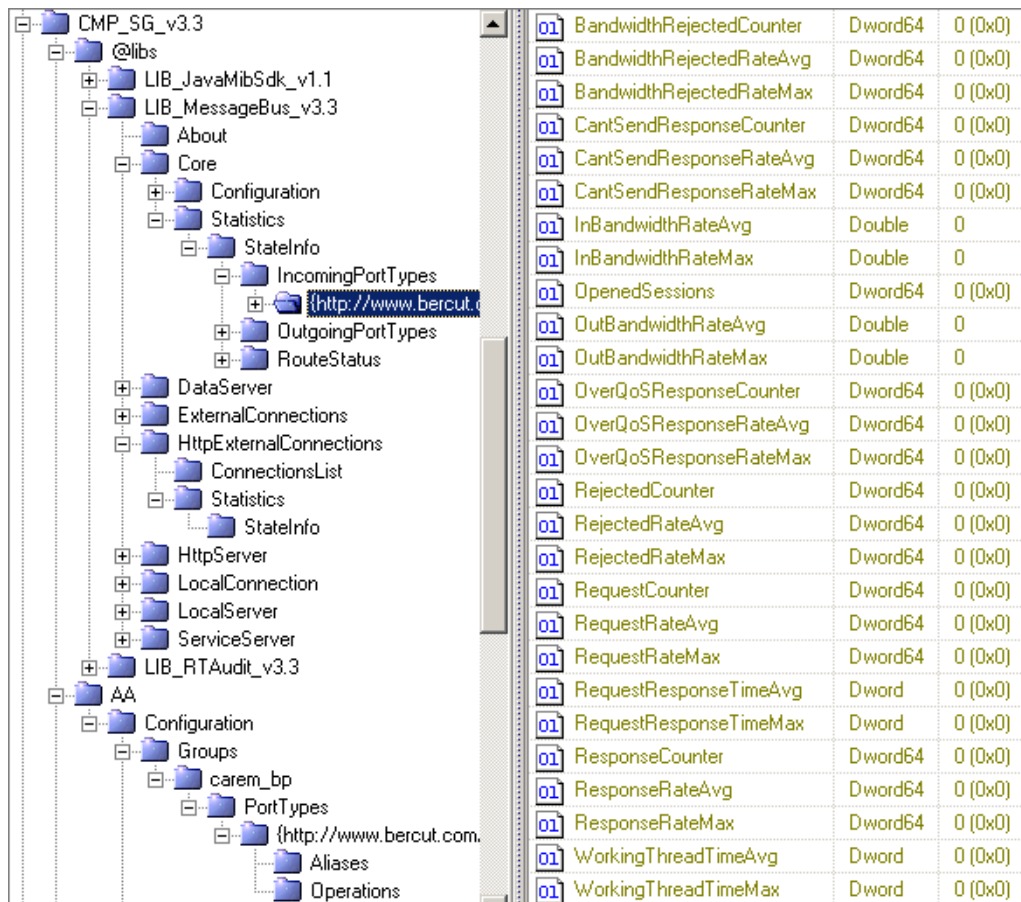
Система собирает статистику в следующих MIB-группах:

- @libs/MessageBus/Core/Statistics/StateInfo/IncomingPortTypes/<имя portType> — по входящему portType;
- @libs/MessageBus/Core/Statistics/StateInfo/IncomingPortTypes/<имя portType>/Operations/<имя операции> — по операции входящих portType;
- @libs/MessageBus/HttpServer/Statistics/StateInfo — по всем входящим сообщениям на HTTP/HTTPS-сервере;
- @libs/MessageBus/HttpServer/Statistics/StateInfo/IncomingPortTypes/<имя portType> — по входящему portType на HTTP/HTTPS-сервере;
- @libs/MessageBus/HttpServer/Statistics/StateInfo/IncomingPortTypes/<имя portType>/Operations/<имя операции> — по операции входящего portType на HTTP/HTTPS-сервере;
- @libs/MessageBus/ExternalConnections/Statistics/StateInfo.

Статистические переменные по входящим сообщениям:

- *BandwidthRejectedCounter*. Общее количество отклоненных запросов по коду '509'.
- *BandwidthRejectedRateAvg*. Среднее количество отклоненных запросов по коду '509' в секунду.
- *BandwidthRejectedRateMax*. Максимальное количество отклоненных запросов по коду '509' в секунду.
- *CantSendResponseCounter*. Общее количество не отправленных ответов.
- *CantSendResponseRateAvg*. Среднее количество не отправленных ответов за секунду.
- *CantSendResponseRateMax*. Максимальное количество не отправленных ответов за секунду.
- *InBandwidthRateAvg*. Средняя входная пропускная способность за секунду, в мегабитах.
- *InBandwidthRateMax*. Максимальная входная пропускная способность за секунду, в мегабитах.
- *OutBandwidthRateAvg*. Средняя выходная пропускная способность за секунду, в мегабитах.
- *OutBandwidthRateMax*. Максимальная выходная пропускная способность за секунду, в мегабитах.
- *OverQoSResponseCounter*. Общее количество отправленных ответов, превышающих значение ExpectedQoSTimeout.
- *OverQoSResponseRateAvg*. Среднее количество отправленных за секунду ответов, превышающих значение ExpectedQoSTimeout.
- *OverQoSResponseRateMax*. Максимальное количество отправленных за секунду ответов, превышающих значение ExpectedQoSTimeout.
- *RejectedCounter*. Общее количество отклоненных запросов.
- *RejectedRateAvg*. Среднее количество отклоненных запросов в секунду.
- *RejectedRateMax*. Максимальное количество отклоненных запросов в секунду.
- *RequestCounter*. Общее количество входящих запросов.

- *RequestRateAvg*. Среднее количество входящих запросов.
- *RequestRateMax*. Максимальное количество входящих запросов.
- *RequestResponseTimeAvg*. Среднее время обработки запроса, в миллисекундах.
- *RequestResponseTimeMax*. Максимальное время обработки запроса, в миллисекундах.
- *ResponseCounter*. Общее количество ответов.
- *ResponseRateAvg*. Среднее количество ответов в секунду.
- *ResponseRateMax*. Максимальное количество ответов в секунду.
- *WorkingThreadTimeAvg*. Среднее время обработки запроса логикой, в миллисекундах.
- *WorkingThreadTimeMax*. Максимальное время обработки запроса логикой, в миллисекундах.



**Рис. 97. Статистика по входящим сообщениям**

### Статистика по исходящим сообщениям

Система собирает статистику в следующих MIB-группах:

- @libs/MessageBus/Core/Statistics/StateInfo/OutgoingPortTypes/<имя portType> — по исходящему portType;
- @libs/MessageBus/Core/Statistics/StateInfo/OutgoingPortTypes/<имя portType>/Operations/<имя операции> — по операции исходящих portType;
- @libs/MessageBus/ExternalConnection/Statistics/StateInfo — по всем исходящим RTSIB-подключениям;
- @libs/MessageBus/ExternalConnection/Statistics/StateInfo/OutgoingPortTypes/<имя portType> — по исходящему portType RTSIB-подключений;

- @libs/MessageBus/ExternalConnection/Statistics/StateInfo/OutgoingPortTypes/<имя portType>/Operations/<имя операции> — по операции исходящего portType RTSIB-подключений;
- @libs/MessageBus/ExternalConnection/ConnectionList/<имя подключения>/Statistics/StateInfo/OutgoingPortTypes/<имя portType> — по исходящему portType для выбранного RTSIB-соединения;
- @libs/MessageBus/ExternalConnection/ConnectionList/<имя подключения>/Statistics/StateInfo/OutgoingPortTypes/<имя portType>/Operations/<имя операции> — по операции исходящего portType для выбранного RTSIB-соединения;
- @libs/MessageBus/HttpExternalConnection/Statistics/StateInfo — по всем исходящим HTTP/HTTPS-соединениям;
- @libs/MessageBus/HttpExternalConnection/Statistics/StateInfo/OutgoingPortTypes/<имя portType> — по исходящему portType HTTP/HTTPS-соединений;
- @libs/MessageBus/HttpExternalConnection/Statistics/StateInfo/OutgoingPortTypes/<имя portType>/Operations/<имя операции> — по операции исходящего portType HTTP/HTTPS-соединений;
- @libs/MessageBus/HttpExternalConnection/ConnectionList/<имя подключения>/Statistics/StateInfo — по выбранному исходящему HTTP/HTTPS-соединению;
- @libs/MessageBus/HttpExternalConnection/ConnectionList/<имя подключения>/Statistics/StateInfo/OutgoingPortTypes/<имя portType> — по исходящему portType для выбранного HTTP/HTTPS-соединения;
- @libs/MessageBus/HttpExternalConnection/ConnectionList/<имя подключения>/Statistics/StateInfo/OutgoingPortTypes/<имя portType>/Operations/<имя операции> — по операции исходящего portType для выбранного HTTP/HTTPS-соединения.

Статистические переменные по исходящим сообщениям:

- *InBandwidthRateAvg*. Средняя входная пропускная способность за секунду, в мегабитах.
- *InBandwidthRateMax*. Максимальная входная пропускная способность за секунду, в мегабитах.
- *OutBandwidthRateAvg*. Средняя выходная пропускная способность за секунду, в мегабитах.
- *OutBandwidthRateMax*. Максимальная выходная пропускная способность за секунду, в мегабитах.
- *OverQoSResponseCounter*. Общее количество полученных ответов, превышающих значение ExpectedQoSTimeout.
- *OverQoSResponseRateAvg*. Среднее количество полученных за секунду ответов, превышающих значение ExpectedQoSTimeout.
- *OverQoSResponseRateMax*. Максимальное количество полученных за секунду ответов, превышающих значение ExpectedQoSTimeout.
- *QueryCounter*. Общее количество отправленных запросов.
- *QueryRateAvg*. Среднее количество отправленных запросов в секунду.
- *QueryRateMax*. Максимальное количество отправленных запросов в секунду.
- *QueryResponseTimeAvg*. Среднее время ожидания ответа на запрос, в миллисекундах.
- *QueryResponseTimeMax*. Максимальное время ожидания ответа на запрос, в миллисекундах.
- *RejectedCounter*. Общее количество отклоненных или потерянных запросов.
- *RejectedRateAvg*. Среднее количество отклоненных или потерянных запросов в секунду.
- *RejectedRateMax*. Максимальное количество отклоненных или потерянных запросов в секунду.
- *ResponseCounter*. Общее количество ответов.

- *ResponseRateAvg*. Среднее количество ответов в секунду.
- *ResponseRateMax*. Максимальное количество ответов в секунду.
- *TimeoutCounter*. Общее количество тайм-аутов на запросы.
- *TimeoutRateAvg*. Среднее количество тайм-аутов на запросы в секунду.
- *TimeoutRateMax*. Максимальное количество тайм-аутов на запросы в секунду.

01	InBandwidthRateAvg	Double	0
01	InBandwidthRateMax	Double	0
01	OpenedSessions	Dword64	0 (0x0)
01	OutBandwidthRateAvg	Double	0
01	OutBandwidthRateMax	Double	0
01	OverQoSResponseCounter	Dword64	0 (0x0)
01	OverQoSResponseRateAvg	Dword64	0 (0x0)
01	OverQoSResponseRateMax	Dword64	0 (0x0)
01	QueryCounter	Dword64	0 (0x0)
01	QueryRateAvg	Dword64	0 (0x0)
01	QueryRateMax	Dword64	0 (0x0)
01	QueryResponseTimeAvg	Dword	0 (0x0)
01	QueryResponseTimeMax	Dword	0 (0x0)
01	RejectedCounter	Dword64	0 (0x0)
01	RejectedRateAvg	Dword64	0 (0x0)
01	RejectedRateMax	Dword64	0 (0x0)
01	ResponseCounter	Dword64	0 (0x0)
01	ResponseRateAvg	Dword64	0 (0x0)
01	ResponseRateMax	Dword64	0 (0x0)
01	TimeoutCounter	Dword64	0 (0x0)
01	TimeoutRateAvg	Dword64	0 (0x0)
01	TimeoutRateMax	Dword64	0 (0x0)

**Рис. 98. Статистика по исходящим сообщениям**



## 9. Система SLES

Система **SLES** выполняет бизнес-процессы (BPEL-сценарии) и вспомогательные сервисы (Java-сценарии).

### 9.1. Установка и запуск

Для установки системы SLES используйте файл `startup.sh`.

Порядок установки системы SLES:

1. Переключитесь на пользователя `bercut`. Все последующие действия выполняйте от имени этого пользователя.
2. Создайте каталог установки системы SLES, именуемый дальше `$SLES`.
3. Скопируйте каталог дистрибутива в созданный каталог.
4. Распакуйте архив установочного дистрибутива.
5. Установите права на запуск скрипт-файлов в каталоге `bin` для всех пользователей.

```
cd $SLES
```

```
chmod +x *.sh
```

6. Выполните команду:

```
./startup.sh
```

Система автоматически создаст дерево в приложении ATLAS MIB Explorer.

**Примечание.** При последующих запусках и остановках системы SLES используйте на выбор:

- **SSM**.
- команду `./startup.sh` или `./shutdown.sh`.

7. Настройте **порты приложений или взаимодействие** между компонентами.
8. Проверьте трейс в приложении ATLAS MIB Explorer на отсутствие ошибок от системы SLES.

Проверьте в MIB, что SLES доступен. Параметр `Status` группы `CMP_SLES_vX.Y/Core/Statistics` должен иметь значение `available`.

После запуска системы SLES приступайте к **развертыванию** бизнес-процессов.

Чтобы задать специальное имя для экземпляра SLES в скрипт-файле `options-usr.sh.template`, расположенном в каталоге `bin`, укажите MIB-имя в параметре `APP_NAME`. Переименуйте этот файл в `options-usr.sh` и **отредактируйте** его.

Чтобы удалить созданный резервный файл после **миграции** в скрипт-файле `options-usr.sh`, расположенном в каталоге `bin`, присвойте значение `True` параметру `ATLAS_BACKUP_REMOVE`.

Для остановки системы SLES выполните команду:

```
./shutdown.sh
```

**i Примечание.** Вы можете воспользоваться упрощенной настройкой Java-опций. Используйте файл `options-usr.sh.template` в каталоге `bin`. Переименуйте этот файл в `options-usr.sh` и [отредактируйте](#) его. Заданные настройки применятся перед формированием опций запуска Tomcat.

## 9.2. Развертывание и переустановка BP и SE

Для развертывания и переустановки бизнес-процессов (BP) и сервисных драйверов (SE) используйте файл `deployer.sh`.

Проверьте наличие:

- установленной и рабочей системы SLES;
- установленной системы менеджмента, в том числе приложения ATLAS MIB Explorer;
- JAR-файлов развертываемых BP и SE.

Порядок развертывания BP и SE:

1. Перейдите в каталог `$SLES/bin`.

```
cd $SLES/bin
```

2. Выполните команду:

```
./deployer.sh -u <url> -c <command> -j <path>/<filename> -a <name> -s -w
```

Где:

### **-u <url>**

URL-адрес, по которому система SLES принимает команды на установку приложений. Формат: `http://<IP-address>:<Port>`. Обязательный параметр. По умолчанию значение параметра: `http://localhost:3344/deployer`. Администратор может [изменить](#) URL-адрес.

### **-c <command>**

Выполняемая операция:

***deploy*** — установка BP (SE).

***undeploy*** — удаление BP (SE).

***undeploy\_all*** — удаление всех BP и SE, развернутых на данном SLES.

***redploy*** — переустановка BP (SE). Новая версия BP (SE) не должна иметь изменений в XSD- и WSDL-описаниях. Иначе последовательно используйте команды ***undeploy*** и ***deploy***.

Обязательный параметр.

### **-j <path>/<filename>**

**<path>**. Путь — относительный или абсолютный — к JAR-файлу BP (SE). Обязательный параметр при развертывании или переустановке бизнес-процесса: команды ***deploy*** или ***redploy***. Для команды ***undeploy*** вы можете указать '\*', тогда будут удалены все BP и SE.

**<filename>**. Имя JAR-файлу BP (SE). Обязательный параметр при развертывании или переустановке бизнес-процесса: команды ***deploy*** или ***redploy***.

### **-a <name>**

Имя удаляемого BP (SE). Имя приложения, развернутого с помощью скрипта `slesDeployer.sh`. Совпадает с именем JAR-файла приложения

без расширения `.jar`. Имя приложения вы можете увидеть в трейсе при его развертывании: `deploy service '<имя_приложения>' successfully finished, deploy info = '...'`.

**-s**

Флаг прерывания выполнения групповых команд `deploy` или `undeploy` при первой ошибке.

**-w**

Флаг завершения работы всех экземпляров BP (SE) при выполнении команды: `undeploy`. Скрипт-файл продолжает выполняться, пока не будет завершена работа BP (SE).

**-f <name>**

Имя группы MIB, если оно отличается от имени по умолчанию.

Например:

```
./deployer.sh -u http://127.0.0.1:3344 -c deploy -j /opt/deployable/
BP_GetBalance_v1.jar
```

После выполнения операции система выведет в консоль сообщение: `Deploying <имя JAR-файла> is successfully finished`. В трейс системы SLES также выводится соответствующее сообщение.

- Повторите пункты 3-5 для всех бизнес-процессов и сервисных драйверов. Для автоматизации развертывания бизнес-процессов и сервисных драйверов используйте скрипт-файл:

```
export set DIRLIBS=/<путь к каталогу с JAR-файлами BP и SE>/*.jar
for i in ${DIRLIBS}
do
echo $i
deployer.sh -u http://<URL-адрес deployer> -c deploy -j $i
done
```

Скрипт-файл развертывает все BP и SE, которые находятся в каталоге `<путь к директории с JAR-файлами BP и SE>`.

После установки BP или SE в папке `$SLES/deployed/apps` система создает каталог с именем развернутого JAR-файла. Ошибки, произошедшие в процессе развертывания BP или SE, система сохраняет в каталоге `$SLES/bad.namespaces`.

- Для переустановки BP и SE используйте команду:

```
./deployer.sh -c redeploy -j <путь к JAR-файлу>/<имя JAR-файла>
```

Например: `./deployer.sh -c redeploy -j /opt/deployable/BP_BalanceManagement_v8.0.jar`

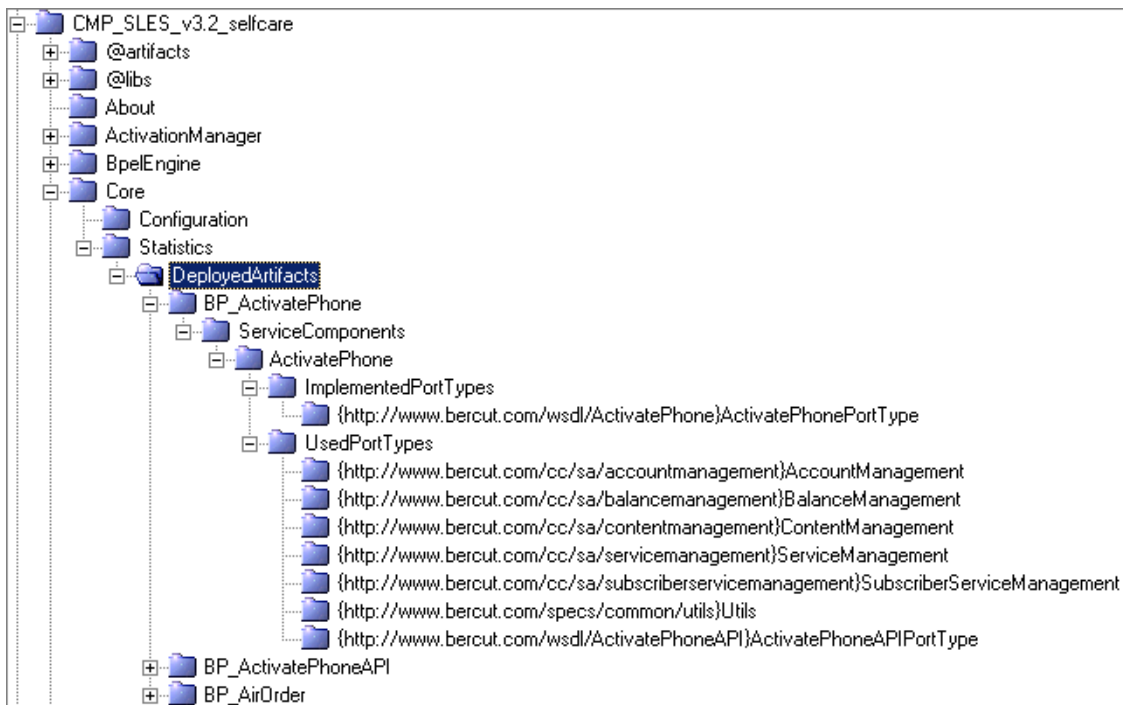
- Для удаления BP и SE используйте команду на выбор:

- `./deployer.sh -c undeploy -a <имя приложения>`
- `./deployer.sh -c undeploy -j <путь к JAR-файлу>/<имя JAR-файла>`

Например:

```
./deployer.sh -c undeploy -a BP_BalanceManagement_v8.0
./deployer.sh -c undeploy -j /opt/deployable/
BP_BalanceManagement_v8.0.jar
```

После развертывания бизнес-процесса в группе `CMP_SLES_vX.Y/Core/Statistics/DeployedArtifacts/` система добавляет группу с именем `<имя BP или SE>`.



**Рис. 99. Отображение в MIB статистики развернутого BP**

В группу CMP\_SLES\_vX.Y/Core/Statistics/DeployedArtifacts/<имя BP или SE>/ServiceComponents/<имя бизнес-процесса>/ImplementedPortTypes/ система добавляет соединение с именем {<targetNameSpace><portType name>}.

Информацию о корректности работы BP или SE смотрите в переменной *Status* группы CMP\_SLES\_vX.Y/@artifacts/<имя JAR-файла>/Statistics. При корректной работе значение **available**.

При пакетной установке новых версий BP и SE может потребоваться **полная переустановка BP и SE**.

### 9.2.1. Полная переустановка BP и SE

Полная переустановка бизнес-процессов и сервисных драйверов выполняется при установке новых версий BP и SE.

Порядок переустановки:

1. Остановите SLES одним из способов:
  - 1.1. С помощью **SSM**.
  - 1.2. С помощью команды **./shutdown.sh**.
2. Удалите содержимое каталога \$SLES/deployed.
3. **Запустите SLES** командой **./startup.sh**.
4. **Разверните** BP и SE.

### 9.2.2. Изменение адреса сервиса deployer

Для изменения URL точки доступа, к которой обращается SLES при установке приложений, измените MIB-настройки группы `CMP_SLES/SlesManagement/ProtocolLayers/TCP`.

При запуске SLES система запускает сервис `deployer` — `{http://sles.sdep.bercut.com/deployer}DeployerPortType`. Этот сервис используется при установке приложений. Чтобы задать адрес для запуска сервиса `deployer`, используйте переменную `ServerAddress` группы `CMP_SLES/SlesManagement/ProtocolLayers/TCP`. Формат значения: `<IP-адрес>:<номер порта>`. Обращение к сервису `deployer` выполняйте в формате: `ServerAddress/deployer`.

### 9.2.3. Особенности выполнения команды `undeploy_all`

Особенности выполнения команды `undeploy_all` при работе SLES в сервисном режиме:

- Пакетный режим `undeploy` выполняется в SLES, а не в `bash`-скрипте, как ранее.
- Команда `undeploy_all` выполняется синхронно. Возврат управления с кодом `'0'` означает, что все запросы успешно завершены и все артефакты выгружены.

Порядок работы системы при выполнении команды `undeploy_all` при работе SLES в сервисном режиме:

1. Приложение дожидается окончания текущих процессов `deploy` и `undeploy`. До окончания выполнения команды `undeploy_all` операции `deploy` и `undeploy` запрещены.
2. Система рассылает всем подписавшимся артефактам нотификацию о переходе в сервисный режим.
3. Система включает HWM-режим для TCP-соединений по всем `portType`.
4. Система отменяет сообщения с кодом `'503'`, которые инициируют новую активность по бизнес-`portType` для HTTP-соединений.
5. Приложение ожидает окончания обработки всех текущих запросов. Количество активных транзакций и сессий становится равным `'0'`.
6. Система выполняет команду `undeploy` для всех артефактов:
  - Признаком корректного выполнения является отсутствие вложенных каталогов в каталоге `deployed/apps`.
  - Если вложенные каталоги остались, SLES принудительно выгружает из внутреннего репозитория все XSD- и WSDL-описания и удаляет папки.
7. Система включает LWM-режим для TCP-соединений по всем `portType`. Разрешаются HTTP-сообщения.
8. Система разрешает выполнение команд `deploy` и `undeploy`.

## 9.3. Настройка SLES

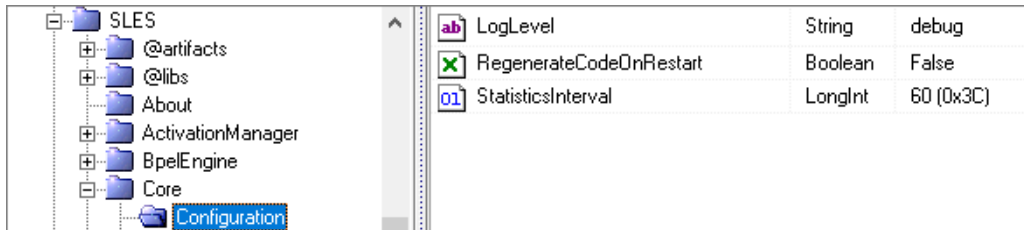
Задайте значения MIB-параметров в группе `CMP_SLES_vX.Y/Core/Configuration`. Также настройте MIB-параметры в подразделах.

**Примечание.** Для локальных соединений настройте маршруты в группе `CMP_SLES_vX.Y/@libs/MessageBus/LocalConnection/ConnectionsList/<имя соединения>/ProtocolLayers/Local`.

Порядок настройки:

1. Общие настройки. В группе CMP\_SLES\_vX.Y/Core/Configuration задайте значения параметра:

- *RegenerateCodeOnRestart*. Флаг перезаписи кода BP и SE при перезапуске SLES:
  - True. При перезапуске SLES система перезаписывает код BP и SE.
  - False. Перезапись не выполняется.



**Рис. 100. Группа CMP\_SLES\_vX.Y/Core/Configuration**

2. Настройте параметры *ActivationManager*.
3. Задайте значения *параметров* группы CMP\_SLES\_vX.Y/BpelEngine/Configuration.
4. Настройте параметры для *планировщика заданий* в группе CMP\_SLES\_vX.Y/Scheduler/Configuration/Schedule.
5. Задайте значения *параметров* группы CMP\_SLES\_vX.Y/SlesManagement.
6. Задайте параметры *распределенных транзакций* в группе CMP\_SLES\_vX.Y/TransactionCoordinator/Configuration.

Также SLES собирает статистику по работе компонентов системы:

- Посмотреть *статистику работы SE* вы можете в группе /CMP\_SLES\_vX.Y/SeEngine/Statistics/.
- Посмотреть *транспортную статистику* по развернутым бизнес-процессам вы можете в группе CMP\_SLES\_vX.Y/TransportEngine/Statistics/ProcessInfo.

### 9.3.1. Настройка ActivationManager

В SLES может быть загружено несколько реализаций одного portType — бизнес-процесса или SE. Настройте фильтры в ActivationManager для выбора бизнес-процесса или SE, в зависимости от входящих параметров сообщения.

ActivationManager позволяет настроить параметры в следующих режимах:

- Параметры для portType не заданы. Для всех запросов, которые поступают на portType, система использует бизнес-процесс, автоматически заданный по умолчанию для данного portType.
- Для portType настроен бизнес-процесс по умолчанию. Система использует указанный бизнес-процесс.
- Для операции настроен фильтр, в котором указан конкретный бизнес-процесс. Для запросов, которые поступают на эту операцию и удовлетворяют условиям фильтра, система использует заданный бизнес-процесс. Для всех остальных запросов на данный portType система использует бизнес-процесс по умолчанию: определенный автоматически или заданный вручную.

Порядок выбора бизнес-процесса:

1. MessageBus отправляет запрос на создание бизнес-процесса или SE.
2. Система определяет, на какой portType поступило сообщение и какую операцию нужно вызвать.

3. Система обращается в ActivationManager, чтобы определить бизнес-процесс, который нужно вызвать.
4. Система проверяет наличие бизнес-процесса по умолчанию для portType и перебирает в алфавитном порядке с учетом регистра все фильтры для операции:
  - Если хотя бы одно условие подходит, система выбирает бизнес-процесс, указанный для фильтра.
  - Если ни одно условие не подходит, система использует бизнес-процесс по умолчанию для portType.

Система формирует сообщение об ошибке ServiceUnavailable, если:

- бизнес-процесс отсутствует в SLES, но фильтр присутствует;
- в SLES отсутствует бизнес-процесс по умолчанию.

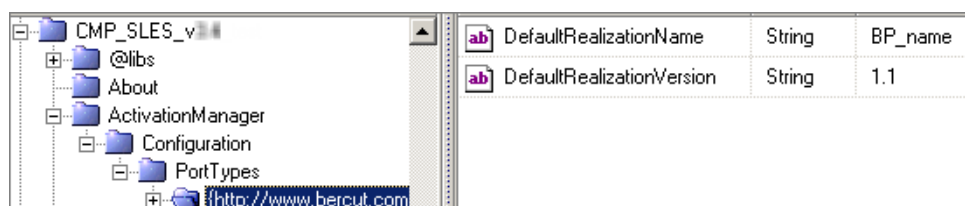
### Настройка бизнес-процесса или SE по умолчанию для portType

Чтобы настроить бизнес-процесс по умолчанию для portType, задайте значения параметров группы CMP\_SLES\_vX.Y/ActivationManager/Configuration/PortTypes/<имя portType>:

- *DefaultRealizationName*. Имя бизнес-процесса или SE по умолчанию. Если параметра нет или значение — пустая строка, то он не используется. Выполняется автоматическое определение бизнес-процесса или SE по умолчанию.
- *DefaultRealizationVersion*. Версия бизнес-процесса или SE по умолчанию. Если настройки нет в MIB, то используется бизнес-процесс или SE по умолчанию с указанным именем. Указанная в MIB версия должна содержать первые две цифры. Третья и четвертая цифры игнорируются при *поиске* реализации.

**i Примечание.** Бизнес-процесс или SE по умолчанию — тот, для которого не назначен ни один из фильтров. Если таких несколько, то выбирается первый по порядку загрузки в SLES.

Имя и версию бизнес-процесса или SE можно посмотреть в тегах name и version XML-описания.



**Рис. 101.** Группа CMP\_SLES\_vX.Y/ActivationManager/Configuration/PortTypes/<имя portType>

Чтобы посмотреть имя и версию бизнес-процесса или SE по умолчанию для portType, используйте параметр *DefaultRealization* группы CMP\_SLES\_vX.Y/ActivationManager/Statistics/PortTypes/<имя portType>.



**Рис. 102.** Группа CMP\_SLES\_vX.Y/ActivationManager/Statistics/PortTypes/<имя portType>

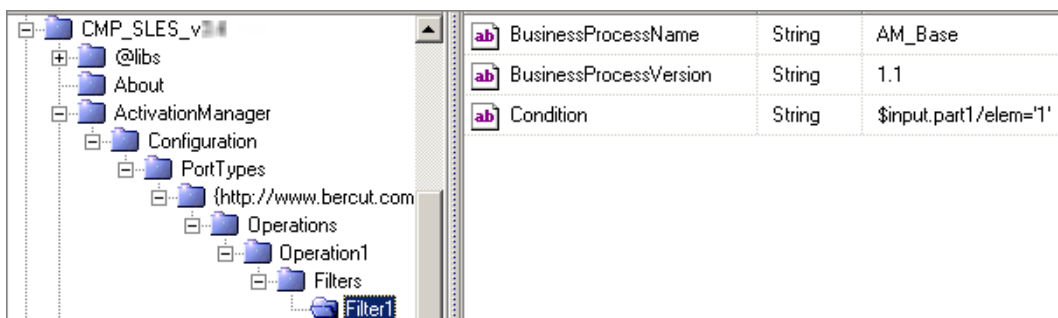
### Настройка бизнес-процесса или SE для конкретных операций

Чтобы настроить бизнес-процесс для операции, задайте значения параметров группы CMP\_SLES\_vX.Y/ActivationManager/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>:

- **BusinessProcessName.** Наименование бизнес-процесса, который будет активирован. Имя вы можете *взять* из значений атрибутов *name* и *version* ATLAS-описания. Если они не заданы — имя берется из WPEL-файла.
- **BusinessProcessVersion.** Версия бизнес-процесса или SE, который будет активирован. Необязательный параметр. Если настройки нет в MIB, то используется бизнес-процесс или SE по умолчанию с указанным именем. Указанная в MIB версия должна содержать первые две цифры. Третья и четвертая цифры игнорируются при *поиске* реализации.

**Примечание.** Бизнес-процесс или SE по умолчанию — тот, для которого не назначен ни один из фильтров. Если таких несколько, то выбирается первый по порядку загрузки в SLES.

- **Condition.** Условие, по которому срабатывает операция. Если условие активации — это XPath-выражение, то результат вычисления приводится к типу boolean. Условие считается выполненным, если после выполнения XPath-выражения и приведения его к типу boolean в результате — true. Описание синтаксиса XPath-выражения в переменной Condition — *разделе*.



**Рис. 103.** Группа CMP\_SLES\_vX.Y/ActivationManager/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>

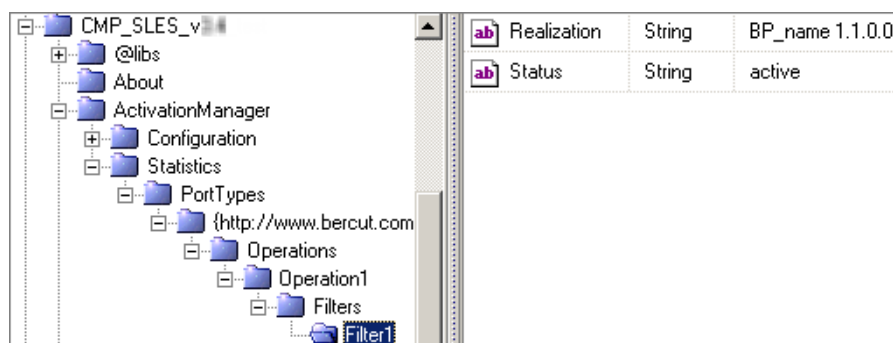


**! Внимание!** Если развернут всего один экземпляр бизнес-процесса или SE, и он указан в одном из фильтров — бизнес-процесса или SE по умолчанию не существует. Все запросы, не попадающие в этот фильтр, будут отклоняться по причине *Service Unavailable*. Если вы настроили фильтр для экземпляра бизнес-процесса или SE, то значит все остальные запросы должны быть маршрутизированы на другой экземпляр.

## Просмотр статистики

Чтобы посмотреть статус фильтра, имя и версию бизнес-процесса для фильтра, используйте параметры группы `CMP_SLES_vX.Y/ActivationManager/Statistics/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>`:

- **Realization.** Имя и версия бизнес-процесса, которые используются для фильтра.
- **Status.** Статус фильтра:
  - active — фильтр в операции активен;
  - inactive — фильтр в операции не активен.



**Рис. 104.** Группа `CMP_SLES_vX.Y/ActivationManager/Statistics/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>`

### 9.3.1.1. Особенности работы ActivationManager

В SLES может быть загружено несколько реализаций одного `portType` — бизнес-процесса или SE. *ActivationManager* используется для выбора реализации, в зависимости от входящих параметров сообщения.

В процессе работы *ActivationManager* определяет имя и версию бизнес-процесса или SE, используя следующие данные:

#### 1. Значения атрибутов *name* и *version* в ATLAS-описании артефакта:

```
<?xml version="1.0" encoding="UTF-8"?>
<BusinessProcess xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
name="Invoker"  xsi:noNamespaceSchemaLocation="http://plv3.office.bercut.ru/atlas/3.0/
xsd/STD_PlV3_BusinessProcessDescriptor_v3.xsd">
  <About vendor="Bercut Ltd." version="1.0.1.19" name="Invoker">
    <Ru>Описание бизнес-процесса</Ru>
    <En>Description of this Business Process</En>
  </About>

<?xml version="1.0" encoding="UTF-8"?>
<ServiceEnabler xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
name="SEHello"
xsi:noNamespaceSchemaLocation="STD_PlV3_ServiceEnablerDescriptor_v3.xsd">
  <About name="SEHello" version="1.0.0.0" vendor="Bercut LLC.">
    <Ru/>
    <En></En>
```

```
</About>
```

2. BPEL — для бизнес-процессов, у которых в сборке нет ATLAS-описания и оно было сформировано автоматически при установке.

```
<?xml version="1.0" encoding="UTF-8"?>
<process
  name="Invoker"
  targetNamespace="http://enterprise.netbeans.org/bpel/BPInvokesHello/Invoker"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

**Настройка** выполняется для каждого portType и операции по отдельности. После настройки активации portType и операции в трейс выводится информационное сообщение:

```
Установлено условие активации '$message.part1/str1='se'' для
'{http://j2ee.netbeans.org/wsdl/hello}helloPortType' операция
'helloOperation' имя реализации 'SEHello 1.0.0.5'
```

При ошибках в условии выводятся предупреждения:

```
Условие активации $message.part5/str1='se' для
{http://j2ee.netbeans.org/wsdl/hello}helloPortType для операции
helloOperation не установлены: com.bercut.mb.parser.xsd.impl.xpath.XPathException:
field part5 not found in {http://j2ee.netbeans.org/wsdl/hello}helloOperationRequest
```

Если для каждого бизнес-процесса или SE установлен свой фильтр, и не настроен portType без фильтра — отсутствует настройка по умолчанию. Запросы, которые не попали ни под один из фильтров, не будут обработаны. Соответствующее предупреждение выводится в трейс:

```
Не найдена ни одна реализация для активации по умолчанию для {http://j2ee.netbeans.org/
wsdl/hello}helloPortType
```

### 9.3.2. Настройка BPELEngine

*Настройте параметры бизнес-процессов и SE для BPEL-сервера CMP\_SLES\_vX.Y/BpelEngine/.*

1. Задайте значения параметров группы CMP\_SLES\_vX.Y/BpelEngine/Configuration:

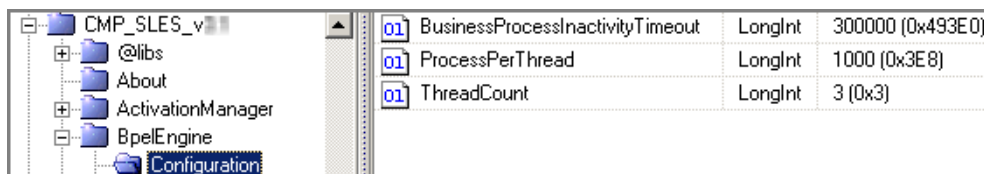
- **BusinessProcessInactivityTimeout.** Тайм-аут ожидания внешних событий, срабатывания таймеров, ответа от внешних систем. Тайм-аут задается в миллисекундах. При неактивности бизнес-процесса в течение тайм-аута система ограничивает время жизни бизнес-процесса. Тайм-аут устанавливается заново каждый раз при завершении выполнения части процесса. После окончания тайм-аута SLES удаляет бизнес-процесс. Значение по умолчанию: '300000'.
- **ProcessPerThread.** Количество бизнес-процессов, обрабатываемых в одном потоке. Значение по умолчанию: '100'. Диапазон: от '100' до '10000'.

Это значение ограничивает количество бизнес-процессов, создаваемых по входящим запросам. Это не относится к созданию бизнес-процессов в результате локальных вызовов, когда один бизнес-процесс вызывает другой. Это сделано для того, чтобы уже запущенные бизнес-процессы могли завершить выполнение, даже если для их выполнения требуется вызвать другой бизнес-процесс. Поэтому число запущенных бизнес-процессов в статистике рабочего *потока* может превышать это значение, иногда — в несколько раз.

**! Внимание!** Параметр `ProcessPerThread` используется для ограничения максимального количества одновременно выполняемых бизнес-процессов.  $MaxProcessCount = ProcessPerThread * ThreadCount$ . Ограничение максимального количества бизнес-процессов необходимо для предотвращения исполнения Full GC на SLES. Когда достигнуто значение `MaxProcessCount` в одновременно запущенных бизнес-процессах, новые процессы перестают создаваться. При попытке создать новый экземпляр система формирует исключение `MaximumProcessExceeded`. Настраивайте значение `ProcessPerThread` в зависимости от веса бизнес-процесса — объема передаваемых данных, сложности логики бизнес-процесса — и от объема памяти, выделенной SLES. Для оценки оптимального значения `MaxProcessCount` необходимо провести нагрузочное тестирование бизнес-процесса. Будет получено количество бизнес-процессов в SLES, при котором начинает запускаться Full GC (`FullGCProcessCounter`). Значение `ProcessPerThread` вычисляется по формуле:  $ProcessPerThread = (0.8 * FullGCProcessCounter) / ThreadCount$ .

- `ThreadCount`. Количество потоков обработки на BPEL-сервере. Значение по умолчанию: '3'. Диапазон: от '1' до '20'.

**i Примечание.** Рекомендуемое значение `ThreadCount` —  $0.8 * CPU$  сервера. Остальные CPU остаются свободными для использования транспортными потоками RTSIB.



**Рис. 105.** Группа `CMP_SLES_vX.Y/BpelEngine/Configuration`

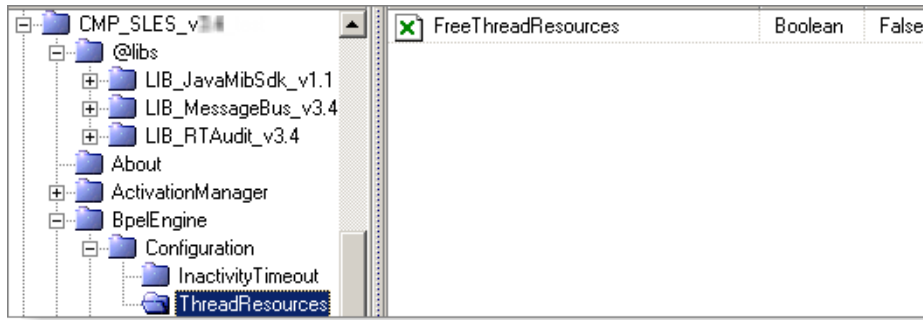
2. Чтобы указать индивидуальные настройки для входящих `portType`, задайте значения параметров группы `/CMP_SLES_vX.Y/BpelEngine/Configuration/InactivityTimeout/<имя portType>/`:

- `InactivityTimeout`. Количество потоков обработки на BPEL-сервере для конкретного `portType`.



**Рис. 106.** Группа `/CMP_SLES_vX.Y/BpelEngine/Configuration/InactivityTimeout/<имя portType>/`

3. В группе `CMP_SLES_vX.Y/BpelEngine/Configuration/ThreadResources` хранятся настройки потокозависимых ресурсов. Чтобы очистить потоков от объектов, накопленных в кеше, задайте значение параметра `FreeThreadResources`. При значении `True` выполняется очистка потоков от объектов, накопленных в кеше. После этого кеш наполняется заново. Значение по умолчанию: 'False'.

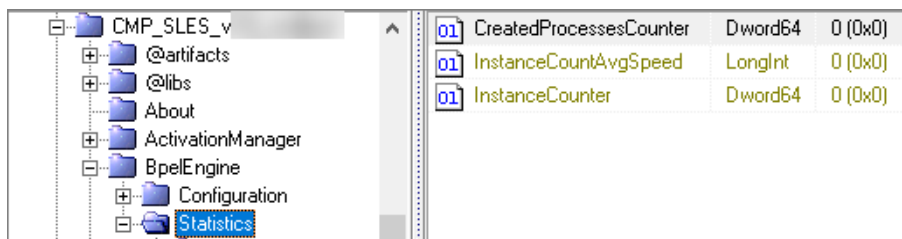


**Рис. 107. Группа CMP\_SLES\_vX.Y/BpelEngine/Configuration/ThreadResources**

### Просмотр статистики

1. Чтобы посмотреть информацию о бизнес-процессах, используйте параметры группы /CMP\_SLES\_vX.Y/BpelEngine/Statistics/:

- *CreatedProcessesCounter*. Количество бизнес-процессов, созданных с момента запуска SLES.
- *InstanceAvgSpeed*. Среднее количество запусков бизнес-процессов за интервал времени сбора статистики.
- *InstanceCounter*. Количество бизнес-процессов, выполняемых в данный момент.



**Рис. 108. Группа /CMP\_SLES\_vX.Y/BpelEngine/Statistics/**

2. Чтобы посмотреть информацию по определенному бизнес-процессу, используйте параметры группы /CMP\_SLES\_vX.Y/BpelEngine/Statistics/ProcessInfo/<имя portType>:

- *InstanceCounter*. Количество экземпляров данного бизнес-процесса, выполняемых в данный момент.
- *ProcessLifeTimeAvg*. Среднее время жизни экземпляра бизнес-процесса, в миллисекундах.
- *CreatedProcessesAvgSpeed*. Среднее количество экземпляров бизнес-процесса, создаваемых за интервал времени сбора статистики.
- *CreatedProcessesCounter*. Количество бизнес-процессов, созданных с момента запуска SLES.
- *ProcessNormalFinishedCounter*. Количество экземпляров бизнес-процесса, завершенных без ошибок.
- *RequestAvgSpeed*. Среднее количество входящих запросов к бизнес-процессу за интервал времени сбора статистики.

Группа содержит информацию о бизнес-процессах, запущенных в данный момент времени.

01	CreatedProcessesAvgSpeed	LongInt	0 (0x0)
01	CreatedProcessesCounter	Dword64	0 (0x0)
01	InstanceCounter	Dword64	0 (0x0)
01	ProcessFailFinishedCounter	Dword64	0 (0x0)
01	ProcessLifeTimeAvg	Int64	0 (0x0)
01	RequestAvgSpeed	Double	0

**Рис. 109.** Группа /CMP\_SLES\_vX.Y/BpelEngine/Statistics/ProcessInfo/<имя portType>

3. Чтобы посмотреть информацию по определенному потоку, используйте параметры группы /CMP\_SLES\_vX.Y/BpelEngine/StatisticsWorkingThreads/BPEL Thread <номер потока>/:

**Примечание.** Число потоков задается в параметре /CMP\_SLES\_vX.Y/BpelEngine/Configuration/ProcessPerThread.

- **MessageCounter.** Количество сообщений в очереди на обработку в данном потоке.
- **ProcessPerThreadCounter.** Количество бизнес-процессов, обрабатываемых в данном потоке. Это значение может превышать значение из параметра **ProcessPerThread**, поскольку SLES разрешает создавать экземпляры бизнес-процессов сверх этого значения, если это требуется для продолжения выполнения уже запущенных бизнес-процессов.

01	MessageCounter	LongInt	0 (0x0)
01	ProcessPerThreadCounter	LongInt	0 (0x0)

**Рис. 110.** Группа /CMP\_SLES\_vX.Y/BpelEngine/StatisticsWorkingThreads/BPEL Thread <номер потока>/

### 9.3.3. Настройка Scheduler

Каждый экземпляр SLES имеет планировщик, который позволяет запускать по расписанию определенные бизнес-процессы.

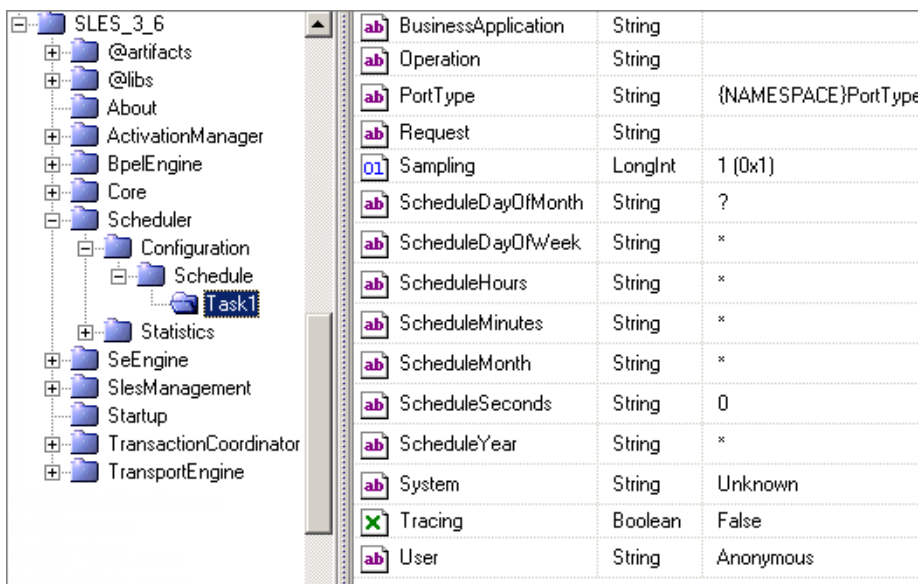
Чтобы создать список задач:

1. Создайте для каждой задачи группу, вложенную в CMP\_SLES\_vX.Y/Scheduler/Configuration/Schedule. Имя группы должно соответствовать имени задачи.
2. Для каждой задачи задайте значения параметров:
  - **BusinessApplication.** Имя бизнес-приложения для соответствующего объекта.
  - **Operation.** Целевая операция.
  - **PortType.** Целевой portType в формате: {namespace}name.
  - **Request.** XPath-выражения, которые формируют запрос к бизнес-процессу.

- **Sampling.** Порядковый номер сообщений для трассировки. Например, при значении '5' выполняется трассировка каждого пятого сообщения. Значение используется при поставленном флаге *Enabled*.
- **ScheduleYear, ScheduleMonth, ScheduleDayOfMonth, ScheduleDayOfWeek, ScheduleHours, ScheduleMinutes, ScheduleSeconds.** Интервалы в формате Cron: годы, месяцы, дни месяца, дни недели, часы, минуты и секунды.
- **System.** Имя системы, которая выполняет задание по расписанию. Значение доступно из бизнес-процесса в специальной переменной *principal-SystemPrincipal*, а в исходящих вызовах передается в заголовке *SystemPrincipal*.
- **Tracing.** Флаг включения трассировки вызовов:
  - True — трассировка включена;
  - False — трассировка выключена. Значение по умолчанию.

Если трассировка уже включена уровнем выше, то этот флаг задает, нужно ли переопределять настройки трассировки для этого уровня. Если значение — False, то переопределять не нужно.

- **User.** Пользователь, который выполняет задание по расписанию. Значение доступно из бизнес-процесса в специальной переменной *principal-UserPrincipal*, а в исходящих вызовах передается в заголовке *UserPrincipal*.



**Рис. 111.** CMP\_SLES\_vX.Y/Scheduler/Configuration/Schedule/<имя задачи>

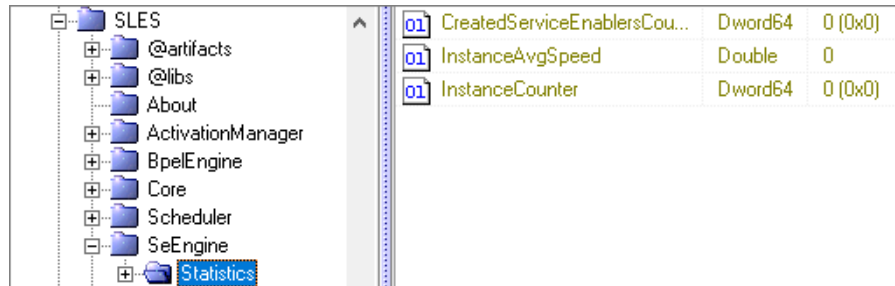
Результаты настройки и работы каждой задачи записываются в лог. Формирование SOAP-запроса и вызов операции система выполняет по расписанию заданий, ответ игнорируется. В случае возникновения ошибок отображается соответствующее предупреждение с указанием причины. В статистике отображается текущее состояние каждого элемента, время следующего запуска и другие данные.

### 9.3.4. Настройка SeEngine

Группа содержит статистику выполнения SE. Настройка SE выполняются в группе [BpelEngine](#).

1. Чтобы посмотреть информацию о SE, используйте параметры группы / CMP\_SLES\_vX.Y/SeEngine/Statistics/:

- *CreatedProcessesCounter*. Количество экземпляров всех SE, запущенных с момента запуска SLES.
- *InstanceAvgSpeed*. Среднее количество запусков всех SE за интервал времени сбора статистики.
- *InstanceCounter*. Количество экземпляров всех SE, выполняемых в данный момент.

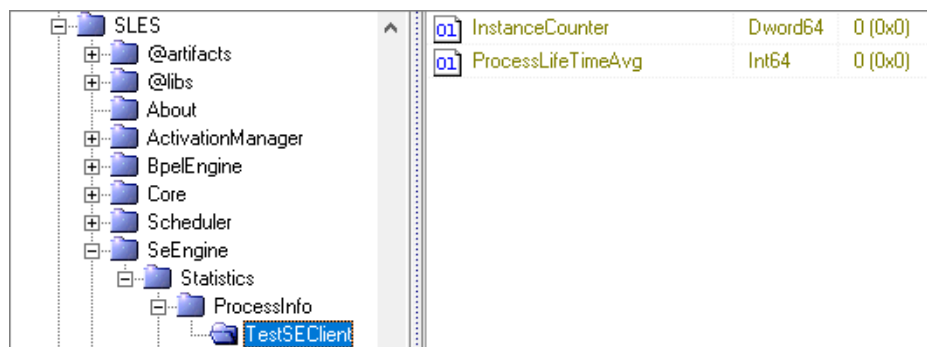


01	CreatedServiceEnablersCou...	Dword64	0 (0x0)
01	InstanceAvgSpeed	Double	0
01	InstanceCounter	Dword64	0 (0x0)

**Рис. 112.** Группа /CMP\_SLES\_vX.Y/SeEngine/Statistics/

2. Чтобы посмотреть информацию о конкретном экземпляре SE, используйте параметры группы /CMP\_SLES\_vX.Y/SeEngine/Statistics/ProcessInfo/<имя SE>:

- *InstanceCounter*. Количество экземпляров конкретного SE, исполняемых в текущий момент.
- *ProcessLifeTimeAvg*. Среднее время жизни экземпляров конкретного SE.

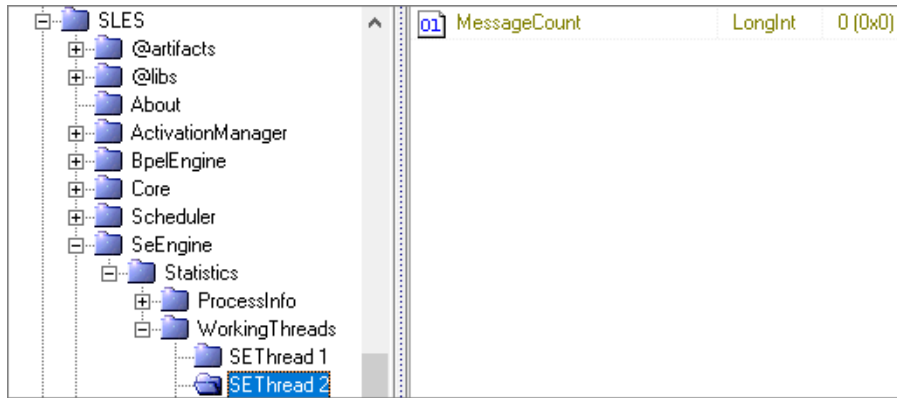


01	InstanceCounter	Dword64	0 (0x0)
01	ProcessLifeTimeAvg	Int64	0 (0x0)

**Рис. 113.** Группа /CMP\_SLES\_vX.Y/SeEngine/Statistics/ProcessInfo/<имя SE>

3. Чтобы посмотреть информацию о конкретном потоке, используйте параметры группы /CMP\_SLES\_vX.Y/SeEngine/Statistics/WorkingThreads/SEThread <номер потока>:

- *MessageCount*. Количество сообщений, поставленных в очередь на обработку.



**Рис. 114. Группа /CMP\_SLES\_vX.Y/SeEngine/Statistics/WorkingThreads/SEThread <номер потока>**

### 9.3.5. Настройка SlesManagement

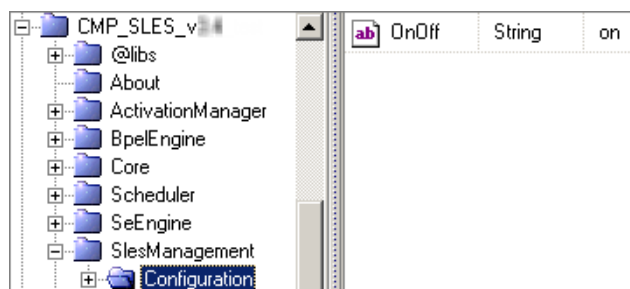
Настройте параметры HTTP-сервера.

SlesManagement — это HTTP-сервер, который работает с использованием порта, заданного в параметре /SlesManagement/ProtocolLayers/Tcp/ServerAddress.

Ниже описаны переменные, настройка которых может потребоваться в SlesManagement. Остальные переменные являются стандартными для RTSIB и HTTP-сервера и не нуждаются в настройке:

1. Вы можете отключить сервер. Например, это требуется, если все артефакты уже установлены и надо предупредить случайное срабатывание команд deploy и undeploy на SLES. Для этого поставьте или снимите флаг включения точки доступа. Значения параметра OnOff группы CMP\_SLES\_vX.Y/SlesManagement/Configuration:

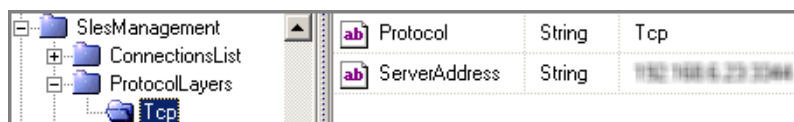
- on — включение точки доступа;
- off — выключение точки доступа.



**Рис. 115. Группа CMP\_SLES\_vX.Y/SlesManagement/Configuration**

2. Задайте IP-адрес и номер порта сервера, используемого для установки BP и SE, в параметре ServerAddress группы CMP\_SLES\_vX.Y/SlesManagement/ProtocolLayers/Tcp. Формат: IP address:Port.

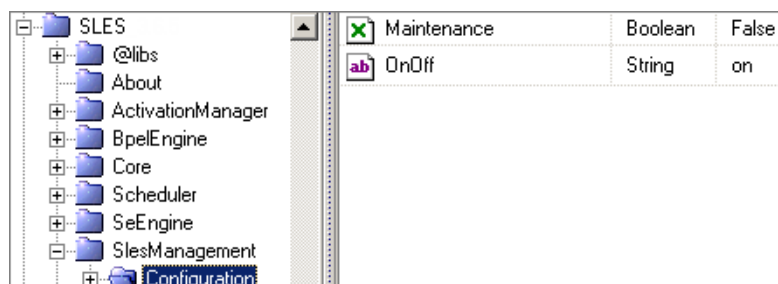




**Рис. 116.** Группа *CMP\_SLES\_vX.Y/SlesManagement/ProtocolLayers/Tcp*

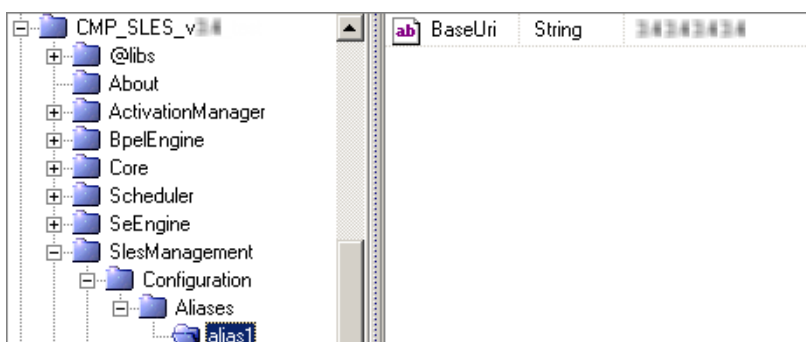
3. Включите или отключите *сервисный режим*. Значения параметра *Maintenance*:

- True — режим включен;
- False — режим выключен.



**Рис. 117.** Группа *CMP\_SLES\_vX.Y/SlesManagement/Configuration*

4. При необходимости задайте URI алиасов в группе *CMP\_SLES\_vX.Y/SlesManagement/Configuration/Aliases/<название алиаса>*.



**Рис. 118.** Группа *CMP\_SLES\_vX.Y/SlesManagement/Configuration/Aliases/<название алиаса>*

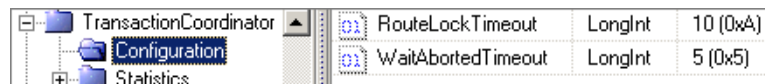
### 9.3.6. Настройка TransactionCoordinator

*Transaction Coordinator* — координатор транзакций, который управляет распределенными транзакциями.

Чтобы настроить координатор транзакций, задайте значения параметров в группе *SLES\_vX.Y/TransactionCoordinator/Configuration*:

- *RouteLockTimeout*. Максимальное время блокирования таблицы маршрутизации транзакции, в секундах. Если после завершения заданного времени сигнал на разблокирование данной транзакции не поступил, система отменяет транзакцию.
- *WaitAbortedTimeout*. Значение по умолчанию для тайм-аута ожидания сообщения aborted после отмены транзакции, в секундах. После окончания тайм-аута система

удаляет транзакцию из таблицы маршрутизации, если хотя бы одна операция транзакции была отменена.



**Рис. 119. Настройка координатора транзакций**

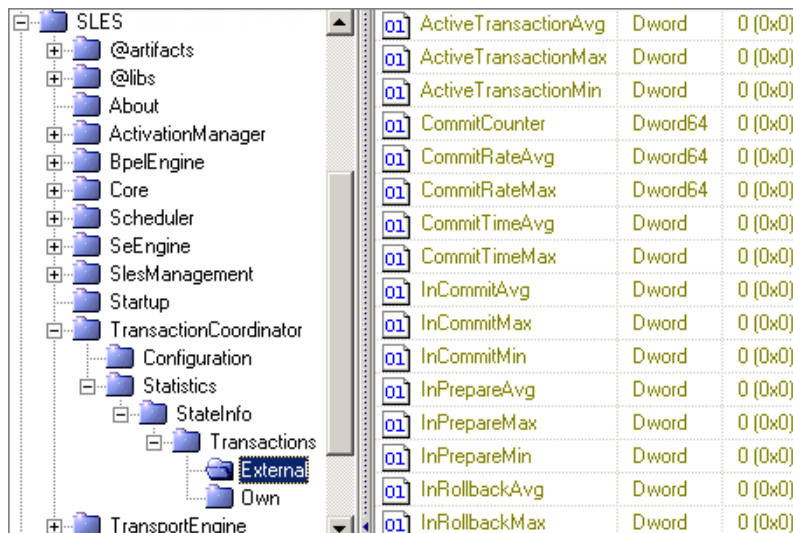
Система собирает статистику по распределенным транзакциям в следующих MIB-группах:

- `CMP_SLES_vX.Y/TransactionCoordinator/Statistics/StateInfo/Transactions/External` — по внешним распределенным транзакциям, которые поступили из внешнего координатора;
- `CMP_SLES_vX.Y/TransactionCoordinator/Statistics/StateInfo/Transactions/Own` — по распределенным транзакциям, которые запущены на данном SLES.

Статистические переменные:

- *ActiveTransactionAvg*. Среднее количество активных транзакций.
- *ActiveTransactionMax*. Максимальное количество активных транзакций.
- *ActiveTransactionMin*. Минимальное количество активных транзакций.
- *CommitCounter*. Общее количество фиксаций транзакций (commit).
- *CommitRateAvg*. Среднее количество фиксаций (commit) в секунду.
- *CommitRateMax*. Максимальное количество фиксаций (commit).
- *CommitTimeAvg*. Среднее время фиксации (commit) транзакции в миллисекундах.
- *CommitTimeMax*. Максимальное время фиксации (commit) транзакции в миллисекундах.
- *InCommitAvg*. Среднее количество транзакций, которые выполняют фиксацию (commit).
- *InCommitMax*. Максимальное количество транзакций, которые выполняют фиксацию (commit).
- *InCommitMin*. Минимальное количество транзакций, которые выполняют фиксацию (commit).
- *InPrepareAvg*. Среднее количество транзакций, которые выполняют подготовку.
- *InPrepareMax*. Максимальное количество транзакций, которые выполняют подготовку.
- *InPrepareMin*. Минимальное количество транзакций, которые выполняют подготовку.
- *InRollbackAvg*. Среднее количество транзакций, которые выполняют отмену (rollback).
- *InRollbackMax*. Максимальное количество транзакций, которые выполняют отмену (rollback).
- *InRollbackMin*. Минимальное количество транзакций, которые выполняют отмену (rollback).
- *PrepareCounter*. Общее количество подготовок транзакций.
- *PrepareRateAvg*. Среднее количество подготовок в секунду.
- *PrepareRateMax*. Максимальное количество подготовок.
- *PrepareTimeAvg*. Среднее время подготовки транзакции в миллисекундах.
- *PrepareTimeMax*. Максимальное время подготовок транзакции в миллисекундах.
- *RollbackCounter*. Общее количество отмены транзакций (rollback).
- *RollbackRateAvg*. Среднее количество отмен (rollback) в секунду.
- *RollbackRateMax*. Максимальное количество отмен (rollback).
- *RollbackTimeAvg*. Среднее время отмены транзакции (rollback) в миллисекундах.
- *RollbackTimeMax*. Максимальное время отмены транзакции (rollback) в миллисекундах.

- *TransactionCounter*. Общее количество новых транзакций.
- *TransactionRateAvg*. Среднее количество транзакций в секунду.
- *TransactionRateMax*. Максимальное количество транзакций.
- *TransactionTimeAvg*. Среднее время выполнения транзакции в миллисекундах.
- *TransactionTimeMax*. Максимальное время выполнения транзакции в миллисекундах.



01	ActiveTransactionAvg	Dword	0 (0x0)
01	ActiveTransactionMax	Dword	0 (0x0)
01	ActiveTransactionMin	Dword	0 (0x0)
01	CommitCounter	Dword64	0 (0x0)
01	CommitRateAvg	Dword64	0 (0x0)
01	CommitRateMax	Dword64	0 (0x0)
01	CommitTimeAvg	Dword	0 (0x0)
01	CommitTimeMax	Dword	0 (0x0)
01	InCommitAvg	Dword	0 (0x0)
01	InCommitMax	Dword	0 (0x0)
01	InCommitMin	Dword	0 (0x0)
01	InPrepareAvg	Dword	0 (0x0)
01	InPrepareMax	Dword	0 (0x0)
01	InPrepareMin	Dword	0 (0x0)
01	InRollbackAvg	Dword	0 (0x0)
01	InRollbackMax	Dword	0 (0x0)

**Рис. 120. Статистика по распределенным транзакциям**

### 9.3.7. Настройка TransportEngine

*SLES формирует транспортную статистику по развернутым бизнес-процессам.*

Для каждого развернутого бизнес-процесса система создает папку <ИМЯ бизнес-процесса> в MIB-группе `CMP_SLES_vX.Y/TransportEngine/Statistics/ProcessInfo`. В созданной папке хранится статистика с группировкой по артефактам:

- по входящим сообщениям;
- по исходящим сообщениям.

*Общая статистика* по portType отображается в группах:

- `CMP_SLES_vX.Y/@libs/MessageBus/Core/Statistics/StateInfo/`
- `CMP_SLES_vX.Y/@libs/MessageBus/HttpServer/Statistics/StateInfo`
- `CMP_SLES_vX.Y/@libs/MessageBus/ExternalConnections/Statistics/StateInfo`

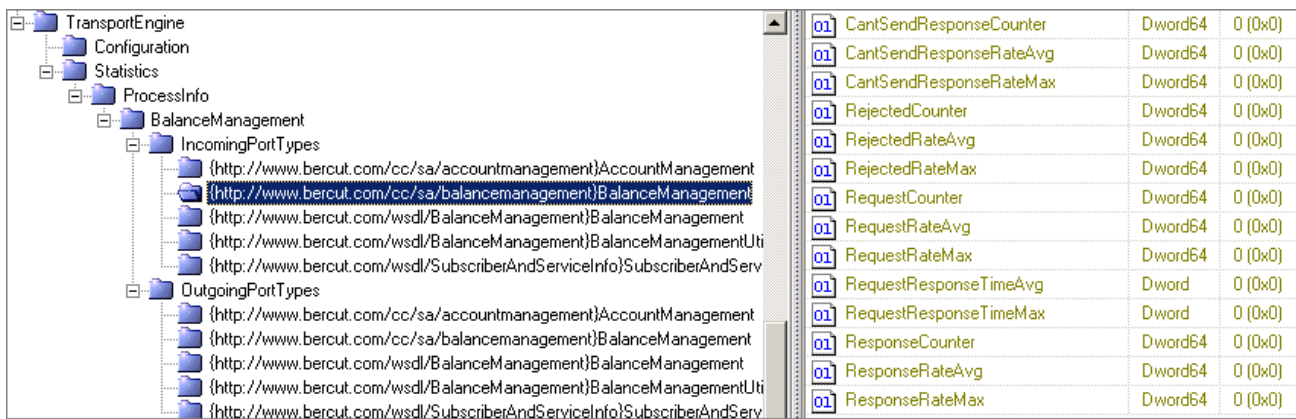
#### Статистика по входящим сообщениям

Система собирает статистику в следующих MIB-группах:

- `CMP_SLES_vX.Y/TransportEngine/Statistics/ProcessInfo/<ИМЯ бизнес-процесса>/IncomingPortTypes/<ИМЯ portType>` — по входящему portType бизнес-процесса;
- `CMP_SLES_vX.Y/TransportEngine/Statistics/ProcessInfo/<ИМЯ бизнес-процесса>/IncomingPortTypes/<ИМЯ portType>/Operations/<ИМЯ операции>` — по операции входящего portType бизнес-процесса.

Статистические переменные по входящим сообщениям:

- *CantSendResponseConter*. Общее количество неотправленных запросов.
- *CantSendResponseRateAvg*. Среднее количество неотправленных запросов в секунду.
- *CantSendResponseRateMax*. Максимальное количество неотправленных запросов в секунду.
- *RejectedCounter*. Общее количество отклоненных запросов.
- *RejectedRateAvg*. Среднее количество отклоненных запросов в секунду.
- *RejectedRateMax*. Максимальное количество отклоненных запросов в секунду.
- *RequestCounter*. Общее количество входящих запросов.
- *RequestRateAvg*. Среднее количество входящих запросов.
- *RequestRateMax*. Максимальное количество входящих запросов.
- *RequestResponseTimeAvg*. Среднее время обработки запроса, в миллисекундах.
- *RequestResponseTimeMax*. Максимальное время обработки запроса, в миллисекундах.
- *ResponseCounter*. Общее количество отправленных ответов.
- *ResponseRateAvg*. Среднее количество ответов в секунду.
- *ResponseRateMax*. Максимальное количество ответов в секунду.



**Рис. 121. Статистика по входящим сообщениям**

### Статистика по исходящим сообщениям

Система собирает статистику в следующих MIB-группах:

- CMP\_SLES\_vX.Y/TransportEngine/Statistics/ProcessInfo/<имя бизнес-процесса>/OutgoingPortTypes/<имя portType> — по исходящему portType бизнес-процесса;
- CMP\_SLES\_vX.Y/TransportEngine/Statistics/ProcessInfo/<имя бизнес-процесса>/OutgoingPortTypes/<имя portType>/Operations/<имя операции> — по операции исходящего portType бизнес-процесса.

Статистические переменные по исходящим сообщениям:

- *QueryCounter*. Общее количество отправленных запросов.
- *QueryRateAvg*. Среднее количество отправленных запросов в секунду.
- *QueryRateMax*. Максимальное количество отправленных запросов в секунду.
- *QueryResponseTimeAvg*. Среднее время ожидания ответа на запрос, в миллисекундах.
- *QueryResponseTimeMax*. Максимальное время ожидания ответа на запрос, в миллисекундах.
- *RejectedCounter*. Общее количество отклоненных или потерянных запросов.
- *RejectedRateAvg*. Среднее количество отклоненных или потерянных запросов в секунду.

- *RejectedRateMax*. Максимальное количество отклоненных или потерянных запросов в секунду.
- *ResponseCounter*. Общее количество ответов.
- *ResponseRateAvg*. Среднее количество ответов в секунду.
- *ResponseRateMax*. Максимальное количество ответов в секунду.
- *TimeoutCounter*. Общее количество тайм-аутов запроса.
- *TimeoutRateAvg*. Среднее количество тайм-аутов запроса в секунду.
- *TimeoutRateMax*. Максимальное количество тайм-аутов запроса в секунду.

01	QueryCounter	Dword64	0 (0x0)
01	QueryRateAvg	Dword64	0 (0x0)
01	QueryRateMax	Dword64	0 (0x0)
01	QueryResponseTimeAvg	Dword	0 (0x0)
01	QueryResponseTimeMax	Dword	0 (0x0)
01	RejectedCounter	Dword64	0 (0x0)
01	RejectedRateAvg	Dword64	0 (0x0)
01	RejectedRateMax	Dword64	0 (0x0)
01	ResponseCounter	Dword64	0 (0x0)
01	ResponseRateAvg	Dword64	0 (0x0)
01	ResponseRateMax	Dword64	0 (0x0)
01	TimeoutCounter	Dword64	0 (0x0)
01	TimeoutRateAvg	Dword64	0 (0x0)
01	TimeoutRateMax	Dword64	0 (0x0)

**Рис. 122. Статистика по исходящим сообщениям**

## 9.4. Настройка конфигурационных параметров ВР

Настройте конфигурационные параметры ВР в MIB-группах: `CMP_SLES_vX.Y/@artifacts/<название ВР>_<версия ВР>/Configuration`.

Каждый ВР может иметь специфический набор конфигурационных параметров. Настройте конфигурационные параметры ВР в MIB-группах `CMP_SLES_vX.Y/@artifacts/<название ВР>_<версия ВР>/Configuration`.

ab	ChangeTrplPrefix	String	CHANGE_TRPL_TEXT_SMS:
ab	ClientTypeId	String	9,10,11,12,13,14
01	InvParamApplicationId	LongInt	-1 (0xFFFFFFFF)
ab	LogLevel	String	minimum
ab	MessageSplitterRegExp	String	(".*"["\n"]+)
ab	RBTBlackListClientTypesParamName	String	RBT_BLACK_LIST_CT_ID
ab	RBTEnableServiceIdParamName	String	RBT_ACCESS_MARKER_FOR_B2B
ab	RBTEnableServiceStatusesParamName	String	RBT_ACCESS_MARKER_STATUSES
01	RBTServiceId	LongInt	214 (0xD6)
ab	SMSSourceAddress	String	Operator
ab	SMSprefix	String	SMS:
ab	ServiceCodeList	String	115,116,121,127,128,142
01	StatisticsInterval	LongInt	60 (0x3C)
ab	USSDprefix	String	USSD:

**Рис. 123. Параметры ВР**

## 9.5. Сервисный режим

*Сервисный режим, или режим технического обслуживания, позволяет на SLES устанавливать и удалять сервисы, выполнять тесты без последствий для работающей системы.*

Для включения сервисного режима задайте значение 'True' в параметре *Maintenance* группы `CMP_SLES_vX.Y/SlesManagement/Configuration`.

Сервисный режим позволяет тестировать сервис по HTTP — RTSIB/HTTP, SOAP/HTTP или REST — с использованием локальной маршрутизации и транзакционных или сессионных исходящих вызовов.

Возможности артефактов при сервисном режиме:

- артефакты могут подписаться на события включения и отключения режима;
- артефакты не могут отменить переход в режим.

Бизнес-portType — это portType, которые реализуются развернутыми артефактами. Они перечисляются в верхней части HTML-страницы публикации.

При включении режима система выполняет следующие действия:

1. Отображает в логе сообщение о начале работы приложения в сервисном режиме.
2. Приложение дожидается окончания текущих процессов `deploy` и `undeploy`. На период перехода в сервисный режим процессы `deploy` и `undeploy` запрещены.
3. Рассылает всем подписавшимся артефактам нотификацию о переходе в сервисный режим.
4. Устанавливает HWM-режим для TCP- и HTTP-соединений для всех portType:
  - 4.1. Все portType доступны по локальным соединениям, что позволяет приложениям корректно завершать выполнение текущих сервисов при выполнении локальных вызовов или по RTSIB к другим приложениям.
  - 4.2. В период перехода HTTP-вызовы на бизнес-portType завершаются с кодом '503 Service Unavailable', если эти вызовы инициируют создание новой сессии или транзакции.
  - 4.3. Не учитывает HWM-режим при отправке и получении сообщений по служебным portType, поэтому текущие транзакции и сессии завершаются корректно.
  - 4.4. Приложение ожидает окончания обработки всех текущих запросов. Количество активных транзакций и сессий должно быть '0'.
5. Устанавливает статус приложения `Status=nonoperating` в группе `Core/Statistics`.
6. Разрешает выполнение команд `deploy` и `undeploy`.
7. Устанавливает LWM режим для HTTP-соединений для всех бизнес-portType, что позволяет выполнять тестовые HTTP-вызовы.
8. Отображает в логе сообщение о выполненном переходе приложения в сервисный режим.

При включенном сервисном режиме:

- SLES недоступен для обработки продуктивных запросов;
- все бизнес-portType находятся в состоянии HWM и недоступны для взаимодействия по RTSIB/TCP, но доступны для локальных вызовов;
- команды `deploy` и `undeploy` доступны;

- для portType, развернутых во время действия режима, устанавливается HWM-статус;
- для артефактов, развернутых во время действия режима, отправляется нотификация о включении режима;
- portType для RTSIB/TCP-клиентов находятся в HWM-статусе;
- portType доступны для сервисных и служебных HTTP-вызовов.

При выключении режима система выполняет следующие действия:

1. Отображает в логе сообщение о начале отключения сервисного режима.
2. Устанавливает LWM-режим для RTSIB/TCP-соединений для всех portType.
3. Рассылает всем подписавшимся артефактам нотификацию о выключении режима.
4. Устанавливает статус приложения Status=available в группе Core/Statistics.
5. Отображает в логе сообщение об отключении сервисного режима.

При запуске приложения в сервисном режиме система выполняет следующие действия:

1. Отображает в логе сообщение о запуске приложения в сервисном режиме.
2. Выполняет запуск приложения с установленным HWM-статусом для всех portType.
3. Устанавливает после запуска статус приложения Status=nonoperating в группе Core/Statistics.
4. Отправляет каждому артефакту нотификацию о переходе в сервисный режим.
5. После окончания процедуры запуска отображает в логе сообщение о работе приложения в сервисном режиме.

**i Примечание.**

1. Отдельный артефакт невозможно перевести в сервисный режим.
2. При отключении входящего HTTP-трафика могут возникать проблемы.

*Для перехода SLES в сервисный режим на него должна перестать поступать продуктивная нагрузка, чтобы завершились все активные процессы. Для этого portType переводятся в HWM-режим. Для RTSIB/TCP-соединений информация об этом рассылается в служебных сообщениях. Для HTTP-соединений сообщения не отправляются. Для HTTP-соединений используется один и тот же порт для входящей нагрузки и для публикации, поэтому закрыть его нельзя.*

*Чтобы отключить входящий HTTP-трафик при переходе в сервисный режим, SLES отменяет все HTTP-сообщения, которые инициировали новую активность, с кодом '503'. Удаленная HTTP-сторона при получении таких сообщений переключается на резервный сервер и перестает отправлять сообщения.*

*Если удаленная сторона не может переключиться на резервный сервер или использует автоматическое переключение в обратном направлении, то возникают проблемы. Для устранения проблем перехода в сервисный режим SLES, которые обрабатывают HTTP-трафик, внимательно изучите схемы переключения на резерв в клиентских системах.*

## 9.6. Логирование

Логирование в SLES — многоуровневое, в соответствии со структурой. Каждая библиотека (ATLAS, MessageBus), функциональная часть (Activation Manager,

BP Engine, TransportEngine) или артефакт записывают логи в рамках приложения. Чтобы различить их, система использует имя класса логирования и имя экземпляра. Лог содержит текст сообщения и дополнительные параметры. К дополнительным параметрам относятся Business Application и SessionId, которые нужны системе Operations&Management.

### 9.6.1. Параметр Business Application для логирования

Артефакты (BP или SE) выполняются в рамках бизнес-приложения (Business Application). Система записывает имя бизнес-приложения в сообщения ATLAS-логов и алармов по работе этого артефакта.

Способы определения бизнес-приложения:

1. Бизнес-приложение получено в ATLAS-контейнере во входящем сообщении.
2. Если бизнес-приложение не получено в ATLAS-контейнере, система использует бизнес-приложение из файла META-INF/manifest.mf артефакта. Бизнес-приложение указывается в заголовке Business-Application.
3. Если бизнес-приложение не указано ни во входящем контейнере, ни в файле manifest.mf артефакта, в качестве бизнес-приложения используется ATLAS имя артефакта.

Имя бизнес-приложения не отображается в трейсе, если оно неизвестно. Например:

- TransportEngine не знает, к какому бизнес-приложению относится входящий вызов.
- ActivationManager выбрал реализацию вызываемой операции, но еще не создал ее. Артефакт, определенный при развертывании, неизвестен бизнес-приложению.

### 9.6.2. Параметр SessionId для логирования

SessionId существует в рамках обработки вызова. При логировании запуска, остановки приложения и изменении параметров конфигурации он отсутствует. В сообщениях при обработке вызова он может отсутствовать, если на этот момент SessionId еще не определен, например, при получении HTTP-сообщения от внешней системы.

Алгоритм получения SessionId:

- SessionId может быть получен по шине во входящем сообщении в ATLAS контейнере.
- Если SessionId не получен во входящем сообщении, то SessionId формируется в процессе его обработки.

## 9.7. Статистика по развернутым бизнес-процессам

*SLES формирует статистику по развернутым бизнес-процессам.*

По статистике бизнес-процесса можно определить:

- может ли бизнес-процесс быть инициатором транзакции;
- может ли бизнес-процесс быть полностью транзакционным;
- может ли бизнес-процесс использовать корреляцию;
- может ли бизнес-процесс быть инициатором сессии;
- может ли бизнес-процесс быть сессионным.



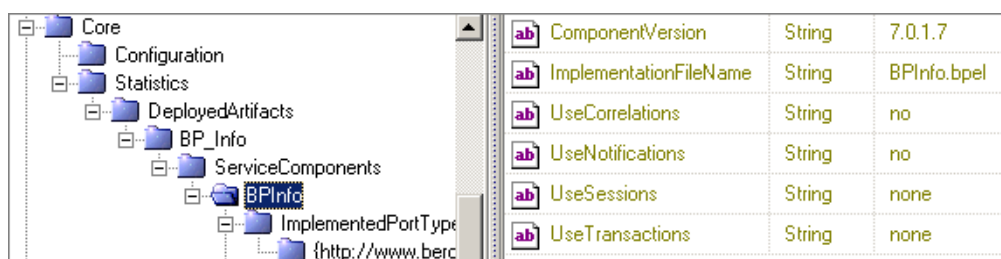
Для каждого развернутого бизнес-процесса система создает папку <имя бизнес-процесса> в MIB-группе CMP\_SLES\_vX.Y/Core/Statistics/DeployedArtifacts/. В созданной папке хранится статистика:

- по реализуемым portType;
- по используемым portType.

По статистике portType вы можете определить, является ли portType транзакционным и сессионным.

В группе CMP\_SLES\_vX.Y/Core/Statistics/DeployedArtifacts/<имя бизнес-процесса>/ServiceComponents/<имя компонента>/ хранится общая статистика по бизнес-процессу:

- *UseTransactions*. Тип *транзакционности* бизнес-процесса:
  - coordinator
  - participant
  - none — не транзакционный.
- *UseSessions*. Тип *сессионности* бизнес-процесса:
  - client — клиентский;
  - server — серверный;
  - client\_server — клиентский и серверный;
  - none — не сессионный;
  - unknown — неизвестный.
- *UseCorrelations*. Флаг использования корреляции.
- *UseNotifications*. Флаг использования нотификаций.
- *ImplementationFileName*. Имя файла бизнес-процесса.
- *ComponentVersion*. Версия бизнес-процесса.



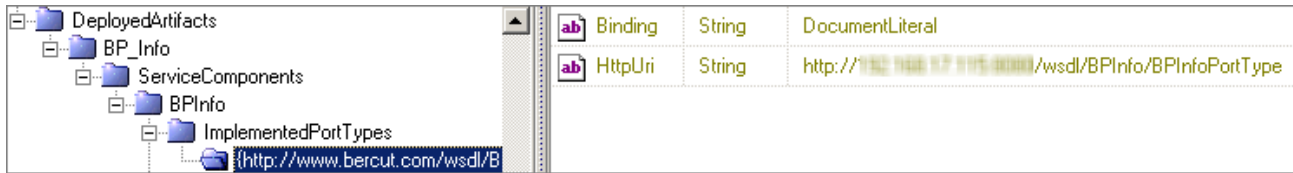
ComponentVersion	String	7.0.1.7
ImplementationFileName	String	BPInfo.bpel
UseCorrelations	String	no
UseNotifications	String	no
UseSessions	String	none
UseTransactions	String	none

**Рис. 124. Статистика по бизнес-процессу**

### Статистика по реализуемым portType

Система собирает статистику в MIB-группе CMP\_SLES\_vX.Y/Core/Statistics/DeployedArtifacts/<имя бизнес-процесса>/ServiceComponents/<имя компонента>/ImplementedPortTypes/<имя portType>:

- *Binding*. Тип binding:
  - DocumentLiteral
  - RpcEncoded
- *HttpUri*. HTTP-URI сервиса.

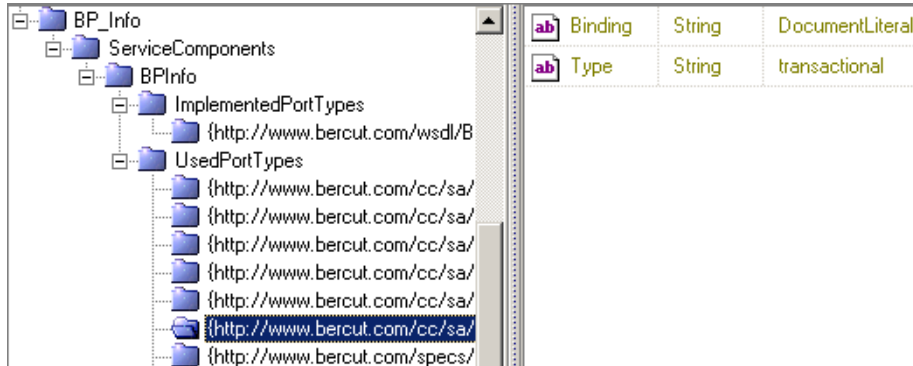


**Рис. 125. Статистика по реализуемым portType**

### Статистика по используемым portType

Система собирает статистику в MIB-группе CMP\_SLES\_vX.Y/Core/Statistics/DeployedArtifacts/<имя бизнес-процесса>/ServiceComponents/<имя компонента>/UsedPortTypes/<имя portType>:

- **Binding.** Тип binding:
  - DocumentLiteral
  - RpcEncoded
- **Type.** Тип portType:
  - regular
  - transactional
  - sessional
  - multicast
  - notification
  - unknown



**Рис. 126. Статистика по используемым portType**

## 9.8. Настройка Java-опций

Для настройки используйте файл `options-usr.sh.template` в каталоге `conf`. Переименуйте этот файл в `options-usr.sh` и отредактируйте его. Заданные настройки применяются перед формированием опций запуска Tomcat.

Вы можете задать настройки:

Название	Тип	Описание
APP_NAME	String	Имя приложения для отображения в дереве MIB. По умолчанию отображает

Название	Тип	Описание
		имя, сформированное по правилам ATOMS. Например, для SLES 3.6 — «CMP_SLES_v3.6».
HEAP_MEM	String	Размер heap-памяти. Резервируется весь объем памяти, чтобы избежать задержек при работе приложения. Это касается только виртуальной памяти, которая будет отображаться в физической по мере выполнения приложения. Это может занять некоторое время. Значение по умолчанию: '1G'.
USE_ALL_MEM_ON_START	String	Если значение — 'on', то JVM при запуске занимает всю физическую память, указанную в параметре HEAP_MEM. Данная настройка уменьшает степень зависимости от системы, время отклика и процессорное время при работе GC. При этом следует учесть, что мониторинг использования физической памяти, занятой процессом, становится неэффективным, так как значение HEAP_MEM всегда будет превышено: размер самой HEAP_MEM + DIRECT_MEM + инфраструктура JVM — потоки, классы, и другие. Необходимо выполнять мониторинг при помощи JMX-подключений к самой JVM, например — использовать приложение JvmMonitor. Значение по умолчанию: 'off'.
DIRECT_MEM	String	Размер direct (off-heap)-памяти, которая необходима для работы с сетью. Этот размер необходим, чтобы избежать возникновения ошибки 'OutOfMemory direct memory'. Значение по умолчанию: '512M'.
MAX_GC_PAUSE	Integer	Максимальная пауза, в миллисекундах. GC G1 избегает превышения этого значения. Значение по умолчанию: '50'. Рекомендуемое значение для сервисов, чувствительных к превышению: '10'.
RESERVE_PERCENT	Integer	Процент зарезервированной памяти. Если объем свободной памяти — меньше, чем задано в данной переменной, то G1 пересматривает начальный процент порога занятости кучи (IHOP) для запуска GC, и, как правило, стремится сразу запустить сбор мусора. Процент можно задать от '0' до '50'. Значение по умолчанию — '25'.
LOG_GC	Boolean	Признак логирования работы GC G1. Значение по умолчанию: 'off'.
NEW_SIZE		Размер памяти, которая может быть использована под молодое поколение. <b>Не рекомендуется</b> задавать данный параметр, если вы не уверены в правильности своих действий. Задание данного параметра <b>отключает эвристику G1</b> , и он перестает

Название	Тип	Описание
		<p>вычислять ИНОР. При этом <b>не гарантируется</b> установка максимального размера паузы в миллисекундах, заданной в переменной MAX_GC_PAUSE. Отсутствие гарантий обусловлено двумя факторами:</p> <ol style="list-style-type: none"> <li>1. Отключение эвристики для расчета ИНОР.</li> <li>2. При переполнении молодого поколения данные будут перенесены в старое поколение, и скорее всего будет вызвана полная сборка мусора.</li> </ol>
DEBUG_ON	Boolean	<p>Включает возможность подключения к приложению отладчиком. Допустимые значения:</p> <ul style="list-style-type: none"> <li>▪ True;</li> <li>▪ False.</li> </ul> <p>Значение по умолчанию — False.</p>
DEBUG_PORT	Integer	<p>Порт для подключения отладчика. Диапазон допустимых значений: от '1' до '65536'. Значение по умолчанию — '5012'.</p>
DEBUG_SUSPEND	String	<p>Допустимые значения: 'y' или 'n'. Включает режим suspend для отладки. Режим означает, что JVM запустится и начнет выполнять приложение только после того, как будет подключен отладчик. Значение по умолчанию — 'n'.</p>
ATLAS_DESC_DISABLE	String	<p>Любая строка в качестве значения выключает использование ATLAS-описания.</p>
ATLAS_BACKUP_REMOVE	Boolean	<p>Флаг удаления резервной копии при миграции версии компонента:</p> <ul style="list-style-type: none"> <li>▪ True — удалять резервную копию;</li> <li>▪ False — не удалять резервную копию (выключено).</li> </ul> <p>Значение по умолчанию — False.</p>
OPTIONS	String	<p>Дополнительные опции для JVM.</p>

Параметры DIRECT\_MEM, HEAP\_MEM, LOG\_GC, MAX\_GC\_PAUSE, NEW\_SIZE, RESERVE\_PERCENT, USE\_ALL\_MEM\_ON\_START используются при работе [сборщика мусора](#) Oracle Garbage Collector (GC) G1.

**⚠ Внимание!** Если параметру `MAX_GC_PAUSE` присвоено небольшое значение, память используется в большем объеме. В результате GC G1 будет выполнять дефрагментацию постепенно, дожидаясь, пока весь регион станет неиспользуемым. Это позволяет более эффективно использовать память и ресурсы процессора.

Для корректного мониторинга нехватки памяти нужно выполнять мониторинг старой памяти и предотвращать частое превышение 100 % — `RESERVE_PERCENT`. Значение `RESERVE_PERCENT` по умолчанию — 25 %. Это позволяет учитывать поведение CMS GC, который обычно при нормальной работе не превышает использования 75 % доступной памяти.

Пример настройки мониторинга: если при нормальном использовании старая память занимает 300 Мб, и значение параметра `RESERVE_PERCENT=25`, то для алармов можно задать 700 Мб с целью утилизации старой памяти и утилизации 90 % всего объема памяти.

## 9.9. Синтаксис, используемый при создании условий

Для задания логических выражений используйте синтаксис XPath 1.0.

При создании условий вы можете использовать следующие функции XPath [21]:

- Булевы выражения: `and`, `or`, `=`, `!=`, `<`, `<=`, `>`, `>=`.
- Булевы функции: `boolean`, `not`, `true`, `false`.
- Числовые выражения: `+`, `-`, `*`, `mod`, `div`.
- Числовые функции: `Number`, `Floor`, `Ceiling`, `Round`.
- Строковые функции: `string`, `concat`, `starts-with`, `contains`, `substring-before`, `substring-after`, `substring`, `string-length`, `normalize-space`, `translate`.
- Специальные числовые значения: `NaN`, `+Infinity`, `-Infinity`.
- Расширения Sun XPath Extension: `current-date` `current-time`, `current-dateTime`.

### Синтаксис с использованием фиктивной переменной `$message`

Для адресации элемента указанием `part` используется синтаксис на основе фиктивной переменной `$message`.

```
$message.[PART_NAME]/[ELEMENT_NAME]
```

Здесь:

- `PART_NAME` — имя `part`;
- `ELEMENT_NAME` — имя элемента.

### Синтаксис с использованием фиктивной переменной `$input`

При использовании старого синтаксиса выполняется подмена выражения с подстановкой имени `part`:

- Выражение `$input/[ELEMENT_NAME]` аналогично записи `$message.[BODY_PART_NAME]/[ELEMENT_NAME]`.
- Выражение `/[ELEMENT_NAME]` аналогично записи `$message.[BODY_PART_NAME]/[ELEMENT_NAME]`.
- Выражение `[ELEMENT_NAME]` аналогично записи `$message.[BODY_PART_NAME]/[ELEMENT_NAME]`.

Здесь `BODY_PART_NAME` — имя `part` типа `body` в сообщении.

**Внимание!** Допускается использовать в выражении только один вид синтаксиса — `$message`, `$input` или же выражение без фиктивной переменной.

## Пример

```
<xsd:complexType name="ManyPartHeaderType">
  <xsd:sequence>
    <xsd:element name="string" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ManyPartRequestType">
  <xsd:sequence>
    <xsd:element name="list" type="xsd:string"/>
    <xsd:element name="string" type="xsd:string"/>
    <xsd:element name="delimiter" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="ManyPartRequest" type="tns:ManyPartRequestType"/>
<xsd:element name="ManyPartHeader" type="tns:ManyPartHeaderType"/>
<message name="ManyPartRequestMessage">
  <part name="header" element="aspm:ManyPartHeader"/>
  <part name="parameters" element="aspm:ManyPartRequest"/>
</message>
```

Адресация элемента `string` в заголовке: `$message.header/string`.

Адресация элемента `list` в теле (body):

- `$message.parameters/list`
- `$input/list`
- `/list`
- `list.`

## 10. Система LW SA Container

Система *LW SA Container* — контейнер на базе Tomcat, который используется для подключения SA к RTSIB. Используется для всех SA, в том числе и SA с доступом к JDBC DataSource, а также JAXWS SA и SADB.

### 10.1. Установка и запуск

Для установки LW SA Container используйте файл `startup.sh`.

Порядок установки:

1. Переключитесь на пользователя *bercut*. Все последующие действия выполняйте от имени этого пользователя.
2. Создайте каталог установки LW SA Container, именуемый дальше `$LWSAContainer`.  
`mkdir LWSAContainer`
3. Скопируйте каталог дистрибутива в созданный каталог.
4. Распакуйте архив установочного дистрибутива.
5. Установите права на запуск скрипт-файлов в каталоге `bin` для всех пользователей.

```
cd /opt/LWSAContainer/bin
```

```
chmod +x *.sh
```

6. Выполните команду:

```
./startup.sh
```

Система автоматически создаст дерево в приложении ATLAS MIB Explorer.

**i Примечание.** При последующих запусках и остановах LW SA Container используйте на выбор:

- *SSM*.
- команду `./startup.sh` или `./shutdown.sh`.

7. Настройте *порты приложений или взаимодействие* между компонентами.
8. Проверьте трейс в приложении ATLAS MIB Explorer на отсутствие ошибок от LW SA Container.
9. Проверьте в MIB, что LW SA Container доступен. Переменная *Status* группы `CMP_LWSAContainer_vX.Y/Core/Statistics` должна иметь значение `available`.

После запуска LW SA Container приступайте к *подключению SA*.

Чтобы задать специальное имя для экземпляра LW SA Container в скрипт-файле `setenv.sh`, расположенном в каталоге `bin`, укажите MIB-имя в параметре `APP_NAME`.

Чтобы удалить созданный резервный файл по окончании *миграции*, в скрипт-файле `setenv.sh`, расположенном в каталоге `bin`, поставьте флаг `-Datlas.removeCreatedBackup`:

```
CATALINA_OPTS=${CATALINA_OPTS}" -Datlas.removeCreatedBackup=true"
```

Для останова LW SA Container выполните команду:

```
./shutdown.sh
```

**Примечание.** Вы можете воспользоваться упрощенной настройкой java-опций. Используйте файл `options-usr.sh.template` в каталоге `conf`. Переименуйте этот файл в `options-usr.sh` и отредактируйте его. Заданные настройки применятся перед формированием опций запуска Tomcat.

## 10.2. Подключение SA

Для подключения SA скопируйте WAR-архив в каталог `$LWSAContainer/webapps`.

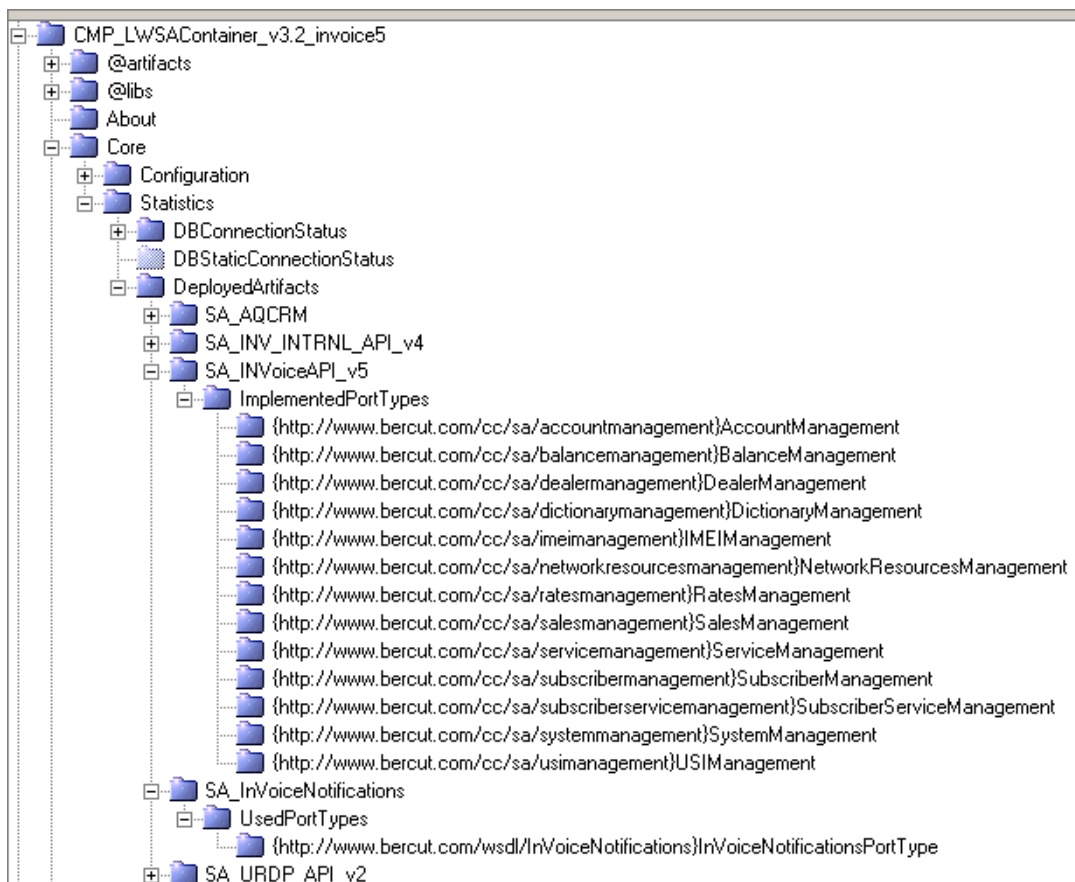
Перед подключением SA проверьте наличие:

- установленной и настроенной системы LW SA Container;
- установленной системы менеджмента, в том числе приложения ATLAS MIB Explorer;
- WAR-файлов подключаемых SA.

Скопируйте WAR-архив в каталог `$LWSAContainer/webapps`.

**Внимание!** Не удаляйте и не заменяйте WAR-файл подключенного SA. На одном LW SA Container можно подключить только один экземпляр SA указанной версии. Несоблюдение указанных правил может привести к аварии.

После подключения SA в группе `CMP_LWSAContainer_vX.Y/Core/Statistics/DeployedArtifacts/` система добавляет группу с именем `<имя SA>`.



**Рис. 127. Отображение в MIB статистики подключенного SA**

В группу `CMP_LWSAContainer_vX.Y/Core/Statistics/DeployedArtifacts/<имя SA>/ImplementedPortTypes/` система добавляет имя реализованного интерфейса `{<targetNameSpace><portType name>`.



Информацию о корректности работы SA смотрите в переменной *Status* группы CMP\_LWSAContainer\_vX.Y/@artifacts/<имя SA>/Core/Statistics. При корректной работе установлено значение **available**.

### 10.3. Настройка источников данных

Для инициализации БД задайте ее параметры в MIB-группе CMP\_LWSAContainer\_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>.

LW SA Container может работать с транзакционными и нетранзакционными SA. Транзакционные SA используют пул соединений, поддерживающих распределенные XA-транзакции. Отдельной настройки требуют *JMS подключения*, выполняющие прием JMS-сообщений от сервера Oracle.

LW SA Container работает со следующими СУБД, в среде которых развернуты БД:

- PostgreSQL
- Oracle
- Oracle timesten.

Тип СУБД определяет параметр *ConnectionType*.

**! Внимание!** Перед началом работы обновите драйверы JDBC и JMS.

1. При использовании СУБД Oracle замените следующие библиотеки в каталоге <каталог установки LW SA Container>/lib на библиотеки из используемой версии СУБД Oracle:

- *aqapi.jar*. Каталог \$ORACLE\_HOME/rdbms/jlib
- *ojdbc6.jar*. Каталог \$ORACLE\_HOME/jdbc/lib
- *orai18n.jar*. Каталог \$ORACLE\_HOME/jlib.

2. При использовании СУБД Oracle timesten замените библиотеку *ttjdbc7.jar* в каталоге <каталог установки LW SA Container>/lib на библиотеку из версии клиента СУБД Oracle timesten.

3. При использовании СУБД PostgreSQL замените библиотеку *postgresql-номер-версии.jar* в каталоге <каталог установки LW SA Container>/lib на библиотеку из используемой версии СУБД PostgreSQL.

LW SA Container позволяет использовать два DataSource к разным БД в рамках одной транзакции.

Количество доступных соединений с БД зависит от количества рабочих потоков для SA. Количество рабочих потоков на portType в каждом SA не должно превышать максимальное количество доступных соединений для DataSource. Если настроен *мониторинг статуса подключения к БД*, увеличьте количество соединений на единицу.

**i Примечание.** Не забывайте учитывать нагрузку для Participant, если SA работает по транзакционной схеме.

Настройки рабочих потоков для portType хранятся в *MIB-группе* CMP\_LWSAContainer\_vX.Y/@libs/MessageBus/Core/Configuration/ComponentSettings/Settings/<имя portType>. Если настройки для portType отсутствуют, система использует значение по умолчанию для количества рабочих потоков: '5'.

### 10.3.1. Настройка подключений к схеме данных

Для настройки подключения к схеме данных задайте переменные MIB-группы `CMP_LWSAContainer_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>`.

Порядок настройки соединений со схемой данных:

1. Создайте вложенную группу с именем источника данных в группе `CMP_LWSAContainer_vX.Y/DataSources/ConnectionsList`.

Формат:

- для нетранзакционных SA: `jdbc/<имя источника данных>`;
- для транзакционных SA: `jdbc/<имя ха-пула>`. Установите в True переменную `XATransactionsEnabled` группы `CMP_LWSAContainer_vX.Y/Core/Configuration`.

Например: `jdbc/audit` или `jdbc/xaspm`.

**! Внимание!** Имена источников данных разных БД не должны совпадать в рамках одного экземпляра LW SA Container.

2. Задайте значения обязательных переменных группы `CMP_LWSAContainer_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc`:

- **Enabled.** Флаг включения источника данных:

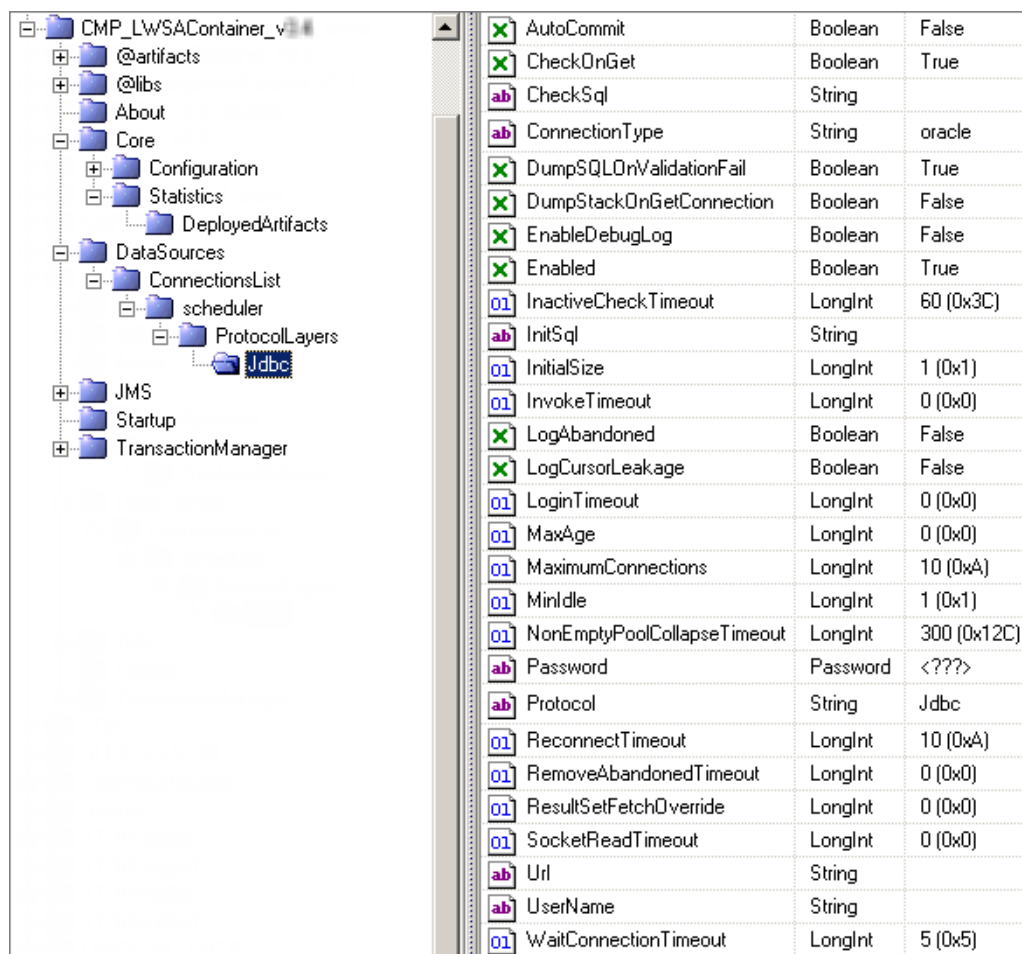
- True — источник данных включен.
- False — источник данных выключен.

Значение по умолчанию: True.

- **ConnectionType.** Тип СУБД, в среде которой развернута схема данных:
  - oracle
  - postgres
  - timesten

Значение по умолчанию: 'oracle'.

- **Url.** URL для подключения к БД в формате JDBC. Например, URL для подключения к БД oracle по тонкому клиенту: `jdbc:oracle:thin:@<IP-адрес или доменное имя сервера, на котором установлена БД>:<номер порта>:<сервисное имя БД>`.
- **UserName.** Имя пользователя, заданное при создании схемы данных.
- **Password.** Пароль пользователя, заданный при создании схемы данных.
- **Protocol.** Наименование протокола. Значение должно быть 'Jdbc'.



**Рис. 128. Группа CMP\_LWSAContainer\_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc**

3. При необходимости задайте значения необязательных переменных группы CMP\_LWSAContainer\_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc:

- **InitialSize.** Количество подключений к БД при старте пула. Значение по умолчанию: '1'.
- **MinIdle.** Минимальное количество свободных подключений к БД. Значение по умолчанию: '1'.

Пример: Под нагрузкой всегда занято 4 соединения. При значении по умолчанию пул будет держать еще одно свободное соединение для обеспечения всплеска нагрузки.

- **MaximumConnections.** Максимальное количество подключений к БД. Если все соединения заняты и достигнуто максимальное количество подключений к БД, система через **WaitConnectionTimeout** формирует исключение для пользовательского кода, ожидающего получения подключения. Значение по умолчанию: '10'.
- **WaitConnectionTimeout.** Тайм-аут ожидания подключения при пустом пуле подключений, в секундах. Значение по умолчанию: '5'.
- **LoginTimeout.** Максимальное время установления подключения, в секундах. При значении '0' время не ограничено. Значение по умолчанию: '0'.

- *SocketReadTimeout*. Максимальный тайм-аут ожидания ответа от БД, в секундах. При значении '0' тайм-аут отсутствует. После окончания тайм-аута подключение считается неуспешным. Текущее действие через данное подключение завершается ошибкой, подключение закрывается и уничтожается при возвращении в пул. Значение по умолчанию: '0'.
- *InvokeTimeout*. Максимальное время выполнения SQL-логики, в секундах. При значении '0' время не ограничено. При превышении значения текущая логика завершается на стороне БД, код клиента получает ошибку. Соединение остается активным и работа может продолжаться. Не устанавливайте значение выше, чем *SocketReadTimeout*. Значение по умолчанию: '0'.
- *ReconnectTimeout*. Тайм-аут ожидания установки подключения после неуспешной попытки, в секундах. Если БД недоступна, пул будет выполнять попытки подключения не чаще, чем указанный тайм-аут. Значение по умолчанию: '10'.
- *NonEmptyPoolCollapseTimeout*. Тайм-аут, после окончания которого система уменьшает количество подключений в пуле, если они не используются. Тайм-аут указывается в секундах. Если в течение тайм-аута количество свободных подключений в пуле больше значения *MinIdle*, система закрывает одно из подключений. Значение по умолчанию: '300'.
- *InactiveCheckTimeout*. Период проверки работоспособности свободных подключений, в секундах. При значении '0' система не проверяет работоспособность. Значение по умолчанию: '60'.
- *RemoveAbandonedTimeout*. Тайм-аут подключения бизнес-приложения, в минутах. После окончания тайм-аута система закрывает подключение как ошибочное и удаляет его из пула. При значении '0' тайм-аут не задается. Значение по умолчанию: '0'.
- *MaxAge*. Максимальное время продолжительности соединения, в минутах. При превышении значения система закрывает соединение для повторного пересоздания, если это необходимо. Значение по умолчанию: '0'.
- *CheckSql*. SQL-выражение для проверки подключения. Если значение не задано, подключение проверяется стандартным запросом в зависимости от типа БД. Для oracle значение: `select 13 from dual`. Для postgres значение: `select 13`. Значение по умолчанию зависит от типа БД.
- *ResultSetFetchOverride*. Переопределение количества зачитываемых строк в *ResultSet* за один раз. При значении '0' система не переопределяет количество строк. Значение по умолчанию: '0'.
- *LogAbandoned*. Флаг вывода логов по подключению артефактов после окончания тайм-аута *AbandonTimeout*:
  - True — система собирает java-стеки подключения методов и выводит логи после окончания тайм-аута *AbandonTimeout*.
  - False — после окончания тайм-аута *AbandonTimeout* система не выводит логи.

Значение по умолчанию: False.

- *LogCursorLeakage*. Флаг логирования стеков и идентификаторов потоков при открытии курсора и не закрытии после возвращения подключения:
  - True — система логирует стеки и идентификаторы потоков.
  - False — система не логирует стеки и идентификаторы потоков.

Значение по умолчанию: False.

- *CheckOnGet*. Флаг проверки подключения перед отправкой приложению клиента:

- True — система проверяет подключение с помощью контрольного запроса перед каждой выдачей соединения клиентскому приложению.
- False — система не проверяет подключение перед выдачей соединения клиентскому приложению.

Значение по умолчанию: True.

- *AutoCommit*. Флаг автоматического применения:
  - True — автоматическое применение включено.
  - False — автоматическое применение отключено.

Значение по умолчанию: False.

- *DumpSQLOnValidationFail*. Флаг отображения выполняемого SQL-скрипта при неуспешной проверке активности потока:
  - True — SQL-скрипт отображается.
  - False — SQL-скрипт не отображается.

Значение по умолчанию: True.

- *DumpStackOnGetConnection*. Флаг отображения стека при подключении из пула соединений:
  - True — стек отображается.
  - False — стек не отображается.

Значение по умолчанию: False.

- *EnableDebugLog*. Флаг отображения сообщений уровня Debug при соответствующем значении параметра *LogLevel*:
  - True — сообщения уровня Debug отображаются.
  - False — сообщения уровня Debug не отображаются.

Значение по умолчанию: False.

- *InitSql*. SQL-скрипт, который выполняется один раз после создания физического подключения к БД.

#### 4. *Перезапустите* LW SA Container.

Статистика пула соединений с БД отображается в группе `CMP_LWSAContainer_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/Statistics/StateInfo`:

- *Status*. Статус источника данных:
  - Available — источник данных подключен.
  - Unavailable — источник данных недоступен.
- *ConnectionsCountAvg*. Среднее количество подключений к источнику данных за статистический интервал.
- *ConnectionsCountMax*. Максимальное количество подключений к источнику данных за статистический интервал.
- *ConnectionsCountMin*. Минимальное количество подключений к источнику данных за статистический интервал.
- *FreeConnectionsCountAvg*. Среднее количество свободных подключений к источнику данных за статистический интервал.
- *FreeConnectionsCountMax*. Максимальное количество свободных подключений к источнику данных за статистический интервал.

- *FreeConnectionsCountMin*. Минимальное количество свободных подключений к источнику данных за статистический интервал.
- *QueryCounter*. Общее количество отправленных запросов.
- *QueryRateAvg*. Среднее количество отправленных запросов в секунду за статистический интервал.
- *QueryRateMax*. Максимальное количество отправленных запросов в секунду за статистический интервал.
- *RejectCounter*. Общее количество отклоненных ответов.
- *RejectRateAvg*. Среднее количество отклоненных ответов в секунду за статистический интервал.
- *RejectRateMax*. Максимальное количество отклоненных ответов в секунду за статистический интервал.
- *ResponseCounter*. Общее количество полученных ответов.
- *ResponseRateAvg*. Среднее количество полученных ответов в секунду за статистический интервал.
- *ResponseRateMax*. Максимальное количество полученных ответов в секунду за статистический интервал.
- *TimeoutCounter*. Общее количество тайм-аутов на запросы.
- *TimeoutRateAvg*. Среднее количество тайм-аутов на запросы в секунду за статистический интервал.
- *TimeoutRateMax*. Максимальное количество тайм-аутов на запросы в секунду за статистический интервал.
- *QueryResponseTimeAvg*. Средний период ожидания ответа на запрос за статистический интервал.
- *QueryResponseTimeMax*. Максимальный период ожидания ответа на запрос за статистический интервал.

01	ConnectionsCountAvg	Dword	1 (0x1)
01	ConnectionsCountMax	Dword	1 (0x1)
01	ConnectionsCountMin	Dword	1 (0x1)
01	FreeConnectionsCountAvg	Dword	1 (0x1)
01	FreeConnectionsCountMax	Dword	1 (0x1)
01	FreeConnectionsCountMin	Dword	1 (0x1)
01	QueryCounter	Dword64	2686 (0xA7E)
01	QueryRateAvg	Dword64	0 (0x0)
01	QueryRateMax	Dword64	2 (0x2)
01	QueryResponseTimeAvg	Dword	1 (0x1)
01	QueryResponseTimeMax	Dword	1 (0x1)
01	RejectedCounter	Dword64	0 (0x0)
01	RejectedRateAvg	Dword64	0 (0x0)
01	RejectedRateMax	Dword64	0 (0x0)
01	ResponseCounter	Dword64	2686 (0xA7E)
01	ResponseRateAvg	Dword64	0 (0x0)
01	ResponseRateMax	Dword64	2 (0x2)
ab	Status	String	Available

**Рис. 129. Группа CMP\_LWSAContainer\_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/Statistics/StateInfo**

### 10.3.2. Настройка JMS-подключений СУБД Oracle

Для настройки JMS-подключений используйте переменные MIB-группы `CMP_LWSAContainer_vX.Y/JMS/ConnectionsList`.

Модуль JMS Notification обрабатывает нотификации из БД Oracle в приложениях, развернутых в среде LW SA Container. При настройке JMS-очереди LW SA Container предоставляет упрощенную инфраструктуру подписки на JMS-сообщения для определенного сервиса.

Порядок настройки JMS-нотификаций:

1. Добавьте вложенную группу с именем источника данных, заданным при настройке подключения к БД, в группу `CMP_LWSAContainer_vX.Y/JMS/ConnectionsList`.
2. **! Внимание!** Имена источников данных не должны совпадать в рамках одного экземпляра LW SA Container.

Задайте значения переменных группы `CMP_LWSAContainer_vX.Y/JMS/ConnectionsList/<имя источника данных>/ProtocolLayers/JMS`:

- **AutoRollbackTime.** Тайм-аут автоматической отправки rollback на входящее JMS-сообщение. Значения меньше '0' означают, что автоматическая отправка отключена. Значение задается в миллисекундах. Значение по умолчанию: '5000'.
- **ConsumingMode.** Режим считывания JMS-сообщений из очереди:
  - `sequential` — система последовательно обрабатывает сообщения. Отправка нового сообщения клиенту выполняется после его выхода из метода нотификации.
  - `parallel` — вычитка сообщений из БД выполняется параллельными потоками, система вызывает метод нотификации SA. Количество потоков определяется параметром `ParallelConsumerSize`. Подтверждение `commit` или `rollback` система отправляет на все считанные из очереди на текущий момент времени сообщения в рамках одной сессии или одного подключения. Если `ParallelConsumerSize=1`, режим аналогичен режиму `sequential`.

Значение по умолчанию: 'sequential'.

- **DataSourceName.** *JNDI-имя* для подключения JMS.

**i Примечание.** JMS-подключения выбираются из пула и не возвращаются в него, пока подписчик — SA с реализацией интерфейса `JmsNotificationListener` — не отпишется от нотификаций. При задании параметров группы `CMP_LWSAContainer_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc` соблюдайте следующие условия:

- Включен **мониторинг** в группе `CMP_LWSAContainer_vX.Y/Core/Configuration/DBConnectionMonitoring`. **Параметр** `MonitoringEnabled=True`.
- Значение `MaximumConnections` должно быть больше значения `JdbcConnectionsCount`.
- `CheckOnGet=True`.
- `RemoveAbandonedTimeout=0`.
- Пулы соединений для SA и JMS разделены.

- **Enabled.** Статус подключения текущего JMS-соединения:

- True — JMS-соединение подключено.
- False — JMS-соединение отключено.

Значение по умолчанию: 'True'.

- *FastMessageTimeToLiveSec*. Время жизни «быстрых» сообщений, в секундах. При значении '-1' время не ограничено. Сообщение считается «быстрым», если время его жизни меньше или равно значению переменной. Значение по умолчанию: '60'. Параметр используется для метода `JmsMessage.isFastMessage()` в API, предоставляемом для SA.
- *JdbcConnectionsCount*. Количество используемых подключений к БД. Значение по умолчанию: '5'. Получение нотификаций по каждому соединению выполняется независимо и параллельно в отдельных рабочих потоках. При *ConsumingMode*=*sequentia* система отправляет *commit* или *rollback* только на данное сообщение. При *ConsumingMode*=*parallel* система отправляет *commit* или *rollback* на все полученные по данному соединению сообщения.
- *JmsQueueName*. Имя JMS-очереди сообщений.
- *KeepAliveTimeout*. Тайм-аут проверки активности потока для запросов, которые поступили на указанный *portType*. Указывается в секундах. Значение по умолчанию: '5'.
- *McAllowNewSubscribers*. Выбор подписчика к *multiconsumer*-очереди:
  - False — подписка к *multiconsumer*-очереди подписчика, имя которого зарегистрировано в БД.
  - True — возможность создания подписчика к *multiconsumer*-очереди, имя которого не зарегистрировано в БД.
- *McSubscriberType*. Тип JMS-потребителей для *multiconsumer*-очереди:
  - *simple* — доставка сообщений в режиме радио. Подписчик получает нотификации, которые были отправлены, пока поддерживалось физическое подключение к БД.
  - *durable*— гарантированная доставка получателю при подписке. При наличии подписки и отсутствии физического подключения система доставляет сообщения подписчику после восстановления подключения.
- *McUnSubscribe*. Флаг отмены подписки при отключении:
  - True — отмена подписки при отключении.
  - False — система оставляет подписку с именем *SubscriberName* в очереди при отключении.

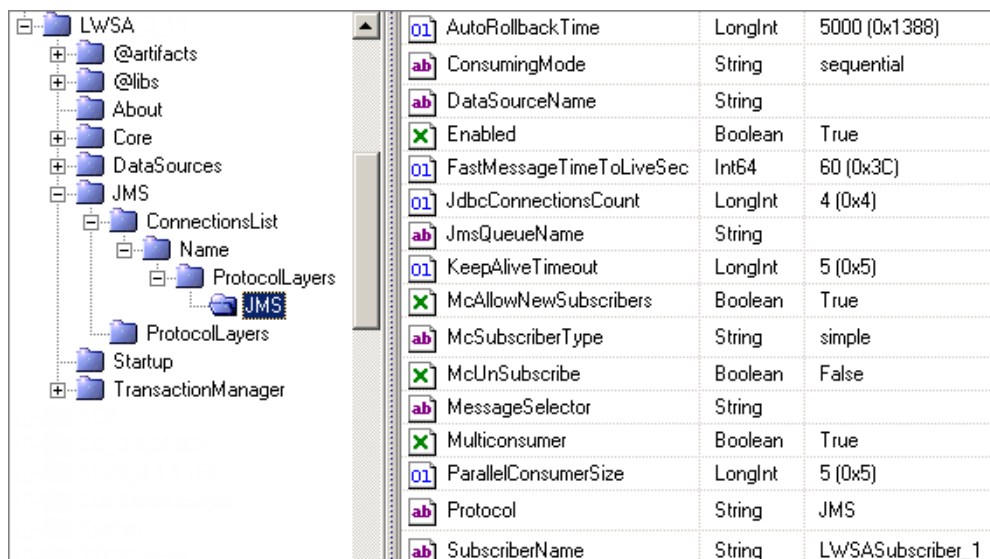
Значение по умолчанию: 'False'.

- *MessageSelector*. Выражение, которое задает условие выборки сообщений из очереди для данного подписчика. При пустом значении подписчику доставляются все сообщения. Пример значения: `JMSPriority = 2 OR JMSCorrelationID = 'corr1'`.
- *Multiconsumer*. Тип очереди, к которой выполняется подключение:
  - True — у очереди может быть несколько подписчиков (JMS listener), она является *multiconsumer*-очередью. Названия настроек такой очереди начинаются с **Mc**.
  - False — у очереди может быть только один подписчик, она является *singleconsumer*-очередью. Названия настроек такой очереди начинаются с **Sc**.

Значение по умолчанию: 'True'.



- *ParallelConsumerSize*. Размер пула потоков в режиме ConsumingMode.parallel. При значении '0' пул не ограничен. Значение по умолчанию: '5'.
- *Protocol*. Наименование протокола. Значение должно быть 'JMS'.
- *SubscriberName*. Имя durable-JMS-подписчика multiconsumer-очереди, который использует данное JMS-подключение. Значение должно быть уникальным среди других имен подписчиков данной очереди.



**Рис. 130.** Группа CMP\_LWSAContainer vX.Y/JMS/  
ConnectionsList/<имя источника данных>/ProtocolLayers/JMS

#### **i** **Примечание.**

1. Мониторинг статуса подключения к БД должен быть активен.
2. Настройте количество подключений в пуле *maxActive* больше значения *JdbcConnectionsCount*.
3. Задайте следующие параметры пула:
  - параметр *testOnBorrow* — значение *True*.
  - параметр *testWhileIdle* — значение *True*.
  - параметр *removeAbandoned* — значение *False*.
4. Не используйте пул, сконфигурированный для JMS, в SA для SQL-запросов. Разделяйте пулы соединений для SA и для JMS.

3. Сохраните внесенные изменения и **перезапустите** LW SA Container.

### **10.3.2.1. Алгоритм работы SA с очередью нотификаций**

Алгоритм работы:

1. SA подключается к очереди, которая задана в настройках JMS.
2. SA обрабатывает сообщение и пытается отправить его по указанному маршруту в систему, которая должна получить нотификацию:
  - если *portType* доступен и сообщение успешно доставлено, SA отправляет в БД — *confirm*. Если при попытке отправки по маршруту возвращается ошибка,

например, корреляции нет на удаленном компоненте, система выполняет откат (rollback) сообщения в JMS-очереди.

- если при отправке сообщения произошла ошибка, SA отправляет в БД отказ отправки сообщения (reject).

3. SA ожидает ответа от удаленной стороны в течение тайм-аута.

### 10.3.2.2. Нотификационная схема БД

Компонент Oracle JMS Queues содержит скрипт-файл установки схемы БД для работы с очередями Oracle.

Для установки схемы данных выполните следующие действия:

1. Измените значения следующих параметров в файле `define.sql`:

- `DB_ALIAS` — имя базы данных, на которую выполняется установка.
- `SRV_NAME` — имя сервиса, для которого создаются очереди. Желательно, чтобы имя сервиса совпадало с настройкой `service_name` для JMS в LW SA Container.

Файл `define.sql` расположен в каталоге `install`.

2. Выполните скрипт-файл `install.cmd`.

Скрипт-файл создает двух пользователей:

- администратора очередей — `jmsadm`;
- пользователя — `jmsusr`.

Администратору доступен установочный пакет `OJA_API`, который содержит процедуры управления очередями:

- удаление;
- создание;
- остановка;
- запуск;
- отправка сообщения в очередь.

Пользователь выполняет настройку JMS в LW SA Container.

После выполнения скрипт-файла запускает следующие очереди:

- `src_q_<srv_name>` — исходная очередь;
- `pri_q_<srv_name>` — основная очередь;
- `sec_q_<srv_name>` — резервная очередь.

Система распределяет сообщения из исходной очереди, которая нагружается подсистемой в БД, в основную и резервную. К основной и резервной очереди подключается два обработчика JMS-сообщений, которые развернуты в двух разных контейнерах.

### Особенности отправки сообщения в очередь

Для отправки сообщения в очередь используется процедура `DBMS_AQ.ENQUEUE`. Входные параметры:

- `queue_name` — имя очереди;
- `enqueue_options` — параметры, которые доступны для операции постановки в очередь;
- `message_properties` — информация, которую Oracle Database Advanced Queuing использует для управления отдельными сообщениями;
- `payload` — полезная нагрузка, которая публикуется в анонимной подписке.

Выходной параметр `msgid` — идентификатор сообщения.

Для входных структур `enqueue_options` и `message_properties` учитывайте следующие особенности:

- `enqueue_options.visibility` — определяет транзакционное поведение отправляемого сообщения. Задайте значение `'dbms_aq.on_commit'`, которое означает, что отправка сообщения будет частью текущей транзакции.
- `message_properties.user_property` — дополнительный атрибут для передачи опциональной информации. Рекомендуется не использовать, так как значение данного атрибута может привести к ошибке `'Protocol violation'`.

Вызов процедуры `DBMS_AQ.ENQUEUE` должен быть завершен командой `commit` для закрытия текущей транзакции.

### 10.3.3. Настройка мониторинга статуса подключения к БД

Для настройки мониторинга статуса подключений к БД и нотификации SA об изменении статуса используйте переменные MIB-группы `CMP_LWSAContainer_vX.Y/Core/Configuration/DBConnectionMonitoring/DataSources`.

Порядок настройки мониторинга:

1. Установите значение `True` для переменной `MonitoringEnabled` группы `CMP_LWSAContainer_vX.Y/Core/Configuration/DBConnectionMonitoring`. При значении `False` мониторинг подключения всех БД будет отключен и SA не будут получать нотификации при изменении статуса подключения.
2. Создайте вложенную группу с именем `jdbc/<имя источника данных, заданное при подключении БД>` в группе `CMP_LWSAContainer_vX.Y/Core/Configuration/DBConnectionMonitoring/DataSources`.

В созданной группе задайте значения следующих переменных:

- `CheckQuery`. SQL-запрос для проверки состояния БД:
  - `select 1 from dual` — для СУБД Oracle.
  - `select 1` — для PostgreSQL.
- `MonitoringEnabled`. Флаг включения или выключения мониторинга подключения БД. Рекомендованные значения:
  - Для базовой фабрики XA-пула в `False`.
  - Для JMS-подключений в `True`.
- `SuccessCheckPeriod`. Тайм-аут между проверками статуса подключения к БД при успешной предыдущей проверке. Тайм-аут задается в секундах.
- `CheckQueryTimeout`. Тайм-аут выполнения SQL-запроса для проверки состояния БД, в секундах. При значении `'0'` время выполнения не ограничено.
- `FailCheckPeriod`. Тайм-аут между проверками статуса подключения к БД при неуспешной предыдущей проверке. Тайм-аут задается в секундах.



**Рис. 131. Группа CMP\_LWSAContainer\_vX.Y/Core/Configuration/DBConnectionMonitoring/DataSources/jdbc/<имя источника данных, заданное при подключении БД>**

Статистику статуса подключения смотрите в переменных группы CMP\_LWSAContainer\_vX.Y/Core/Statistics/DBConnectionStatus/<имя источника данных, заданное при подключении схемы данных>:

- **ConnectionState.** Статус подключения к БД:
  - 1 — соединение установлено.
  - 0 — соединение разорвано.
- **LastConnectTime.** Дата и время подключения при последней проверке.
- **LastFailCause.** Причина последней неуспешной проверки.
- **LastFailTime.** Дата и время последней неуспешной проверки.
- **LastQueryExecutionTime.** Продолжительность последней проверки.
- **LastSuccessTime.** Дата и время последней успешной проверки.
- **LastWarning.** Предупреждение, возникшее в момент последней успешной проверки.
- **QueryConnectTimeAvg.** Средняя продолжительность подключения к БД.
- **QueryConnectTimeMax.** Максимальная продолжительность подключения к БД.
- **QueryConnectTimeMin.** Минимальная продолжительность подключения к БД.
- **QueryConnectTotalTime.** Суммарная продолжительность подключения к БД.
- **QueryExecutionTimeAvg.** Средняя продолжительность проверки подключения к БД.
- **QueryExecutionTimeMax.** Максимальная продолжительность проверки подключения к БД.
- **QueryExecutionTimeMin.** Минимальная продолжительность проверки подключения к БД.
- **QueryExecutionTotalTime.** Суммарная продолжительность проверки подключения к БД.

Имя статистики	Тип	Значение
01 ConnectionState	LongInt	1 (0x1)
01 LastConnectTime	Int64	0 (0x0)
01 LastQueryExecutionTime	Int64	2 (0x2)
LastSuccessTime	DateTime	06.04.2016 15:32:29
ab LastWarning	String	
01 QueryConnectTimeAvg	Int64	1 (0x1)
01 QueryConnectTimeMax	Int64	1 (0x1)
01 QueryConnectTimeMin	Int64	1 (0x1)
01 QueryConnectTotalTime	Int64	155 (0x9B)
01 QueryExecutionTimeAvg	Int64	1 (0x1)
01 QueryExecutionTimeMax	Int64	1 (0x1)
01 QueryExecutionTimeMin	Int64	1 (0x1)
01 QueryExecutionTotalTime	Int64	1520 (0x5F0)

**Рис. 132.** CMP\_LWSAContainer\_vX.Y/Core/Statistics/DBConnectionStatus/  
 <имя источника данных, заданное при подключении схемы данных>

3. Сохраните внесенные изменения и [перезапустите](#) LW SA Container.

#### 10.3.4. Соответствие параметров Tomcat DataSource и MIB

Соответствие параметров файла `server.xml` и MIB-группы `CMP_LWSAContainer_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>`.

**Таблица 3**

Параметр файла <code>server.xml</code>	MIB-параметр	Описание параметра
name	имя MIB-группы CMP_LWSAContainer_vX.Y/ DataSources/ ConnectionsList/ <jdbc-имя источника данных>	JDBC-имя источника данных.
username	UserName	Имя пользователя схемы данных.
password	Password	Пароль пользователя схемы данных.
url	Url	URL для подключения к БД в формате JDBC, характерном для используемого типа БД.
defaultAutoCommit	AutoCommit. Значение по умолчанию: True.	Флаг включения автоматического применения изменений (commit).
maxWait	WaitConnectionTimeout	Максимальное время ожидания доступного соединения из пула.
initialSize	InitialSize	Количество соединений, создаваемых в момент запуска пула.

Параметр файла <i>server.xml</i>	MIB-параметр	Описание параметра
maxActive	MaximumConnections	Максимальное количество активных соединений
maxIdle	При наличии переменной maxActive (MaximumConnections) данная настройка не имеет смысла и убрана из MIB-параметров.	Количество соединений, которые постоянно присутствуют в пуле.
minIdle	MinIdle	Минимальное количество свободных подключений к БД.
validationQuery	CheckSql	SQL-выражение, по которому проверяется соединение с базой данных.
testOnBorrow	CheckOnGet	Флаг включения проверки наличия свободных соединений в пуле в момент установки соединения с приложением.
testWhileIdle	Параметр убран, достаточно проверить наличие свободных соединений в момент изменения их количества.	Флаг включения периодической проверки наличия свободных соединений в пуле. Частота проверки задается в параметре <i>timeBetweenEvictionRunsMillis</i> .
timeBetweenEvictionRunsMillis	Параметр убран, достаточно проверить наличие свободных соединений в момент изменения их количества.	Период (в миллисекундах) выполнения проверки наличия свободных соединений в пуле. Используется при включенном параметре <i>testWhileIdle</i> .
removeAbandoned	Лишний параметр, достаточно <i>RemoveAbandonedTimeout</i> . При <i>RemoveAbandonedTimeout</i> > 0, удаление соединения из пула включено, при <i>RemoveAbandonedTimeout</i> = 0 — выключено.	Флаг удаления соединения из пула после окончания тайм-аута <i>removeAbandonedTimeout</i> .
removeAbandonedTimeout	RemoveAbandonedTimeout	Период с момента выдачи соединения из пула до момента, когда соединение перестало быть в пуле, в секундах. Значение '0' — соединение не удаляется из пула.
validationQueryTimeout	В MIB этот тайм-аут не настраивается, используется	Тайм-аут проверки соединения с БД, в секундах.

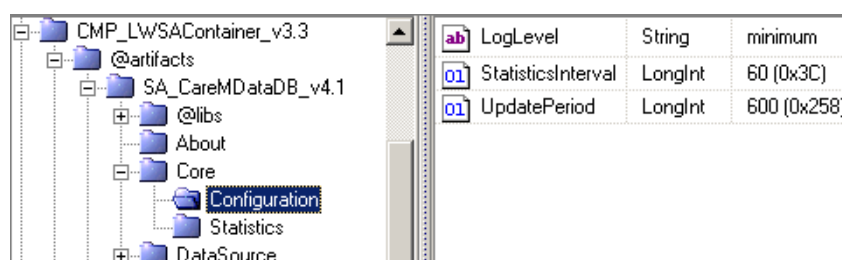
Параметр файла server.xml	MIB-параметр	Описание параметра
	общий тайм-аут для всех типов логик — InvokeTimeout	После окончания тайм-аута система считает запрос неуспешным.
validationInterval	InactiveCheckTimeout. Значение указывается в секундах.	Минимальная частота проверки соединения с БД.
logAbandoned	LogAbandoned	Флаг сбора стеков при подключении.
suspectTimeout	Параметр убран. При достаточной частоте и минимальной сложности проверки параметр не нужен.	Тайм-аут проверки пула на «забытые» подключения, в секундах. Работает при removeAbandoned=False.
maxAge	MaxAge	Максимальное время продолжительности соединения, в минутах.
dataSourceJNDI	Параметр убран, MIB-datasource работает только с предустановленными типами БД.	Имя базовой фабрики ХА-пула.
Параметр отсутствовал	LoginTimeout	Максимальное время установления подключения, в секундах. При значении '0' время не ограничено.
Параметр отсутствовал, настройка выполнялась через опции запуска JVM	SocketReadTimeout	Максимальный тайм-аут ожидания ответа от БД.
Параметр отсутствовал, настройка выполнялась через опции запуска JVM	InvokeTimeout	Максимальное время выполнения SQL-логики, общее для всех логик.
Параметр отсутствовал	ReconnectTimeout	Тайм-аут ожидания установки подключения после неуспешной попытки — частота попыток подключения.
Параметр отсутствовал	NonEmptyPoolCollapseTimeout	Тайм-аут, после окончания которого система уменьшает количество подключений в пуле на единицу, если они не используются. Чем больше значение, тем больше реакционность пула: размер пула уменьшается медленнее.
Параметр отсутствовал	ResultSetFetchOverride	Переопределение количества строк, зачитываемых в ResultSet за один раз.
Параметр отсутствовал	LogCursorLeakage	Флаг логирования стеков и идентификаторов потоков при появлении алармов о незакрытом курсоре.

Параметр файла <i>server.xml</i>	MIB-параметр	Описание параметра
Параметр отсутствовал	ConnectTimeout	Максимальное время установки соединения с БД.

## 10.4. Настройка конфигурационных параметров SA

Настройте конфигурационные параметры SA в MIB-группах: *CMP\_LWSAContainer\_vX.Y/@artifacts/<название SA>\_<версия SA>/Core/Configuration*.

Каждый SA может иметь специфический набор конфигурационных параметров. Настройте конфигурационные параметры в MIB-группах *CMP\_LWSAContainer\_vX.Y/@artifacts/<название SA>\_<версия SA>/Core/Configuration*.



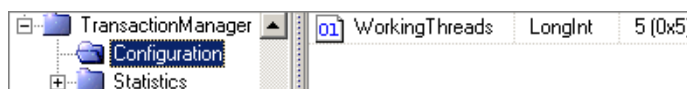
**Рис. 133. Параметры SA**

## 10.5. Менеджер транзакций

Менеджер транзакций управляет распределенными и автономными транзакциями LWSA Container.

Менеджер транзакций выполняет подготовку, фиксацию или отмену распределенных транзакций.

Для настройки менеджера транзакций задайте количество потоков для обработки транзакций в параметре *WorkingThreads* группы *LWSA\_X.Y/TransactionManager/Configuration*.



**Рис. 134. Настройка менеджера транзакций**

### Статистика менеджера транзакций

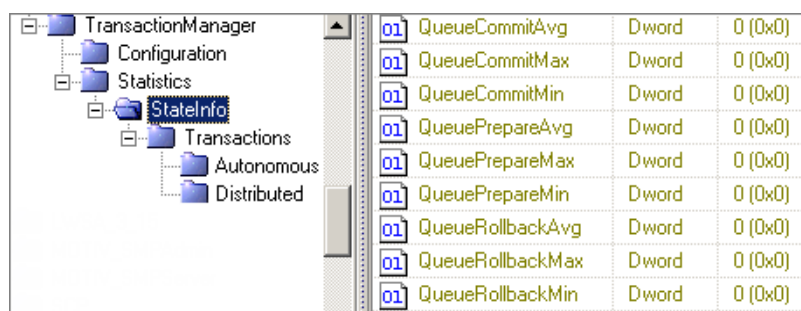
Статистические параметры позволяют скорректировать настройки менеджера транзакций и настройки источников данных для оптимальной работы сервисов.

Статистика по количеству сигналов в очереди на обработку отображается в группе *LWSA\_X.Y/TransactionManager/Statistics/StateInfo*:

- *QueueCommitAvg* — среднее количество сигналов на фиксацию транзакции в очереди менеджера транзакций;



- *QueueCommitMax* — максимальное количество сигналов на фиксацию транзакции в очереди менеджера транзакций;
- *QueueCommitMin* — минимальное количество сигналов на фиксацию транзакции в очереди менеджера транзакций;
- *QueuePrepareAvg* — среднее количество сигналов на подготовку транзакции в очереди менеджера транзакций;
- *QueuePrepareMax* — максимальное количество сигналов на подготовку транзакции в очереди менеджера транзакций;
- *QueuePrepareMin* — минимальное количество сигналов на подготовку транзакции в очереди менеджера транзакций;
- *QueueRollbackAvg* — среднее количество сигналов на отмену транзакции в очереди менеджера транзакций;
- *QueueRollbackMax* — максимальное количество сигналов на отмену транзакции в очереди менеджера транзакций;
- *QueueRollbackMin* — минимальное количество сигналов на отмену транзакции в очереди менеджера транзакций.



01	QueueCommitAvg	Dword	0 (0x0)
01	QueueCommitMax	Dword	0 (0x0)
01	QueueCommitMin	Dword	0 (0x0)
01	QueuePrepareAvg	Dword	0 (0x0)
01	QueuePrepareMax	Dword	0 (0x0)
01	QueuePrepareMin	Dword	0 (0x0)
01	QueueRollbackAvg	Dword	0 (0x0)
01	QueueRollbackMax	Dword	0 (0x0)
01	QueueRollbackMin	Dword	0 (0x0)

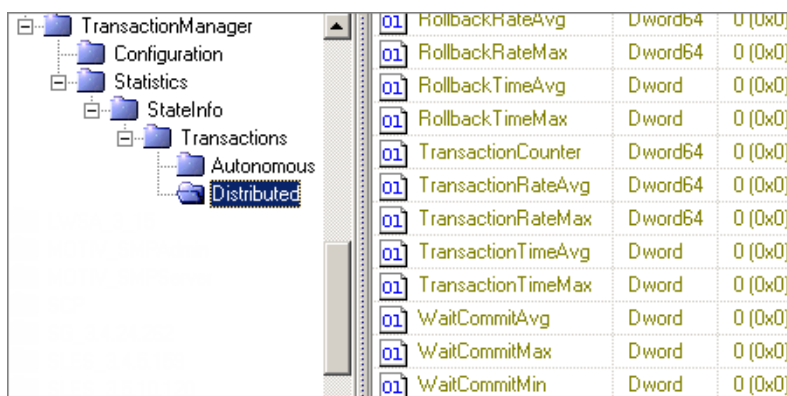
**Рис. 135. Статистика по количеству сигналов в очереди на обработку**

Статистика по распределенным и автономным транзакциям отображается в группах `LWSA_X.Y/TransactionManager/Statistics/StateInfo/Transactions/Distributed` и `LWSA_X.Y/TransactionManager/Statistics/StateInfo/Transactions/Autonomous` соответственно:

- *ActiveTransactionAvg* — среднее количество активных транзакций;
- *ActiveTransactionMax* — максимальное количество активных транзакций;
- *ActiveTransactionMin* — минимальное количество активных транзакций;
- *CommitCounter* — общее количество фиксаций транзакций;
- *CommitRateAvg* — среднее количество фиксаций транзакций в секунду;
- *CommitRateMax* — максимальное количество фиксаций транзакций в секунду;
- *CommitTimeAvg* — среднее время фиксации транзакции в миллисекундах;
- *CommitTimeMax* — максимальное время фиксации транзакции в миллисекундах;
- *InCommitAvg* — среднее количество транзакции, которые выполняют фиксацию;
- *InCommitMax* — максимальное количество транзакции, которые выполняют фиксацию;
- *InCommitMin* — минимальное количество транзакции, которые выполняют фиксацию;
- *InRollbackAvg* — среднее количество транзакции, которые выполняют отмену;
- *InRollbackMax* — максимальное количество транзакции, которые выполняют отмену;
- *InRollbackMin* — минимальное количество транзакции, которые выполняют отмену;
- *InPrepareAvg* — среднее количество транзакции, которые выполняют подготовку;
- *InPrepareMax* — максимальное количество транзакции, которые выполняют подготовку;
- *InPrepareMin* — минимальное количество транзакции, которые выполняют подготовку;

- *PrepareCounter* — общее количество подготовок транзакций;
- *PrepareRateAvg* — среднее количество подготовок транзакций в секунду;
- *PrepareRateMax* — максимальное количество подготовок транзакций в секунду;
- *PrepareTimeAvg* — среднее время подготовки транзакции в миллисекундах;
- *PrepareTimeMax* — максимальное время подготовки транзакции в миллисекундах;
- *RollbackCounter* — общее количество отмен транзакций;
- *RollbackRateAvg* — среднее количество отмен транзакций в секунду;
- *RollbackRateMax* — максимальное количество отмен транзакций в секунду;
- *RollbackTimeAvg* — среднее время отмены транзакции в миллисекундах;
- *RollbackTimeMax* — максимальное время отмены транзакции в миллисекундах;
- *TransactionCounter* — общее количество созданных транзакций;
- *TransactionRateAvg* — среднее количество транзакций в секунду;
- *TransactionRateMax* — максимальное количество транзакций в секунду;
- *TransactionTimeAvg* — среднее время выполнения транзакции в миллисекундах;
- *TransactionTimeMax* — максимальное время выполнения транзакции в миллисекундах;
- *WaitPrepareAvg* — среднее количество транзакций, которые ожидают подготовку;
- *WaitPrepareMax* — максимальное количество транзакций, которые ожидают подготовку;
- *WaitPrepareMin* — максимальное количество транзакций, которые ожидают подготовку;
- *WaitCommitAvg* — среднее количество транзакций, которые ожидают фиксацию;
- *WaitCommitMax* — максимальное количество транзакций, которые ожидают фиксацию;
- *WaitCommitMin* — максимальное количество транзакций, которые ожидают фиксацию;
- *WaitRollbackAvg* — среднее количество транзакций, которые ожидают отмену;
- *WaitRollbackMax* — максимальное количество транзакций, которые ожидают отмену;
- *WaitRollbackMin* — максимальное количество транзакций, которые ожидают отмену;
- *WaitCommitTimeAvg* — среднее время ожидания фиксации транзакции в миллисекундах;
- *WaitCommitTimeMax* — максимальное время ожидания фиксации транзакции в миллисекундах;
- *WaitRollbackTimeAvg* — среднее время ожидания отмены транзакции в миллисекундах;
- *WaitRollbackTimeMax* — максимальное время ожидания отмены транзакции в миллисекундах;
- *WaitPrepareTimeAvg* — среднее время ожидания подготовки транзакции в миллисекундах;
- *WaitPrepareTimeMax* — максимальное время ожидания подготовки транзакции в миллисекундах.

**i Примечание.** Чем больше значения параметров *Wait\*\*\**, тем больше транзакций ожидает обработки. Это означает, что не хватает рабочих потоков или подключений к БД.



01	RollbackRateAvg	Dword64	0 (0x0)
01	RollbackRateMax	Dword64	0 (0x0)
01	RollbackTimeAvg	Dword	0 (0x0)
01	RollbackTimeMax	Dword	0 (0x0)
01	TransactionCounter	Dword64	0 (0x0)
01	TransactionRateAvg	Dword64	0 (0x0)
01	TransactionRateMax	Dword64	0 (0x0)
01	TransactionTimeAvg	Dword	0 (0x0)
01	TransactionTimeMax	Dword	0 (0x0)
01	WaitCommitAvg	Dword	0 (0x0)
01	WaitCommitMax	Dword	0 (0x0)
01	WaitCommitMin	Dword	0 (0x0)

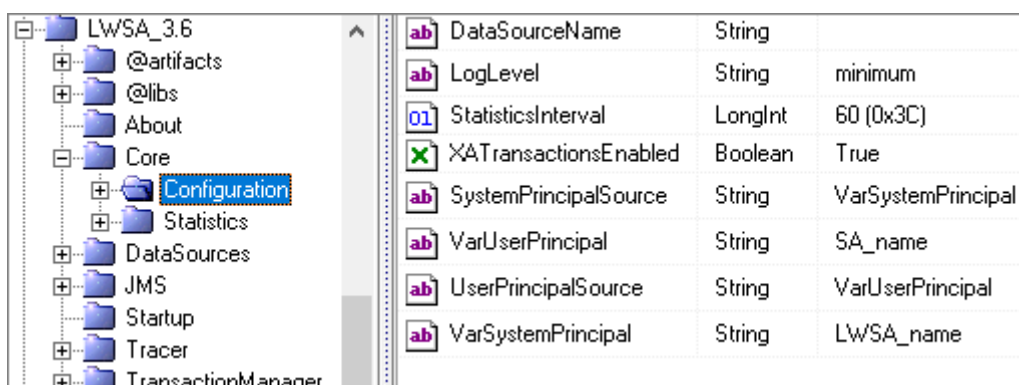
**Рис. 136. Статистика по распределенным и автономным транзакциям**

## 10.6. Настройка UserPrincipal и SystemPrincipal

Задать *UserPrincipal* и *SystemPrincipal* можно в группе *LWSAContainer/Core/Configuration*.

Если разработчик SA не задал программно *UserPrincipal* и *SystemPrincipal*, LW SA Container инициализирует их согласно настройкам MIB-группы *LWSAContainer/Core/Configuration*:

- *SystemPrincipalSource*. Источник значения для *SystemPrincipal*. Если параметр отсутствует, в качестве *SystemPrincipal* используется MIB-имя LW SA Container, который иницирует клиентский запрос.
- *UserPrincipalSource*. Источник значения для *UserPrincipal*. Если параметр отсутствует, в качестве *UserPrincipal* используется MIB-имя SA, который иницирует клиентский запрос.
- *VarSystemPrincipal*. Текстовое значение *SystemPrincipal*, если параметр *SystemPrincipalSource*='VarSystemPrincipal'.
- *VarUserPrincipal*. Текстовое значение *UserPrincipal*, если параметр *UserPrincipalSource*='VarUserPrincipal'.



ab	DataSourceName	String	
ab	LogLevel	String	minimum
01	StatisticsInterval	LongInt	60 (0x3C)
X	XATransactionsEnabled	Boolean	True
ab	SystemPrincipalSource	String	VarSystemPrincipal
ab	VarUserPrincipal	String	SA_name
ab	UserPrincipalSource	String	VarUserPrincipal
ab	VarSystemPrincipal	String	LWSA_name

**Рис. 137. Настройка UserPrincipal и SystemPrincipal**

Эти параметры используются в следующих случаях:

- SA выполняет клиентский вызов и иницирует цепочку вызовов, например, по расписанию;

- на серверный интерфейс SA поступает запрос без ATLAS-контейнера, например, из SoapUI.

## 10.7. Логирование

Логирование в LW SA Container многоуровневое, в соответствии со структурой. Каждая библиотека (ATLAS, MessageBus), функциональная часть или артефакт записывают логи в рамках приложения. Чтобы их различить, система использует имя класса логирования и имя экземпляра. Лог содержит текст сообщения и дополнительные параметры. К дополнительным параметрам относятся Bussines Application и SessionId, которые нужны системе Operations&Management.

### 10.7.1. Параметр Bussines Application для логирования

Алгоритм получения параметра Bussines Application:

- Bussines Application может быть получен по шине во входящем сообщении в ATLAS контейнере.
- Если Bussines Application не получен во входящем сообщении, то LW SA Container использует бизнес-приложение из файла `manifest.mf` каталога `META-INF SA`. Бизнес-приложение указывается в заголовке `Business-Application`.

### 10.7.2. Параметр SessionId для логирования

SessionId существует в рамках обработки вызова. При логировании запуска, остановки приложения и изменении параметров конфигурации он отсутствует. В сообщениях при обработке вызова он может отсутствовать, если на этот момент SessionId еще не определен, например, при получении HTTP-сообщения от внешней системы.

Алгоритм получения SessionId:

- SessionId может быть получен по шине во входящем сообщении в ATLAS контейнере.
- Если SessionId не получен во входящем сообщении, то SessionId формируется в процессе его обработки.

### 10.7.3. Запись логов SA

SA может записывать логи и алармы, которые объявлены в ATLAS-описании. Система с помощью LDC формирует файлы Java-классов (заглушки).

Указанные классы используются в коде SA для записи логов и алармов в агенте. Для этого используется базовый ATLAS-контейнер приложения. Кроме базового существует сессионный контейнер, который создан на основе десериализованного ATLAS-контейнера, поступившего из сети в заголовке SOAP-пакета. В сетевом контейнере хранится информация об идентификаторе сессии, который создан при инициализации сессии на шине. Идентификатор сессии присутствует во всей цепочке вызовов внутри сервиса. Логи и алармы сессионного контейнера ставят в соответствие логи и алармы приложения с цепочкой, которая определена идентификатором сессии.

LW SA Container использует механизм подмены объекта трейса на этапе логирования. Это позволяет использовать информацию из сессионного контейнера при записи логов из несессионных объектов. Особенности работы механизма:

- LW SA Container обрабатывает входящее сообщение, вызывает операции SA и записывает лог в одном потоке. При многопоточной обработке запроса в SA система создает объекты трейса и аларма из сессионного контейнера. Если разработчик SA обрабатывает запрос в потоках, которые отличны от рабочих потоков шины, он должен создать сессионный контейнер из ATLAS-контейнера, поступившего из сети. С помощью созданного контейнера обеспечивается запись логов.
- LW SA Container перед вызовом операции в SA сохраняет в переменной ThreadLocal сессионный контейнер, полученный в запросе или созданный, если его не было.
- ATLAS при записи сообщения от SA проверяет наличие сессионного контейнера. Если он есть, система ATLAS создает сессионные объекты для трейса и аларма и выполняет запись с их помощью. Процесс оптимизации не позволяет использовать объекты, которые были ранее созданы в этом потоке для указанной операции.

Вы можете настроить трассировку сообщений:

- для всех portType. MIB-группа LWSAContainer\_vX.Y/Tracer/Configuration.
- для portType. MIB-группа LWSAContainer\_vX.Y/Tracer/Configuration/PortTypes/<имя portType>.
- для операции portType. MIB-группа LWSAContainer\_vX.Y/Tracer/Configuration/PortTypes/<имя portType>/Operations/<имя операции>.
- для фильтра по бизнес-параметрам операции portType. MIB-группа LWSAContainer\_vX.Y/Tracer/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>.

В каждой группе задайте значения следующих MIB-параметров:

- *Enabled* — флаг включения трассировки для соответствующего объекта:
  - True — трассировка включена;
  - False — трассировка выключена. Значение по умолчанию.
- *Sampling* — порядковый номер сообщений для трассировки. Например: при значении '5' выполняется трассировка каждого пятого сообщения. Значение используется при поставленном флаге *Enabled*.
- *BusinessApplication* — имя бизнес-приложения для соответствующего объекта.
- *Type* — включение трассировки для типов вызова. Параметр можно задать только в MIB-группе /Tracer/Configuration/.



**Рис. 138. Настройки трассировки сообщений**

Для всех portType вы можете включить отображение сообщений уровня Debug. Для этого поставьте флаг *DebugTracesEnabled* в группе SG\_X.Y/Tracer/Configuration. В сообщениях уровня Debug отображается применение трассировки для конкретных сообщений.

Для фильтра по бизнес-параметрам операции portType задайте условие срабатывания фильтра. Задайте XPath-выражение в MIB-параметре *Condition* группы SG\_X.Y/

Tracer/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>.

Наивысший приоритет имеют настройки для фильтра, затем для операции, для portType и наименьший приоритет у настроек для всех portType.

## 10.8. Статистические параметры JMS-соединения

*Система LW SA Container собирает статистические данные для JMS-соединения.*

Система формирует статистику:

- производительности;
- качества.

### **Статистика производительности**

Система собирает статистику в MIB-группе: LWSA/JMS/Statistics/Performance.

Группы статистических переменных:

- *ProcessedMessages*. Количество обработанных нотификаций: подтвержденных и отклоненных.
- *ProcessedFastMessages*. Количество обработанных «быстрых» нотификаций.
- *QueryRate*. Количество отправленных запросов.
- *ReceivedMessages*. Количество полученных нотификаций.
- *ReceivedFastMessages*. Количество полученных «быстрых» нотификаций.
- *ReceivedRedeliveredMessages*. Количество полученных повторных нотификаций.

01	ProcessedMessages	Dword64	0 (0x0)
01	ProcessedMessagesAvg	Dword64	0 (0x0)
01	ProcessedMessagesAvgSpeed	Dword64	0 (0x0)
01	ProcessedMessagesCounter	Dword64	10 (0xA)
01	ProcessedMessagesMax	Dword64	0 (0x0)
01	ProcessedMessagesMin	Dword64	0 (0x0)
01	ProcessedMessagesSum	Dword64	0 (0x0)
01	QueryRate	Dword64	0 (0x0)
01	QueryRateAvg	Double	0
01	QueryRateMax	Double	0
01	ReceivedMessages	Dword64	0 (0x0)
01	ReceivedMessagesAvg	Dword64	0 (0x0)
01	ReceivedMessagesAvgSpeed	Dword64	0 (0x0)
01	ReceivedMessagesCounter	Dword64	10 (0xA)
01	ReceivedMessagesMax	Dword64	0 (0x0)
01	ReceivedMessagesMin	Dword64	0 (0x0)
01	ReceivedMessagesSum	Dword64	0 (0x0)
01	ReceivedRedeliveredMessages	Dword64	0 (0x0)
01	ReceivedRedeliveredMessagesAvg	Dword64	0 (0x0)
01	ReceivedRedeliveredMessagesAvgSpeed	Dword64	0 (0x0)
01	ReceivedRedeliveredMessagesCounter	Dword64	5 (0x5)
01	ReceivedRedeliveredMessagesMax	Dword64	0 (0x0)
01	ReceivedRedeliveredMessagesMin	Dword64	0 (0x0)
01	ReceivedRedeliveredMessagesSum	Dword64	0 (0x0)

**Рис. 139. Статистика производительности JMS-соединения**

### Статистика качества

Система собирает статистику в MIB-группе: LWSA/JMS/Statistics/Quality

Группы статистических переменных:

- *CommittedMessages*. Количество подтвержденных нотификаций.
- *EnqueueCommitTimeMs*. Период, который начинается с добавления нотификации в очередь БД и заканчивается подтверждением в JMS-клиенте. Период указывается в миллисекундах.
- *ReceiveResponseTimeMs*. Период, который начинается с извлечения нотификации из очереди БД и заканчивается завершением обработки: подтверждением или отклонением в JMS-клиенте. Период указывается в миллисекундах.
- *RolledBackMessages*. Количество отклоненных нотификаций.

Переменная *LastFailMessages*. Последние N ошибок при обработке JMS-сообщений. По умолчанию N=3.

01	CommittedMessages	Dword64	0 (0x0)
01	CommittedMessagesAvg	Dword64	0 (0x0)
01	CommittedMessagesAvgSpeed	Dword64	0 (0x0)
01	CommittedMessagesCounter	Dword64	3 (0x3)
01	CommittedMessagesMax	Dword64	0 (0x0)
01	CommittedMessagesMin	Dword64	0 (0x0)
01	CommittedMessagesSum	Dword64	0 (0x0)
01	EnqueueCommitTimeMs	Dword64	0 (0x0)
01	EnqueueCommitTimeMsAvg	Dword64	0 (0x0)
01	EnqueueCommitTimeMsAvgSpeed	Dword64	0 (0x0)
01	EnqueueCommitTimeMsCounter	Dword64	12286 (0x2FFE)
01	EnqueueCommitTimeMsMax	Dword64	0 (0x0)
01	EnqueueCommitTimeMsMin	Dword64	0 (0x0)
01	EnqueueCommitTimeMsSum	Dword64	0 (0x0)
01	LastFailMessages	LongInt	0 (0x0)
01	ReceiveResponseTimeMs	Dword64	0 (0x0)
01	ReceiveResponseTimeMsAvg	Dword64	0 (0x0)
01	ReceiveResponseTimeMsAvgSpeed	Dword64	0 (0x0)
01	ReceiveResponseTimeMsCounter	Dword64	21989 (0x55E5)
01	ReceiveResponseTimeMsMax	Dword64	0 (0x0)
01	ReceiveResponseTimeMsMin	Dword64	0 (0x0)
01	ReceiveResponseTimeMsSum	Dword64	0 (0x0)
01	RolledBackMessages	Dword64	0 (0x0)
01	RolledBackMessagesAvg	Dword64	0 (0x0)
01	RolledBackMessagesAvgSpeed	Dword64	0 (0x0)
01	RolledBackMessagesCounter	Dword64	7 (0x7)
01	RolledBackMessagesMax	Dword64	0 (0x0)
01	RolledBackMessagesMin	Dword64	0 (0x0)
01	RolledBackMessagesSum	Dword64	0 (0x0)

**Рис. 140. Статистика качества JMS-соединения**

## 10.9. Статистические параметры обработки транзакций

Система *LW SA Container* собирает статистические данные при обработке транзакций.

Чтобы задать количество потоков для обработки транзакций, используйте MIB-параметр *WorkingThreads* группы *CMP\_LWSAContainer\_vX.Y/TransactionManager/Configuration*.

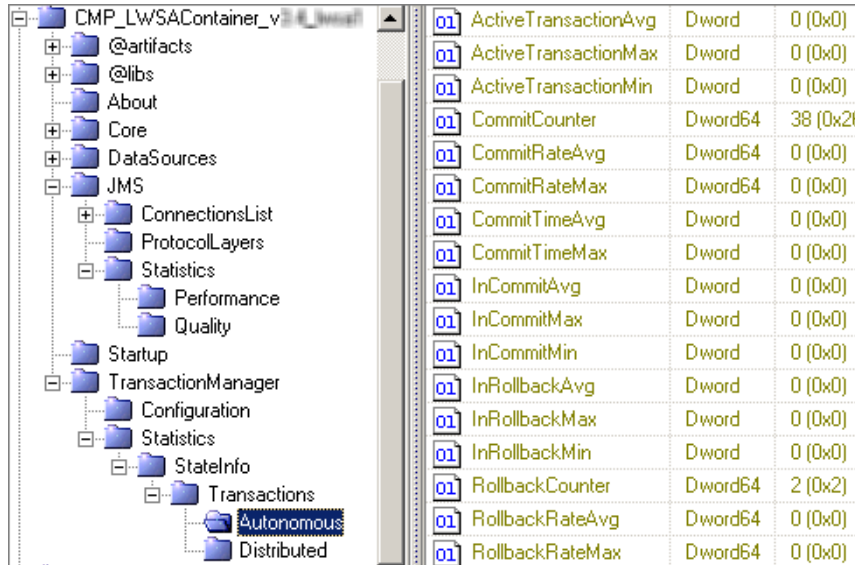
Переменные группы *CMP\_LWSAContainer\_vX.Y/TransactionManager/Statistics/StateInfo/Transactions/<тип транзакции>*:

- *ActiveTransactionAvg*. Среднее количество активных транзакций.
- *ActiveTransactionMax*. Максимальное количество активных транзакций.
- *ActiveTransactionMin*. Минимальное количество активных транзакций.
- *CommitCounter*. Общее количество регистраций транзакций.
- *CommitRateAvg*. Среднее количество регистраций в секунду.
- *CommitRateMax*. Максимальное количество регистраций.
- *CommitTimeAvg*. Среднее время регистрации транзакции, в миллисекундах.
- *CommitTimeMax*. Максимальное время регистрации транзакции, в миллисекундах.



- *InCommitAvg*. Среднее количество транзакций, которые выполняют регистрацию.
- *InCommitMax*. Максимальное количество транзакций, которые выполняют регистрацию.
- *InCommitMin*. Минимальное количество транзакций, которые выполняют регистрацию.
- *InPrepareAvg*. Среднее количество транзакций, которые выполняют подготовку.
- *InPrepareMax*. Максимальное количество транзакций, которые выполняют подготовку.
- *InPrepareMin*. Минимальное количество транзакций, которые выполняют подготовку.
- *InRollbackAvg*. Среднее количество транзакций, которые выполняют откат.
- *InRollbackMax*. Максимальное количество транзакций, которые выполняют откат.
- *InRollbackMin*. Минимальное количество транзакций, которые выполняют откат.
- *PrepareRateAvg*. Среднее количество подготовок в секунду.
- *PrepareRateMax*. Максимальное количество подготовок транзакций.
- *PrepareTimeAvg*. Среднее время подготовки транзакции, в миллисекундах.
- *PrepareTimeMax*. Максимальное время подготовки транзакции, в миллисекундах.
- *PrepareCounter*. Общее количество подготовок транзакций.
- *RollbackCounter*. Общее количество откатов транзакций.
- *RollbackRateAvg*. Среднее количество откатов в секунду.
- *RollbackRateMax*. Максимальное количество откатов
- *RollbackTimeAvg*. Среднее время отката транзакции, в миллисекундах.
- *RollbackTimeMax*. Максимальное время отката транзакции, в миллисекундах.
- *TransactionCounter*. Общее количество новых транзакций.
- *TransactionRateAvg*. Среднее количество транзакций, в секунду.
- *TransactionRateMax*. Максимальное количество транзакций.
- *TransactionTimeAvg*. Среднее время выполнения транзакции, в миллисекундах.
- *TransactionTimeMax*. Максимальное время выполнения транзакции, в миллисекундах.
- *WaitCommitAvg*. Среднее количество транзакций, ожидающих регистрации.
- *WaitCommitMax*. Максимальное количество транзакций, ожидающих регистрации.
- *WaitCommitMin*. Минимальное количество транзакций, ожидающих регистрации.
- *WaitCommitTimeAvg*. Среднее время ожидания регистрации транзакции, в миллисекундах.
- *WaitCommitTimeMax*. Максимальное время ожидания регистрации транзакции, в миллисекундах.
- *WaitPrepareAvg*. Среднее количество транзакций, ожидающих подготовки.
- *WaitPrepareMax*. Максимальное количество транзакций, ожидающих подготовки.
- *WaitPrepareMin*. Минимальное количество транзакций, ожидающих подготовки.
- *WaitPrepareTimeAvg*. Среднее время ожидания подготовки транзакции, в миллисекундах.
- *WaitPrepareTimeMax*. Максимальное время ожидания подготовки транзакции, в миллисекундах.
- *WaitRollbackAvg*. Среднее количество транзакций, ожидающих отката.
- *WaitRollbackMax*. Максимальное количество транзакций, ожидающих отката.

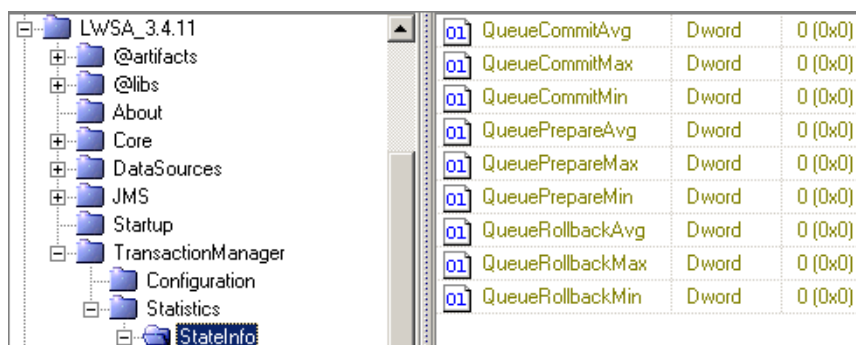
- *WaitRollbackMin*. Минимальное количество транзакций, ожидающих отката.
- *WaitRollbackTimeAvg*. Среднее время ожидания отката транзакции, в миллисекундах.
- *WaitRollbackTimeMax*. Максимальное время ожидания отката транзакции, в миллисекундах.



**Рис. 141. Статистика по обработке транзакций**

Переменные группы CMP\_LWSAContainer\_vX.Y/TransactionManager/Statistics/StateInfo:

- *QueueCommitAvg*. Среднее количество сигналов на регистрацию в очереди.
- *QueueCommitMax*. Максимальное количество сигналов на регистрацию в очереди.
- *QueueCommitMin*. Минимальное количество сигналов на регистрацию в очереди.
- *QueuePrepareAvg*. Среднее количество сигналов на подготовку в очереди.
- *QueuePrepareMax*. Максимальное количество сигналов на подготовку в очереди.
- *QueuePrepareMin*. Минимальное количество сигналов на подготовку в очереди.
- *QueueRollbackAvg*. Среднее количество сигналов на отмену в очереди.
- *QueueRollbackMax*. Максимальное количество сигналов на отмену в очереди.
- *QueueRollbackMin*. Минимальное количество сигналов на отмену в очереди.



The screenshot shows a file explorer window for the directory 'LWSA\_3.4.11'. The left pane displays a tree view with folders: @artifacts, @libs, About, Core, DataSources, JMS, Startup, TransactionManager, Configuration, and Statistics. The 'Statistics' folder is selected and expanded, showing a table of queue statistics.

01	QueueCommitAvg	Dword	0 (0x0)
01	QueueCommitMax	Dword	0 (0x0)
01	QueueCommitMin	Dword	0 (0x0)
01	QueuePrepareAvg	Dword	0 (0x0)
01	QueuePrepareMax	Dword	0 (0x0)
01	QueuePrepareMin	Dword	0 (0x0)
01	QueueRollbackAvg	Dword	0 (0x0)
01	QueueRollbackMax	Dword	0 (0x0)
01	QueueRollbackMin	Dword	0 (0x0)

**Рис. 142. Статистика очередей по обработке транзакций**

# 11. Сборщик мусора Garbage Collector G1

*Oracle Garbage Collector (GC) G1 — сборщик мусора, который разбивает память на регионы и закрепляет за регионом область памяти, к которой он относится в текущий момент. Размер региона по умолчанию: 1 Мб.*

Garbage Collector G1 может выполнять сбор мусора следующими способами:

- сборка молодого поколения;
- смешанная сборка молодого и части старого поколения;
- полная сборка всего старого поколения.

Первые две сборки выполняются часто. Полная сборка выполняется редко — в тех случаях, если закончилась память или из-за фрагментации не хватает ресурсов для выделения памяти новому объекту.

Garbage Collector G1 автоматически определяет начальное пороговое значение процента занятости heap-области (IHOP) на основании раннего поведения приложения и параметров запуска. По результату система выполняет сборку молодого поколения или смешанную сборку.

При очистке мусора выполняются следующие фазы:

- фаза параллельной маркировки:
  - система вычисляет информацию о жизнеспособности памяти одновременно во время работы приложения.
  - по полученной информации система определяет, какие регионы лучше всего восстановить во время эвакуационной паузы.
  - фаза развертывания отсутствует.
- фаза финальной маркировки:
  - использует алгоритм моментального снимка в начале (SATB);
  - становятся доступны для использования полностью пустые регионы.
- фаза копирования и очистки:
  - одновременно восстанавливаются и копируются молодое и старое поколение;
  - регионы старого поколения выбираются на основании их жизнеспособности.

Конфигурационный файл GC содержит следующие параметры для настройки:

Название	Тип	Описание
APP_NAME	String	Имя приложения для отображения в дереве MIB. По умолчанию отображает имя, сформированное по правилам ATOMS. Например, для SLES 3.6 — «CMP_SLES_v3.6».
HEAP_MEM	String	Размер heap-памяти. Резервируется весь объем памяти, чтобы избежать задержек при работе приложения. Это касается только виртуальной памяти, которая будет отображаться в физической по мере выполнения приложения. Это может занять

Название	Тип	Описание
		некоторое время. Значение по умолчанию: '1G'.
USE_ALL_MEM_ON_START	String	Если значение — 'on', то JVM при запуске занимает всю физическую память, указанную в параметре HEAP_MEM. Данная настройка уменьшает степень зависимости от системы, время отклика и процессорное время при работе GC. При этом следует учесть, что мониторинг использования физической памяти, занятой процессом, становится неэффективным, так как значение HEAP_MEM всегда будет превышено: размер самой HEAP_MEM + DIRECT_MEM + инфраструктура JVM — потоки, классы, и другие. Необходимо выполнять мониторинг при помощи JMX-подключений к самой JVM, например — использовать приложение JvmMonitor. Значение по умолчанию: 'off'.
DIRECT_MEM	String	Размер direct (off-heap)-памяти, которая необходима для работы с сетью. Этот размер необходим, чтобы избежать возникновения ошибки 'OutOfMemory direct memory'. Значение по умолчанию: '512M'.
MAX_GC_PAUSE	Integer	Максимальная пауза, в миллисекундах. GC G1 избегает превышения этого значения. Значение по умолчанию: '50'. Рекомендуемое значение для сервисов, чувствительных к превышению: '10'.
RESERVE_PERCENT	Integer	Процент зарезервированной памяти. Если объем свободной памяти — меньше, чем задано в данной переменной, то G1 пересматривает начальный процент порога занятости кучи (IHOP) для запуска GC, и, как правило, стремится сразу запустить сбор мусора. Процент можно задать от '0' до '50'. Значение по умолчанию — '25'.
LOG_GC	Boolean	Признак логирования работы GC G1. Значение по умолчанию: 'off'.
NEW_SIZE		Размер памяти, которая может быть использована под молодое поколение. <b>Не рекомендуется</b> задавать данный параметр, если вы не уверены в правильности своих действий. Задание данного параметра <b>отключает эвристику G1</b> , и он перестает вычислять IHOP. При этом <b>не гарантируется</b> установка максимального размера паузы в миллисекундах, заданной в переменной MAX_GC_PAUSE. Отсутствие гарантий обусловлено двумя факторами: <ol style="list-style-type: none"> <li>1. Отключение эвристики для расчета IHOP.</li> <li>2. При переполнении молодого поколения данные будут перенесены в старое</li> </ol>

Название	Тип	Описание
		поколение, и скорее всего будет вызвана полная сборка мусора.
DEBUG_ON	Boolean	Включает возможность подключения к приложению отладчиком. Допустимые значения: <ul style="list-style-type: none"> <li>True;</li> <li>False.</li> </ul> Значение по умолчанию — False.
DEBUG_PORT	Integer	Порт для подключения отладчика. Диапазон допустимых значений: от '1' до '65536'. Значение по умолчанию — '5012'.
DEBUG_SUSPEND	String	Допустимые значения: 'y' или 'n'. Включает режим suspend для отладки. Режим означает, что JVM запустится и начнет выполнять приложение только после того, как будет подключен отладчик. Значение по умолчанию — 'n'.
ATLAS_DESC_DISABLE	String	Любая строка в качестве значения выключает использование ATLAS-описания.
ATLAS_BACKUP_REMOVE	Boolean	Флаг удаления резервной копии при миграции версии компонента: <ul style="list-style-type: none"> <li>True — удалять резервную копию;</li> <li>False — не удалять резервную копию (выключено).</li> </ul> Значение по умолчанию — False.
OPTIONS	String	Дополнительные опции для JVM.

**⚠️ Внимание!** Если параметру `MAX_GC_PAUSE` присвоено небольшое значение, память используется в большем объеме. В результате GC G1 будет выполнять дефрагментацию постепенно, дожидаясь, пока весь регион станет неиспользуемым. Это позволяет более эффективно использовать память и ресурсы процессора.

Для корректного мониторинга нехватки памяти нужно выполнять мониторинг старой памяти и предотвращать частое превышение 100 % — `RESERVE_PERCENT`. Значение `RESERVE_PERCENT` по умолчанию — 25 %. Это позволяет учитывать поведение CMS GC, который обычно при нормальной работе не превышает использования 75 % доступной памяти.

Пример настройки мониторинга: если при нормальном использовании старая память занимает 300 Мб, и значение параметра `RESERVE_PERCENT=25`, то для алармов можно задать 700 Мб с целью утилизации старой памяти и утилизации 90 % всего объема памяти.

## 12. Система Service Gateway

Система *Service Gateway* служит для подключения внешних приложений и предоставления доступа к интерфейсам Platform v3.

### 12.1. Установка и запуск

Для установки системы Service Gateway используйте файл `startup.sh`.

Порядок установки системы Service Gateway:

1. Переключитесь на пользователя `bercut`. Все последующие действия выполняйте от имени этого пользователя.
2. Создайте каталог установки системы Service Gateway, именуемый дальше `$ServiceGateway`.  
`mkdir ServiceGateway`
3. Скопируйте каталог дистрибутива в созданный каталог.
4. Распакуйте архив установочного дистрибутива.
5. Установите права на запуск скрипт-файлов в каталоге `bin` для всех пользователей.  
`cd /opt/ServiceGateway/bin`

```
chmod +x *.sh
```

6. Выполните команду:

```
./startup.sh
```

Система автоматически создаст дерево в приложении ATLAS MIB Explorer.

**i Примечание.** При последующих запусках и остановках системы Service Gateway используйте на выбор:

- `SSM`.
- команду `./startup.sh` или `./shutdown.sh`.

7. Настройте *порты приложений или взаимодействие* между компонентами.
8. Проверьте трейс в приложении ATLAS MIB Explorer на отсутствие ошибок от системы Service Gateway.  
Проверьте в MIB, что Service Gateway доступен. Переменная `Status` группы `CMP_SG_vX.Y/Core/Statistics` должна иметь значение `available`.

Чтобы задать специальное имя для экземпляра Service Gateway в скрипт-файле `options.sh`, расположенном в каталоге `bin`, укажите MIB-имя в строке:

```
OPTIONS=${OPTIONS}" -Datlas.applicationName=<имя MIB-группы устанавливаемого экземпляра>"
```

Чтобы удалить созданный резервный файл по окончании *миграции*, в скрипт-файле `options.sh`, расположенном в каталоге `bin`, поставьте флаг `-Datlas.removeCreatedBackup`:

```
OPTIONS=${OPTIONS}" -Datlas.removeCreatedBackup=true"
```

Для остановки системы Service Gateway выполните команду:

```
./shutdown.sh
```

## 12.2. Настройка Service Gateway

Настройте параметры аутентификации и авторизации групп доступа, внешних систем или сервисов, пользователей. Для групп доступа определите правила доступа для `portType`, операции и условия.

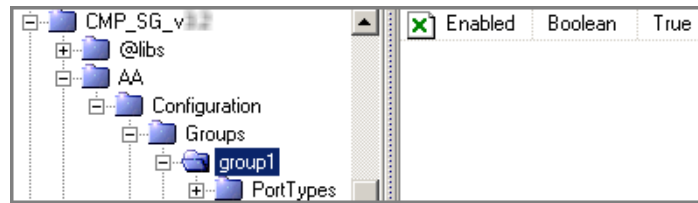
Порядок настройки:

1. Задайте параметры поставщиков информации для аутентификации и авторизации. Для этого используйте переменные MIB-группы `CMP_SG_vX.Y/AA/Configuration`:
  - **GroupsProvider.** Тип провайдера для получения информации о группах авторизации:
    - `mib` — параметры групп настраиваются в MIB или с помощью [MIB](#).
    - `srgw` — параметры групп настраиваются [в профиле](#) Service Gateway, определенному в SA SP.
  - **UsersProvider.** Тип провайдера для получения информации о пользователях для аутентификации и авторизации:
    - `mib` — параметры пользователей настраиваются в MIB.
    - `srgw` — параметры пользователей настраиваются [в профиле](#) Service Gateway, определенному в SA SP.
  - **SystemsProvider.** Тип провайдера для получения информации о системах для аутентификации и авторизации:
    - `mib` — параметры систем настраиваются в MIB.
    - `srgw` — параметры систем настраиваются [в профиле](#) Service Gateway, определенному в SA SP.
    - `sp` — параметры систем настраиваются в [запросе](#) на создание сервиса в теге **ServiceAuth**.
  - **ProviderSystemName.** Имя системы в SA SP, хранящей профиль Service Gateway.

**i Примечание.** При работе с профилем в SA SP [подключите](#) стандартные XSD-файлы системы SPM, а также файл `ServiceGateway.xsd`. XSD-файлы расположены в папке `sp-xsd` каталога дистрибутива ServiceGateway. Для создания сервиса ServiceGateway выполните операцию `createService` с параметрами, заданными в XSD-файле. Параметры задаются в теге **parametersSG**.

2. Задайте параметры групп доступа для аутентификации в Service Gateway. Для этого создайте вложенную в `CMP_SG_vX.Y/AA/Configuration/Groups` группу для каждой группы доступа. Для изменения статуса группы используйте переменную `Enabled` группы `CMP_SG_vX.Y/AA/Configuration/Groups/<имя группы>`:
  - `True` — группа активна.
  - `False` — группа неактивна.





**Рис. 143. Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>**

Для удаления группы доступа выберите соответствующую MIB-группу и нажмите **Удалить группу**.

3. Настройте права доступа для portType. Для этого создайте вложенную в CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes группу для каждого portType.

Для установки политики доступа для portType используйте переменную *Policy* группы CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>:

- allow — все операции portType доступны для группы, кроме отдельно запрещенных.
- deny — все операции portType недоступны для группы, кроме отдельно разрешенных.



**Рис. 144. Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>**

Для удаления установленных прав доступа для portType выберите соответствующую MIB-группу и нажмите **Удалить группу**.

4. Установите доступ к операциям для portType. Для этого создайте вложенную в CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations группу для каждой операции.

Для установки политики доступа для операции используйте переменную *Policy* группы CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<название операции>:

- allow — операция portType доступна для группы, независимо от прав доступа portType.
- deny — операция portType недоступна для группы, независимо от прав доступа portType.



**Рис. 145. Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<название операции>**

Для удаления операции выберите соответствующую MIB-группу и нажмите **Удалить группу**.

5. Настройте права доступа для операции по условиям. Для этого создайте вложенную в CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<название операции>/Conditional группу для каждого условия.

Для созданной группы задайте MIB-параметры:

- **Condition.** XPath-выражение условия авторизации для операции.
- **Policy.** Политика доступа для условия:
  - allow — операция доступна при выполнении условия, независимо от политики доступа portType и политики доступа операции.
  - deny — операция недоступна при выполнении условия, независимо от политики доступа portType и политики доступа операции.



**Рис. 146. Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<название операции>/Conditional/ <название условия>**

Для удаления условия выберите соответствующую MIB-группу и нажмите **Удалить группу**.


6. Задайте права доступа внешних систем или сервисов, обращающихся к Service Gateway. Для этого создайте вложенную в CMP\_SG\_vX.Y/AA/Configuration/Systems группу для каждой внешней системы.

Для созданной группы задайте MIB-параметры:

- **Enabled** — статус внешней системы или сервиса:
  - True — статус «активен».
  - False — статус «неактивен».
- **Access** — тип доступа для внешней системы или сервиса:
  - trust — аутентификация на Service Gateway доступна всем.
  - digest — пользовательская HTTP digest аутентификация на Service Gateway.

- basic — пользовательская HTTP basic аутентификация на Service Gateway.
- off — используется для внешнего менеджера сессий. Аутентификация не выполняется. Система читает cookie с именем **guid**, проверяет в менеджере сессий наличие созданной активной сессии с данным идентификатором.
- *AllowedIPs* — список IP-адресов, доступ с которых разрешен для внешней системы или сервиса. IP-адреса перечисляются через запятую. Можете указать в значении диапазон адресов или маску IP-адреса. Например: `192.168.1.0-192.168.1.55` или `192.168.1.*`.
- *MaximumActiveSessions* — максимальное количество одновременных пользовательских сессий.
- *SessionManager* — тип менеджера сессий для системы:
  - internal — внутренний менеджер сессий, реализованный в Service Gateway.
  - external — внешний менеджер сессий, реализованный в portType `{http://www.bercut.com/specs/aoi/session-management}session-management-port-type`.
  - off — менеджер сессий отключен. Вызов выполняется с использованием аутентификации. Этот тип менеджера используется, если клиенты системы не обрабатывают cookie 'guid'. Этот тип предпочтителен для Trust-систем.
- *UserAgent* — имя агента, передаваемое в HTTP-заголовке. Параметр необязательный.
- *TrustedGroups* — список доступных групп для системы. Для разделения элементов списка используйте запятую. Заполняется только при значении 'trust' переменной *Access*. Параметр необязательный.
- *MessagesPerSecond* — максимальное количество сообщений, отправляемых всеми пользователями системы за секунду. При достижении этого значения Service Gateway прекращает принимать информацию от всех пользователей системы до окончания текущей секунды. Параметр необязательный.
- *MessagesPerSecondType* — тип ограничения максимального количества сообщений в секунду для системы:
  - slow — система имитирует работу медленного канала.
  - reject — система отправляет HTTP-ошибку '509'.

Параметр необязательный.



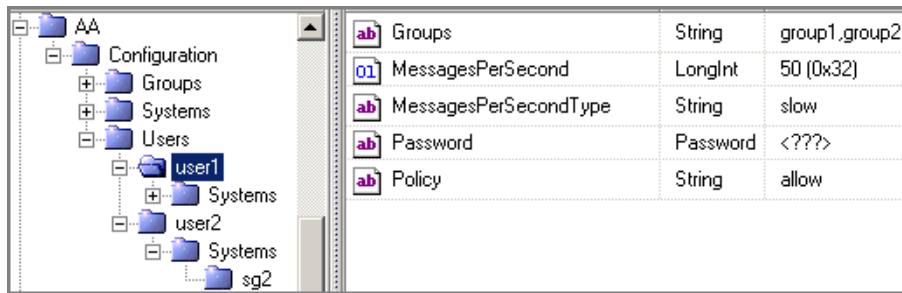
ab	Access	String	digest
ab	AllowedIPs	String	127.0.0.1
X	Enabled	Boolean	True
01	MaximumActiveSessions	LongInt	10 (0x4)
01	MessagesPerSecond	LongInt	50 (0x32)
ab	MessagesPerSecondType	String	slow
ab	SessionManager	String	internal
ab	TrustedGroups	String	group1
ab	UserAgent	String	agent1

**Рис. 147.** Группа CMP\_SG\_vX.Y/AA/Configuration/Systems/<название системы>

7. Задайте права доступа пользователей. Для этого создайте вложенную в CMP\_SG\_vX.Y/AA/Configuration/Users группу для каждого пользователя. Для созданной группы задайте MIB-параметры:

- *Groups* — список групп, доступных пользователю. Группы перечисляются через запятую.
- *Password* — пароль пользователя.
- *Policy* — права доступа пользователя:
  - allow — пользователю доступна аутентификация на Service Gateway.
  - deny — пользователю недоступна аутентификация на Service Gateway.
- *MessagesPerSecond* — максимальное количество сообщений, отправляемых пользователем за секунду. При достижении этого значения Service Gateway прекращает принимать информацию от пользователя до окончания текущей секунды. Параметр необязательный.
- *MessagesPerSecondType* — тип ограничения максимального количества сообщений в секунду для пользователя:
  - slow — система имитирует работу медленного канала.
  - reject — система отправляет HTTP-ошибку '509'.

Параметр необязательный.



**Рис. 148.** Группа CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>

8. Установите доступ к системам для аутентификации пользователя. Для этого создайте вложенную в CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>/Systems группу для каждой системы. Задайте список групп системы, доступных пользователю в переменной *Groups* группы CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>/Systems/<имя системы>. Названия групп перечисляются через запятую.



**Рис. 149.** Группа CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>/Systems/<имя системы>

### 12.2.1. Настройка параметров аутентификации и авторизации в MIB

При значении 'mib' переменных *GroupsProvider*, *SystemsProvider*, *UsersProvider* параметры аутентификации и авторизации.

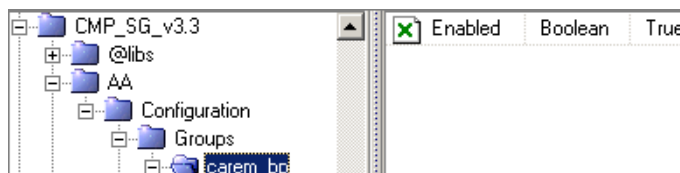
Переменные расположены в группе CMP\_SG\_vX.Y/AA/Configuration.

Порядок настройки:

1. Создайте группы доступа для аутентификации в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups.

1.1. Задайте статус группы в переменной *Enabled* в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>:

- True — группа активна.
- False — группа неактивна.

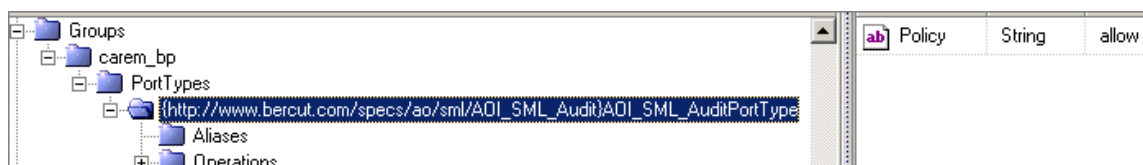


**Рис. 150. Создание групп доступа для аутентификации**

1.2. Создайте список операций для portType в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations.

1.3. Для каждого portType задайте политики доступа в переменной *Policy*:

- allow — все операции portType, кроме отдельно запрещенных, доступны для группы.
- deny — все операции portType, кроме отдельно разрешенных, недоступны для группы.

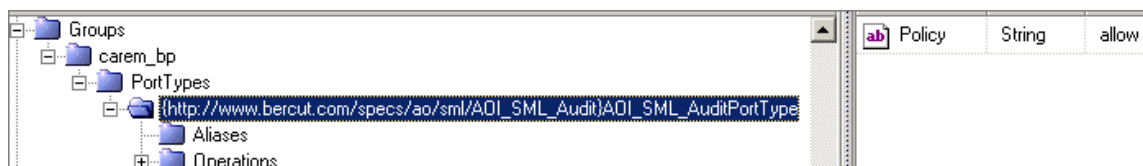


**Рис. 151. Задание политик доступа для portType**

1.4. Создайте список portType в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes.

1.5. Для каждого portType задайте политики доступа в переменной *Policy*:

- allow — все операции portType, кроме отдельно запрещенных, доступны для группы.
- deny — все операции portType, кроме отдельно разрешенных, недоступны для группы.



**Рис. 152. Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>**

1.6. Создайте список операций portType в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations.

1.7. Для каждой операции задайте политики доступа в переменной Policy:

- allow — операция доступна для группы, независимо от политики доступа portType.
- deny — операция недоступна для группы, независимо от политики доступа portType.

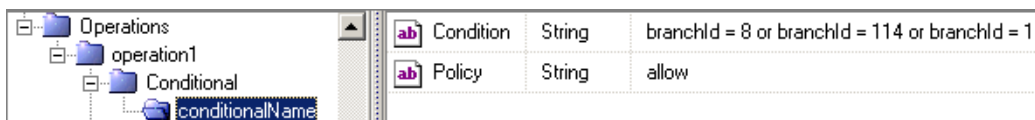


**Рис. 153.** Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<имя операции>

1.8. Создайте список условий для операции в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<имя операции>/Conditional.

1.9. Для каждого условия задайте:

- XPath-выражение, которое задает условие авторизации для операции, в переменной Condition.
- политики доступа в переменной Policy:
  - allow — операция доступна при выполнении условия, независимо от политики доступа portType и политики доступа операции.
  - deny — операция недоступна при выполнении условия, независимо от политики доступа portType и политики доступа операции.



**Рис. 154.**

2. Создайте внешние системы, для которых нужно задать права доступа, в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Systems.

Для каждой системы задайте параметры в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Systems/<имя системы>:

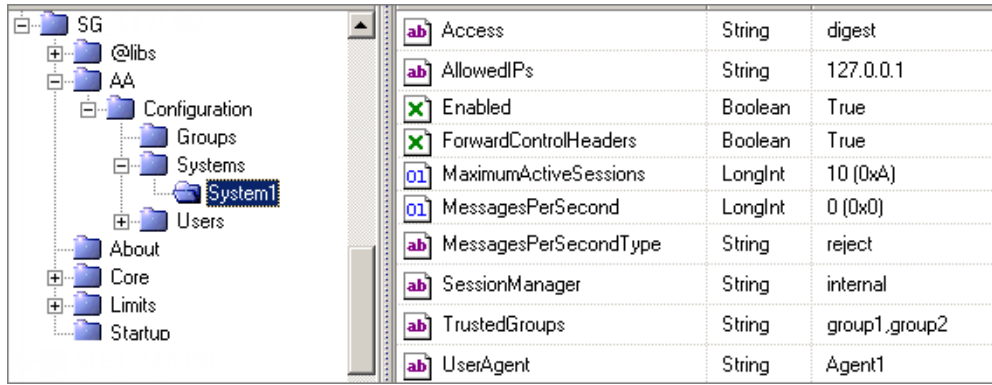
- Статус системы в переменной Enabled:
  - True — система активна.
  - False — система неактивна.
- Тип доступа в переменной Access:
  - trust — аутентификация по имени пользователя не выполняется для всех запросов, отправленных системой. Задайте список доступных для системы групп в переменной TrustedGroups.

- basic — выполняется аутентификация HTTP basic по имени пользователя для всех запросов, отправленных системой.
- digest — аутентификация не выполняется для всех запросов, отправленных системой.
- off — используется для внешнего менеджера сессий. Аутентификация не выполняется. Система читает cookie с именем 'GUID' и проверяет в менеджере сессий наличие созданной активной сессии с указанным идентификатором.
- Список разрешенных IP-адресов в переменной *AllowedIPs*. Элементы списка разделяются запятой. Можно использовать диапазоны и маски, например: `192.168.1.0-192.168.1.55` или `192.168.1.*`.
- Максимальное количество одновременных пользовательских сессий в переменной *MaximumActiveSessions*. Идентификатор пользовательской сессии передается в cookie GUID.
- Тип *менеджера сессий* для данной системы в переменной *SessionManager*:
  - internal — внутренний менеджер сессий, реализованный в Service Gateway.
  - external — внешний менеджер сессий, реализованный в `portType {http://www.bercut.com/specs/aoi/session-management}session-management-port-type`.
  - off — менеджер сессий отключен. Вызов выполняется с использованием аутентификации. Значение используется, если клиенты системы не обрабатывают cookie GUID.
- Список доступных для данной системы групп в переменной *TrustedGroups*. Элементы списка разделяются запятой. Список используется при значении 'trust' переменной *Access*. Параметр необязательный.
- Имя агента, передаваемое в HTTP-заголовке, в переменной *UserAgent*. Параметр необязательный.
- Признак перенаправления управляющих Headers для данной системы в переменной *ForwardControlHeaders*.
- Максимальное количество сообщений, отправляемых всеми пользователями системы за секунду, в переменной *MessagesPerSecond*. При достижении этого значения Service Gateway прекращает принимать информацию от всех пользователей системы до окончания текущей секунды. Параметр необязательный.
- Тип ограничения максимального количества сообщений в секунду для системы в переменной *MessagesPerSecondType*:
  - slow — система использует *медленный* канал.

**i Примечание.** При включении или изменении входящей нагрузки система перерасчитывает значения параметров для ограничения входящей нагрузки с помощью эмуляции медленного канала. При перерасчете возможны неточности, из-за которых система может вычитать из сокета больше запросов, чем указано в ограничении. Запросы, которые система вычла сверх ограничения, будут отклонены с HTTP-кодом '509': *Bandwidth Limit Exceeded*. Система в первую очередь выполняет ограничение по нагрузке, а затем прием всех запросов.

- reject — система отправляет HTTP-ошибку '509'.

Параметр необязательный.



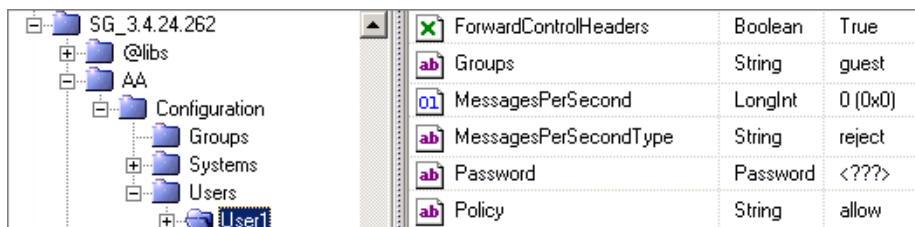
**Рис. 155. Группа CMP\_SG\_vX.Y/AA/Configuration/Systems/<имя системы>**

3. Создайте пользователей, для которых нужно задать права доступа, в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Users.

Для каждого пользователя задайте параметры в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>:

- Пароль пользователя в переменной *Password*.
- Политика доступа в переменной *Policy*:
  - allow — пользователю доступна аутентификация на Service Gateway.
  - deny — пользователю недоступна аутентификация на Service Gateway.
- Список доступных пользователю групп в переменной *Groups*. Элементы списка разделяются запятой.
- Признак перенаправления управляющих Headers для данной системы в переменной *ForwardControlHeaders*.
- Максимальное количество сообщений, отправляемых пользователем за секунду, в переменной *MessagesPerSecond*. При достижении этого значения Service Gateway прекращает принимать информацию от пользователя до окончания текущей секунды. Параметр необязательный.
- Тип ограничения максимального количества сообщений в секунду для пользователя в переменной *MessagesPerSecondType*:
  - slow — система использует *медленный* канал.
  - reject — система отправляет HTTP-ошибку '509'.

Параметр необязательный.



**Рис. 156. Группа CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>**

Создайте системы, которые доступны пользователю в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>/Systems. Для каждой системы задайте список доступных пользователю групп в системе в переменной *Groups*. Элементы списка разделяются запятой. Переменная расположена в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>/Systems/<имя системы>.





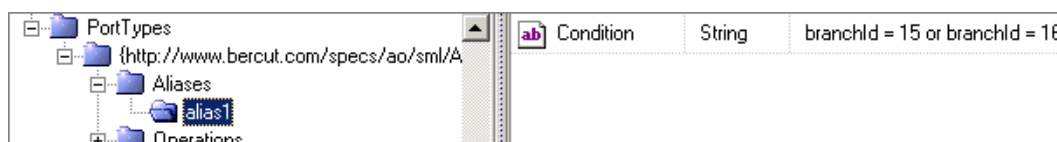
**Рис. 157.** Группа CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>/Systems/<имя системы>

### 12.2.1.1. Настройка доступа к операциям по бизнес-параметрам

Для настройки доступа к операциям по бизнес-параметрам используйте алиасы.

Порядок настройки:

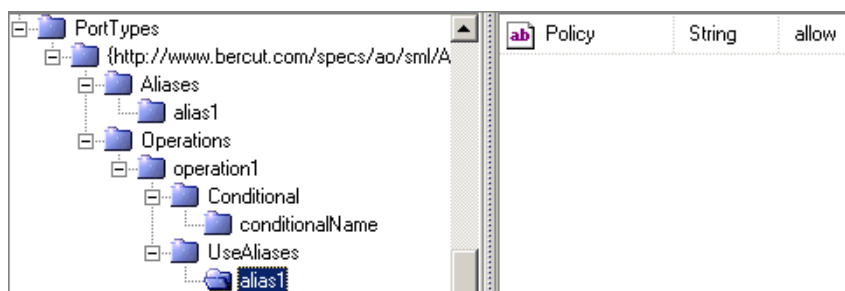
1. Создайте алиасы в MIB-группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Aliases. Для каждого алиаса задайте XPath-выражение, которое задает условие авторизации для операции, в переменной *Condition*.



**Рис. 158.** Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Aliases/<имя алиаса>

2. Задайте политику доступа к операции в переменной *Policy* группы CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<имя операции>/UseAliases/<имя алиаса>:

- allow — операция доступна при выполнении условия, указанного в группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Aliases/<имя алиаса>, независимо от политики доступа portType и политики доступа операции.
- deny — операция недоступна при выполнении условия, указанного в группе CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Aliases/<имя алиаса>, независимо от политики доступа portType и политики доступа операции.



**Рис. 159.** Группа CMP\_SG\_vX.Y/AA/Configuration/Groups/<имя группы>/PortTypes/<имя portType>/Operations/<имя операции>/UseAliases/<имя алиаса>

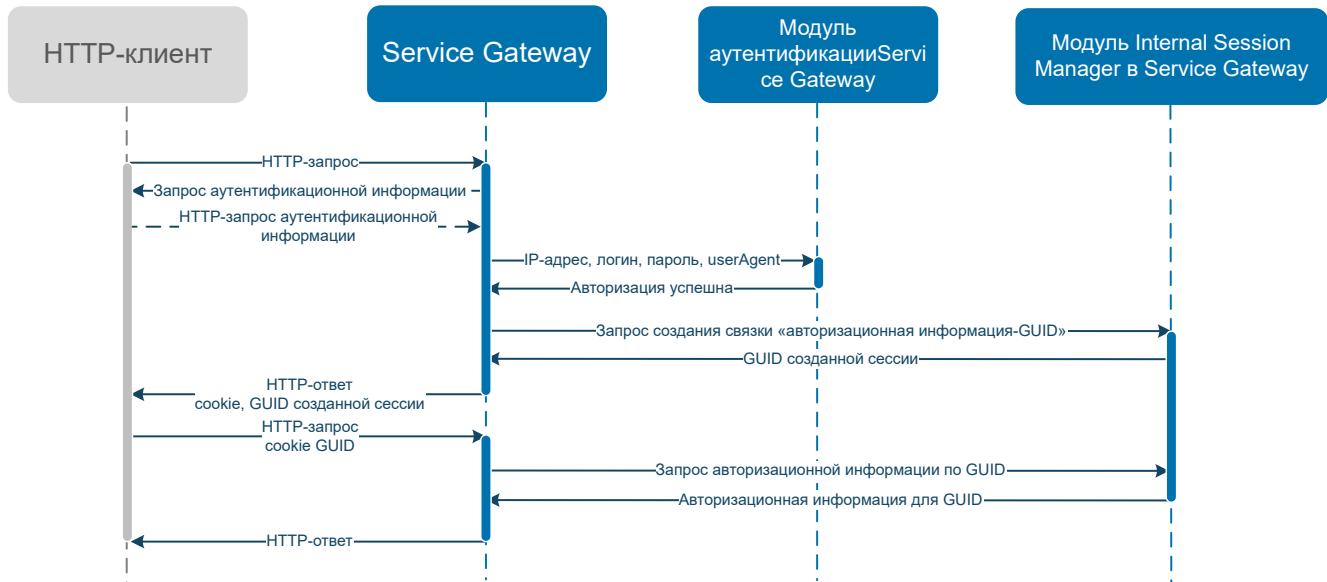
### 12.2.1.2. Принцип работы менеджера сессий в Service Gateway

При авторизации системы в Service Gateway вы можете использовать следующие типы менеджера сессий: *Internal*, *External* или *Off*.

Тип менеджера сессий для каждой системы задается в MIB-параметре *SessionManager* группы `CMR_SG_vX.Y/AA/Configuration/Systems/<имя системы>`.

#### Тип менеджера сессий — Internal

Используется внутренний менеджер сессий, реализованный в Service Gateway.



**Рис. 160. Работа менеджера сессий типа Internal**

При авторизации система выполняет следующие действия:

1. HTTP-клиент отправляет HTTP-запрос в Service Gateway.
2. Service Gateway запрашивает информацию для аутентификации у HTTP-клиента. *Тип доступа к информации*: basic или digest.
3. HTTP-клиент отправляет HTTP-запрос в Service Gateway. Запрос содержит информацию для аутентификации.
4. Service Gateway передает в модуль аутентификации следующие данные:
  - IP-адрес
  - логин
  - пароль
  - userAgent
  - другие данные.
5. Модуль аутентификации отправляет ответ об успешной авторизации. Ответ содержит информацию для авторизации.
6. Service Gateway отправляет запрос на создание связки *информация для авторизации-GUID* в модуль Internal Session Manager.
7. Модуль Internal Session Manager возвращает GUID созданной сессии.
8. Service Gateway отправляет HTTP-клиенту HTTP-ответ, который содержит:
  - GUID созданной сессии
  - cookie.
9. В дальнейшем HTTP-клиент отправляет HTTP-запрос на Service Gateway со следующими данными: GUID и cookie.

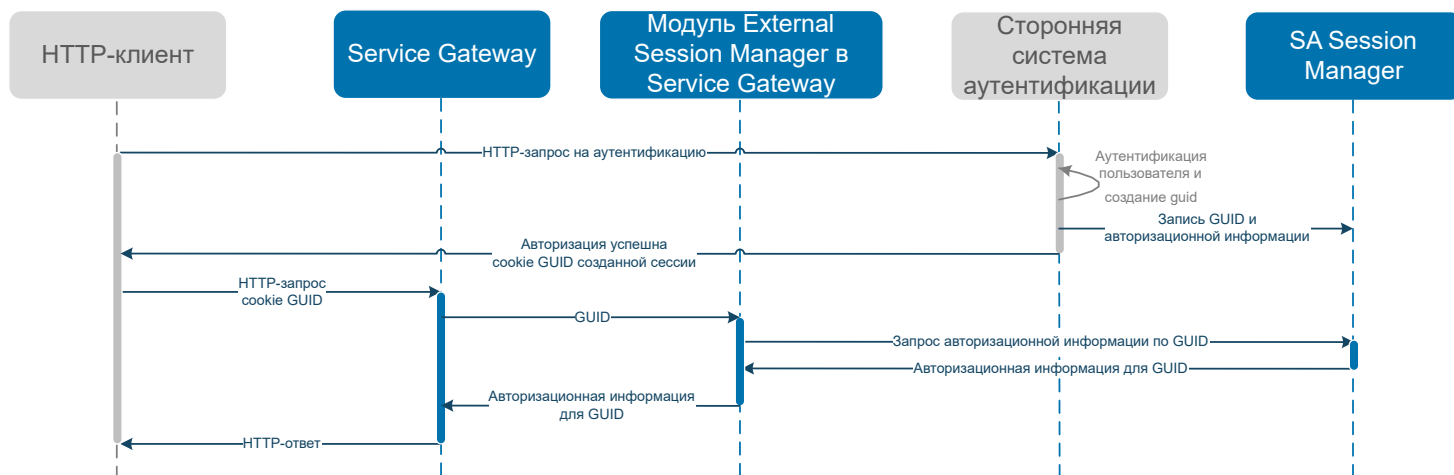
10. Service Gateway отправляет запрос на получение информации для авторизации по GUID в модуль Internal Session Manager.
11. Модуль Internal Session Manager возвращает информацию для авторизации в Service Gateway.
12. Service Gateway отправляет HTTP-клиенту HTTP-ответ.

Используйте этот тип менеджера сессий для экономии ресурсов и сетевого взаимодействия.

**Примечание.** Данный тип менеджера сессий можно использовать, если HTTP-клиент правильно обрабатывает cookie.

### Тип менеджера сессий — External

Используется внешний менеджер сессий, реализованный в portType {http://www.bercut.com/specs/aoi/session-management}session-management-port-type.



**Рис. 161. Работа менеджера сессий типа External**

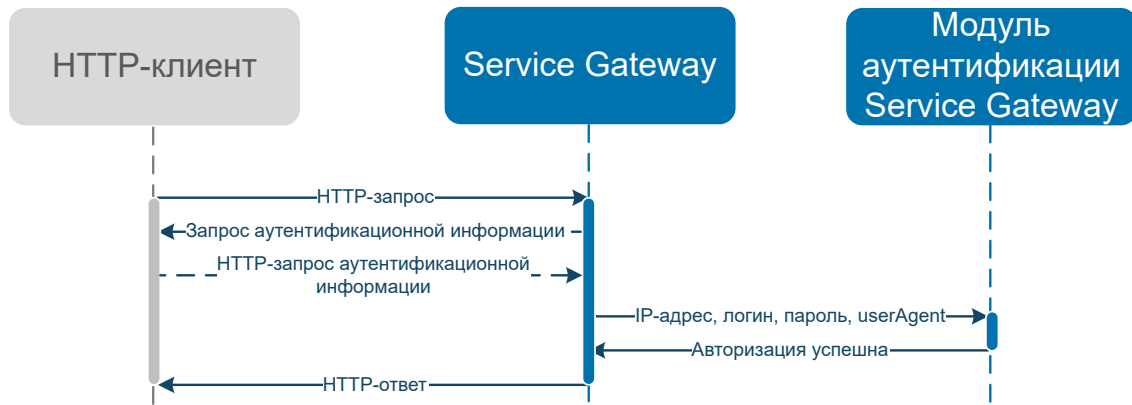
При авторизации система выполняет следующие действия:

1. HTTP-клиент отправляет HTTP-запрос на аутентификацию в стороннюю систему.
2. Сторонняя система выполняет аутентификацию пользователя и создание GUID.
3. Сторонняя система записывает в SA Session Manager следующую информацию:
  - GUID
  - информацию для авторизации.
4. Сторонняя система отправляет ответ об успешной авторизации на HTTP-клиент. Ответ содержит cookie GUID созданной сессии.
5. HTTP-клиент отправляет HTTP-запрос в Service Gateway. Запрос содержит cookie GUID.
6. Service Gateway передает GUID в модуль External Session Manager.
7. Модуль External Session Manager возвращает данные авторизации для GUID.
8. Service Gateway отправляет HTTP-клиенту HTTP-ответ.

Используйте этот тип менеджера сессий для интеграции со сторонней системой аутентификации. Например: при OAuth-аутентификации. При этом потребуется выполнить работы по сопряжению со сторонней системой аутентификации.

### Тип менеджера сессий — Off

Менеджер сессий отключен. Вызов выполняется с использованием аутентификации. Значение используется, если клиенты системы не обрабатывают cookie 'GUID'.



**Рис. 162. Работа менеджера сессий типа Off**

При авторизации система выполняет следующие действия:

1. HTTP-клиент отправляет HTTP-запрос в Service Gateway.
2. Service Gateway запрашивает информацию для аутентификации у HTTP-клиента. *Тип доступа к информации*: basic или digest.
3. HTTP-клиент отправляет HTTP-запрос в Service Gateway. Запрос содержит информацию для аутентификации.
4. Service Gateway передает в свой модуль аутентификации следующие данные:
  - IP-адрес
  - логин
  - пароль
  - userAgent
  - другие данные для аутентификации.
5. Модуль аутентификации отправляет ответ об успешной авторизации. Ответ содержит информацию для авторизации.
6. Service Gateway отправляет HTTP-клиенту HTTP-ответ.

Используйте этот тип менеджера сессий для экономии памяти и других ресурсов клиентских сессий. Рекомендуется использовать этот тип, если клиент не поддерживает работу с cookie или тип аутентификации отличен от 'basic' или 'digest'.

### 12.2.2. Настройка параметров аутентификации и авторизации в профиле Service Gateway

При значении 'spgw' переменных GroupsProvider, SystemsProvider, UsersProvider параметры аутентификации и авторизации извлекаются из профиля Service Gateway, определенного в SA SP.

Переменные расположены в группе CMP\_SG\_vX.Y/AA/Configuration. В профиле Service Gateway задаются параметры групп доступа, пользователей и систем в теге **parametersSG**.

Порядок настройки:

1. Задайте группы доступа для аутентификации в теге **parametersSG** блок **group**.  
Задайте параметры для группы:
  - Имя группы в теге **name**.
  - Статус группы в теге **enabled**:
    - True — группа активна.
    - False — группа неактивна.

- Список `portType` для настройки персональной политики доступа в теге `portType`.

Задайте параметры для `portType`:

- Имя `portType` в теге `name`.
- Политика доступа для `portType` в теге `policy`:
  - `allow` — все операции `portType`, кроме отдельно запрещенных, доступны для группы.
  - `deny` — все операции `portType`, кроме отдельно разрешенных, недоступны для группы.
- Список операций для настройки персональной политики доступа в теге `operation`.

Задайте параметры для операции:

- Название операции в теге `name`.
- Политика доступа для операции в теге `policy`:
  - `allow` — операция доступна для группы, независимо от политики доступа `portType`.
  - `deny` — операция недоступна для группы, независимо от политики доступа `portType`.
- Список условий доступа в теге `condition`.

Задайте параметры для условия:

- Название условия в теге `name`.
- XPath-выражение, которое задает условие авторизации для операции, в теге `rule`.
- Политика доступа для условия в теге `policy`:
  - `allow` — операция доступна при выполнении условия, независимо от политики доступа `portType` и политики доступа операции.
  - `deny` — операция недоступна при выполнении условия, независимо от политики доступа `portType` и политики доступа операции.

Для доступа к операции по бизнес-параметрам можно задать алиас в теге `Alias`:

- Название алиаса в теге `name`.
- XPath-выражение, которое задает условие авторизации для операции, в теге `rule`.

Политику доступа для условия при использовании алиаса задайте в теге `policy` блока `UseAlias`:

- `allow` — операция доступна при выполнении условия, независимо от политики доступа `portType` и политики доступа операции.
- `deny` — операция недоступна при выполнении условия, независимо от политики доступа `portType` и политики доступа операции.

Блок `group`:

```
<tns:group>
  <tns:name>NameGroup</tns:name>
  <tns:enabled>StatusGroup</tns:enabled>
  <tns:portType>
```

```

<tns:name>portType1 for Group</tns:name>
  <tns:alias>
    <tns:name>NameAlias</tns:name>
    <tns:condition>
      <sd:name>NameCondition</sd:name>
      <sd:rule>Rule for Condition</sd:rule>
    </tns:condition>
  </tns:alias>
  <tns:policy>AccessPolicy for portType1</tns:policy>
<tns:operation>
  <tns:name>NameOperation1</tns:name>
  <tns:policy>AccessPolicy for Operation1</tns:policy>
  <tns:UseAlias>
    <tns:policy>AccessPolicy for Condition</tns:policy>
  </tns:UseAlias>
  <tns:condition>
    <sd:name>NameCondition1</sd:name>
    <tns:policy>AccessPolicy for Condition1</tns:policy>
    <sd:rule>Rule for Condition1</sd:rule>
  </tns:condition>
  ...
  <tns:condition>
    <sd:name>NameConditionN</sd:name>
    <tns:policy>AccessPolicy for ConditionN</tns:policy>
    <sd:rule>Rule for ConditionN</sd:rule>
  </tns:condition>
</tns:operation>
...
<tns:operation>
  <tns:name>NameOperationM</tns:name>
  <tns:policy>AccessPolicy for OperationM</tns:policy>
  <tns:condition>
    <sd:name>NameCondition1</sd:name>
    <tns:policy>AccessPolicy for Condition1</tns:policy>
    <sd:rule>Rule for Condition1</sd:rule>
  </tns:condition>
  ...
  <tns:condition>
    <sd:name>NameConditionN</sd:name>
    <tns:policy>AccessPolicy for ConditionN</tns:policy>
    <sd:rule>Rule for ConditionN</sd:rule>
  </tns:condition>
</tns:operation>
</tns:portType>
...
<tns:portType>
<tns:name>portTypeK for Group</tns:name>
  <tns:policy>AccessPolicy for portTypeK</tns:policy>
  <tns:operation>
    <tns:name>NameOperation1</tns:name>
    <tns:policy>AccessPolicy for Operation1</tns:policy>
    <tns:condition>
      <sd:name>NameCondition1</sd:name>
      <tns:policy>AccessPolicy for Condition1</tns:policy>
      <sd:rule>Rule for Condition1</sd:rule>
    </tns:condition>
    ...
    <tns:condition>
      <sd:name>NameConditionN</sd:name>
      <tns:policy>AccessPolicy for ConditionN</tns:policy>
      <sd:rule>Rule for ConditionN</sd:rule>
    </tns:condition>
  </tns:operation>
  ...
  <tns:operation>
    <tns:name>NameOperationM</tns:name>
    <tns:policy>AccessPolicy for OperationM</tns:policy>
    <tns:condition>
      <sd:name>NameCondition1</sd:name>
      <tns:policy>AccessPolicy for Condition1</tns:policy>
      <sd:rule>Rule for Condition1</sd:rule>
    </tns:condition>
    ...
    <tns:condition>
      <sd:name>NameConditionN</sd:name>

```

```

    <tns:policy>AccessPolicy for ConditionN</tns:policy>
      <sd:rule>Rule for ConditionN</sd:rule>
    </tns:condition>
  </tns:operation>
</tns:portType>
</tns:group>

```

2. Задайте параметры доступа к внешним системам или сервисам, которые обращаются к Service Gateway, в теге *parametersSG* блок *system*.

Задайте параметры для системы:

- Имя системы в теге *name*.
- Статус системы в теге *enabled*:
  - True — система активна.
  - False — система неактивна.
- Тип доступа в теге *authenticationType*:
  - trust — аутентификация по имени пользователя не выполняется для всех запросов, отправленных системой. Задайте список доступных для системы групп в теге *TrustedGroups*.
  - basic — выполняется аутентификация HTTP basic по имени пользователя для всех запросов, отправленных системой.
  - digest — аутентификация не выполняется для всех запросов, отправленных системой.
  - off — используется для внешнего менеджера сессий. Аутентификация не выполняется. Система читает cookie с именем **guid** и проверяет в менеджере сессий наличие созданной активной сессии с указанным идентификатором.
- Список разрешенных IP-адресов в теге *allowedIp*. Элементы списка разделяются запятой. Можно использовать диапазоны и маски, например: `192.168.1.0-192.168.1.55` или `192.168.1.*`.
- Максимальное количество одновременных пользовательских сессий в теге *maximumActiveSessions*. Идентификатор пользовательской сессии передается в cookie **guid**.
- Тип менеджера сессий для данной системы в теге *sessionManagerType*:
  - internal — внутренний менеджер сессий, реализованный в Service Gateway.
  - external — внешний менеджер сессий, реализованный в portType {`http://www.bercut.com/specs/aoi/session-management`}*session-management-port-type*.
  - off — менеджер сессий отключен. Вызов выполняется с использованием аутентификации. Значение используется, если клиенты системы не обрабатывают cookie **guid**.
- Список доступных для данной системы групп в теге *groupName*. Элементы списка разделяются запятой. Список используется при значении 'trust' параметра *authenticationType*. Параметр необязательный.
- Имя агента, передаваемое в HTTP-заголовке, в теге *userAgent*. Параметр необязательный.
- Максимальное количество сообщений, отправляемых всеми пользователями системы за секунду, в теге *messagesPerSecond*. При достижении этого значения Service Gateway прекращает принимать информацию от всех пользователей системы до окончания текущей секунды. Параметр необязательный.
- Тип ограничения максимального количества сообщений в секунду для системы в теге *messagesPerSecondType*:
  - slow — система использует *медленный* канал.

- reject — система отправляет HTTP-ошибку '509'.

Параметр необязательный.

Блок **system**:

```
<tns:system>
  <tns:name>NameSystem</tns:name>
  <pd:serviceAuth>
    <pd:enabled>StatusSystem</pd:enabled>
    <pd:authenticationType>AuthenticationType</pd:authenticationType>
    <pd:allowedIp>SystemIPAddress</pd:allowedIp>
    <pd:maximumActiveSessions>MaxActiveSession for System</pd:maximumActiveSessions>
    <pd:sessionManagerType>SessionManagerType</pd:sessionManagerType>
    <pd:userAgent>UserAgentName</pd:userAgent>
    <pd:groupName>SystemGroupsList</pd:groupName>
    <pd:messagesPerSecond>MaxMessagesPerSecond for System</pd:messagesPerSecond>
    <pd:messagesPerSecondType>messagesPerSecondType</pd:messagesPerSecondType>
  </pd:serviceAuth>
</tns:system>
```

### 3. Задайте параметры пользователей в теге **parametersSG** блок **user**.

Задайте параметры для пользователя:

- Имя пользователя в теге **name**.
- Пароль пользователя в теге **password**.
- Политика доступа в теге **policy**:
  - allow — пользователю доступна аутентификация на Service Gateway.
  - deny — пользователю недоступна аутентификация на Service Gateway.
- Список доступных пользователю групп в теге **groupName**. Элементы списка разделяются запятой.
- Максимальное количество сообщений, отправляемых пользователем за секунду, в теге **messagesPerSecond**. При достижении этого значения Service Gateway прекращает принимать информацию от пользователя до окончания текущей секунды. Параметр необязательный.
- Тип ограничения максимального количества сообщений в секунду для пользователя в теге **messagesPerSecondType**:
  - slow — система использует *медленный* канал.
  - reject — система отправляет HTTP-ошибку '509'.

Параметр необязательный.

- Список систем при аутентификации пользователя в этих системах в теге **systemGroups**.

Задайте параметры для системы:

- Имя системы в теге **systemName**.
- Список доступных пользователю групп в системе в теге **groupName**. Элементы списка разделяются запятой.

Блок **user**:

```
<tns:user>
  <tns:name>NameUser</tns:name>
  <sd:password>PasswordUser</sd:password>
  <tns:policy>AccessPolicy for User</tns:policy>
  <tns:groupName>GroupsList for User</tns:groupName>
  <tns:systemGroups>
    <sd:systemName>NameSystem1 for User</sd:systemName>
    <sd:groupName>GroupsList for System1</sd:groupName>
  </tns:systemGroups>
</tns:user>
```



```

</tns:systemGroups>
...
<tns:systemGroups>
  <sd:systemName>NameSystemN for User</sd:systemName>
  <sd:groupName>GroupsList for SystemN</sd:groupName>
</tns:systemGroups>
</tns:user>

```

### Пример создания сервиса ServiceGateway:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
  xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:createServiceRequestParams>
      <sim:branchId>1</sim:branchId>
      <sim:serviceName>ServiceGateway</sim:serviceName>
      <sim:serviceStatus>active</sim:serviceStatus>
      <sim:tag>SG</sim:tag>
      <!--Optional:-->
      <ser:serviceContent>
        <tns:ServiceGateway xmlns:tns="http://www.bercut.com/spec/schema/plv3/
ServiceGateway"
          xmlns:pd="http://www.bercut.com/spec/schema/ProfileDefinition"
          xmlns:cd="http://www.bercut.com/spec/schema/ComplexDefinition"
          xmlns:sd="http://www.bercut.com/spec/schema/SimpleDefinition">
          <tns:parametersSG>
            <tns:group>
              <tns:name>group1</tns:name>
              <tns:enabled>True</tns:enabled>
              <tns:portType>
                <tns:name>{http://www.bercut.com/spec/sp/
AOIDeployer}SchemaDeployerPortType</tns:name>
                <tns:policy>allow</tns:policy>
                <tns:operation>
                  <tns:name>endOperation</tns:name>
                  <tns:policy>allow</tns:policy>
                  <tns:condition>
                    <sd:name>1</sd:name>
                    <tns:policy>allow</tns:policy>
                    <sd:rule>one_string=10</sd:rule>
                  </tns:condition>
                </tns:operation>
              </tns:portType>
            </tns:group>
            <tns:group>
              <tns:name>group2</tns:name>
              <tns:enabled>False</tns:enabled>
              <tns:portType>
                <tns:name>{http://www.bercut.com/spec/dsi/
ServiceProfileManagement}ServiceProfilePortType</tns:name>
                <tns:policy>allow</tns:policy>
              </tns:portType>
            </tns:group>
            <tns:system>
              <tns:name>system1</tns:name>
              <pd:serviceAuth>
                <pd:enabled>True</pd:enabled>
                <pd:authenticationType>digest</pd:authenticationType>
                <pd:allowedIp>127.0.0.1</pd:allowedIp>
                <pd:userAgent>agent1</pd:userAgent>
                <pd:maximumActiveSessions>10</pd:maximumActiveSessions>
                <pd:sessionManagerType>internal</pd:sessionManagerType>
              </pd:serviceAuth>
            </tns:system>
            <tns:system>
              <tns:name>system2</tns:name>
              <pd:serviceAuth>
                <pd:enabled>True</pd:enabled>
                <pd:authenticationType>trust</pd:authenticationType>
                <pd:allowedIp>127.0.0.1</pd:allowedIp>
                <pd:maximumActiveSessions>10</pd:maximumActiveSessions>
              </pd:serviceAuth>
            </tns:system>
          </tns:ServiceGateway>
        </ser:serviceContent>
      </tns:ServiceGateway>
    </ser:createServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        <pd:sessionManagerType>external</pd:sessionManagerType>
        <pd:userAgent>agent2</pd:userAgent>
        <pd:groupName>group1</pd:groupName>
        <pd:messagesPerSecond>10</pd:messagesPerSecond>
        <pd:messagesPerSecondType>slow</pd:messagesPerSecondType>
    </pd:serviceAuth>
</tns:system>
<tns:user>
    <tns:name>user1</tns:name>
    <sd:password>123456</sd:password>
    <tns:policy>allow</tns:policy>
    <tns:groupName>group1</tns:groupName>
    <tns:systemGroups>
        <sd:systemName>system1</sd:systemName>
        <sd:groupName>group1</sd:groupName>
    </tns:systemGroups>
</tns:user>
<tns:user>
    <tns:name>user2</tns:name>
    <sd:password>password2</sd:password>
    <tns:policy>allow</tns:policy>
    <tns:groupName>group2</tns:groupName>
    <tns:systemGroups>
        <sd:systemName>system1</sd:systemName>
        <sd:groupName>group2</sd:groupName>
    </tns:systemGroups>
</tns:user>
</tns:parametersSG>
</tns:ServiceGateway>
</ser:serviceContent>
</ser:createServiceRequestParams>
</soapenv:Body>
</soapenv:Envelope>

```

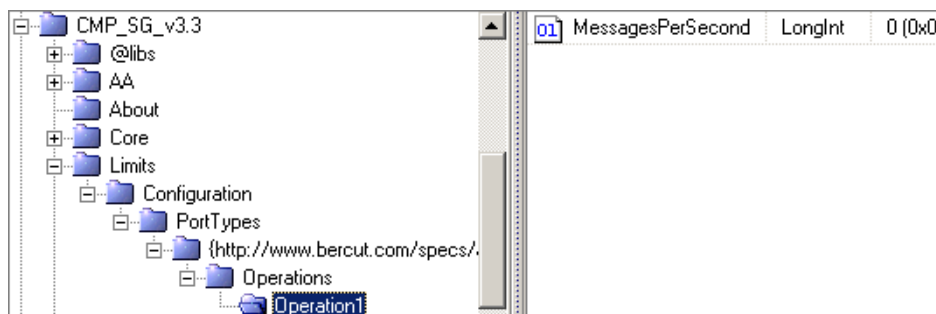
### 12.2.3. Настройка ограничений трафика во внутренней сети

Для настройки ограничений трафика используйте переменные MIB-группы *CMP\_SG\_vX.Y/Limits*.

Пользователь может настроить ограничения трафика, пропускаемого во внутреннюю сеть. Максимальное количество сообщений в секунду задается в переменной *MessagesPerSecond*. Ограничение можно задать на следующих уровнях:

- для приложения в группе *CMP\_SG\_vX.Y/Limits/Configuration*;
- для *portType* в группе *CMP\_SG\_vX.Y/Limits/Configuration/PortTypes/<имя portType>*;
- для операции в группе *CMP\_SG\_vX.Y/Limits/Configuration/PortTypes/<имя portType>/Operations/<имя операции>*.

При превышении ограничения по одному из уровней система не пропускает вновь поступившее сообщение. Система отправляет клиенту код '509 — Bandwidth reject'.



**Рис. 163.** Группа *CMP\_SG\_vX.Y/Limits/Configuration/PortTypes/<имя portType>/Operations/<имя операции>*

#### 12.2.4. Настройка автоматического обновления интерфейсов

Для настройки автоматического обновления интерфейсов используйте параметры MIB-группы *CMP\_SG\_vX.Y/Core/Configuration/Publisher*.

При изменении статуса маршрута система перечитывает WSDL-описания компонентов, которые по доступны по этому маршруту, с интервалом 5 секунд.

Алгоритм публикации WSDL-описания зависит от выявленных ошибок доступности сервисов или совместимости интерфейсов:

- При отсутствии ошибок для каждого сервиса будет опубликовано WSDL-описание самой старой версии.
- При ошибке совместимости интерфейсов будет опубликован интерфейс самой старой неконфликтной версии. Пример: с разных соединений сервиса получены версии 1.0, 1.1, 1.2, 1.3. Версия 1.1 совместима с 1.0, версия 1.2 несовместима с 1.1, версия 1.3 совместима с 1.2. Система опубликует версию 1.1.
- При несовместимости интерфейсов одного сервиса и одной версии система выбирает интерфейс случайным образом.
- Если не удалось отправить WSDL-описание или истек тайм-аут ожидания ответа, Service Gateway анализирует полученные интерфейсы и выбирает версию для публикации среди них.

Все ошибки отображаются в трейсе системы Service Gateway.

Задайте значения переменных группы *CMP\_SG\_vX.Y/Core/Configuration/Publisher*:

- *EventUpdateWsdI*. Флаг обновления WSDL-описаний при наступлении следующих событий:
  - add route
  - remove route
  - unavailable
  - available
  - add connection
  - remove connection
  - connect
  - disconnect
- *ManualUpdateWsdI*. Флаг принудительного обновления WSDL-описаний.
- *OnlyManualUpdateWsdI*. Флаг обновления WSDL-описаний вручную с помощью параметра *ManualUpdateWsdI*. При значении True система отключает обновления при наступлении определенных событий и по расписанию.

- *UpdateWsdIPeriod*. Интервал принудительного обновления публикуемых WSDL-описаний, в минутах. При значении '0' — периодически не обновляются.
- *AddOldStyleAliases*. Флаг добавления идентификатора URI в старом виде (версии 3.0 и 3.1) для публикуемых portType.

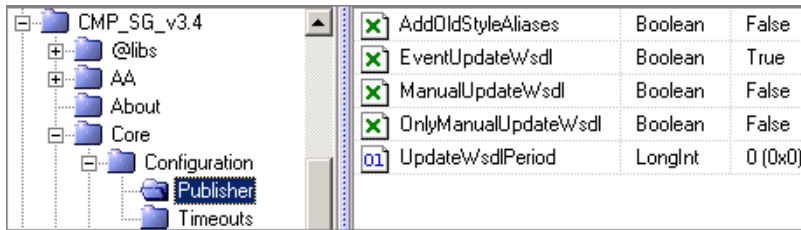


Рис. 164. Группа CMP\_SG\_vX.Y/Core/Configuration/Publisher

### 12.2.5. Настройка трассировки сообщений

Настройки трассировки хранятся в MIB-группе SG\_X.Y/Tracer/Configuration.

Вы можете настроить трассировку сообщений:

- для всех portType. MIB-группа SG\_X.Y/Tracer/Configuration.
- для portType. MIB-группа SG\_X.Y/Tracer/Configuration/PortTypes/<имя portType>.
- для операции portType. MIB-группа SG\_X.Y/Tracer/Configuration/PortTypes/<имя portType>/Operations/<имя операции>.
- для фильтра по бизнес-параметрам операции portType. MIB-группа SG\_X.Y/Tracer/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>.

В каждой группе задайте значения следующих MIB-параметров:

- *Enabled* — флаг включения трассировки для соответствующего объекта:
  - True — трассировка включена;
  - False — трассировка выключена. Значение по умолчанию.
- *Sampling* — порядковый номер сообщений для трассировки. Например: при значении '5' выполняется трассировка каждого пятого сообщения. Значение используется при поставленном флаге *Enabled*.
- *BusinessApplication* — имя бизнес-приложения для соответствующего объекта.

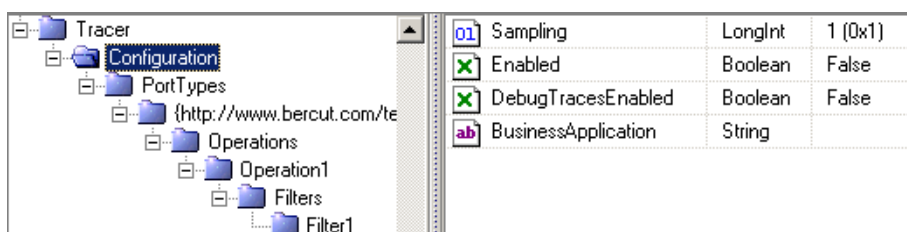


Рис. 165. Настройки трассировки сообщений

Чтобы включить трассировку по всей цепочке вызовов, поставьте флаг *Propagate* в группе SG\_X.Y/Tracer/Configuration. Если в переменной задано значение 'False',

то трассировка выполняется только на текущем узле в соответствии с переменной *Sampling*.

Для всех portType вы можете включить отображение сообщений уровня Debug. Для этого поставьте флаг *DebugTracesEnabled* в группе *SG\_X.Y/Tracer/Configuration*. В сообщениях уровня Debug отображается применение трассировки для конкретных сообщений.

Для фильтра по бизнес-параметрам операции portType задайте условие срабатывания фильтра. Задайте XPath-выражение в MIB-параметре *Condition* группы *SG\_X.Y/Tracer/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>*.

Наивысший приоритет имеют настройки для фильтра, затем для операции, для portType и наименьший приоритет у настроек для всех portType.

### 12.2.6. Управляющие системные SOAP-заголовки

Service Gateway использует следующие управляющие системные SOAP-заголовки:

- Заголовок включения пользовательского логирования текущей сессии.
- Заголовок включения трассировки текущей сессии.

Системные заголовки находятся в пространстве имен {http://bercut.com/}.

Чтобы включить трассировку, добавьте следующие заголовки во входящий пакет:

- Заголовок пользовательского логирования.

```
<soapenv:Header xmlns:ber="http://bercut.com/" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ber:customLogging>debug</ber:customLogging>
</soapenv:Header>
```

- Заголовок трассировки.

```
<soapenv:Header xmlns:ber="http://bercut.com/" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ber:tracing>Имя бизнес-приложения</ber:tracing>
</soapenv:Header>
```

Чтобы Service Gateway обрабатывал заголовок, поставьте флаг *ForwardControlHeaders* для системы или пользователя (12.2.1). Переданное имя бизнес-приложения будет передано в ATLAS-контейнер и отправлено далее по цепочке вызовов. Логирование и трассировка такой цепочки выполняется от имени указанного бизнес-приложения на всех компонентах, которые участвуют в обработке запроса.

### 12.2.7. Настройка тайм-аутов

Тайм-ауты задаются в группе *SG\_vX.Y/Core/Configuration/Timeouts*.

Задайте следующие тайм-ауты в группе *SG\_vX.Y/Core/Configuration/Timeouts*:

- *DefaultHttpSessionTimeout*. Тайм-аут активности HTTP-сессий в минутах. После окончания тайм-аута Service Gateway считает сессию завершенной. Значение по умолчанию: '10'.
- *StartHttpServerTimeout*. Тайм-аут перед запуском HTTP(S)-сервера. Если все внешние portType опрошены до окончания тайм-аута, сервер запускается раньше. Значение по умолчанию: '30'.



**Рис. 166. Тайм-ауты SG**

### 12.2.8. Статистические параметры

Система ATLAS собирает статистические данные по работе Service Gateway.

Статистика по входящим сообщениям хранится в MIB-группах:

- Core/Statistics/ProxyInfo/Users/<имя пользователя> — по пользователю;
- Core/Statistics/ProxyInfo/Users/<имя пользователя>/IncomingPortTypes/<имя portType> — по portType, который вызывал пользователь;
- Core/Statistics/ProxyInfo/Users/<имя пользователя>/IncomingPortTypes/<имя portType>/Operations/<имя операции> — по операциям portType, которые вызывал пользователь;
- Core/Statistics/ProxyInfo/Users/<имя системы> — по системе;
- Core/Statistics/ProxyInfo/Systems/<имя системы>/IncomingPortTypes/<имя portType> — по portType, которые вызывались системой;
- Core/Statistics/ProxyInfo/Systems/<имя системы>/IncomingPortTypes/<имя portType>/Operations/<имя операции> — по операциям portType, которые вызывались системой.

<input type="checkbox"/>	BandwidthRejectedCounter	Dword64	0 (0x0)
<input type="checkbox"/>	BandwidthRejectedRateAvg	Dword64	0 (0x0)
<input type="checkbox"/>	BandwidthRejectedRateMax	Dword64	0 (0x0)
<input type="checkbox"/>	CanSendResponseCounter	Dword64	0 (0x0)
<input type="checkbox"/>	CanSendResponseRateAvg	Dword64	0 (0x0)
<input type="checkbox"/>	CanSendResponseRateMax	Dword64	0 (0x0)
<input type="checkbox"/>	InBandwidthRateAvg	Double	0
<input type="checkbox"/>	InBandwidthRateMax	Double	0
<input type="checkbox"/>	OutBandwidthRateAvg	Double	0
<input type="checkbox"/>	OutBandwidthRateMax	Double	0
<input type="checkbox"/>	OverQoSResponseCounter	Dword64	0 (0x0)
<input type="checkbox"/>	OverQoSResponseRateAvg	Dword64	0 (0x0)
<input type="checkbox"/>	OverQoSResponseRateMax	Dword64	0 (0x0)
<input type="checkbox"/>	RejectedCounter	Dword64	0 (0x0)
<input type="checkbox"/>	RejectedRateAvg	Dword64	0 (0x0)
<input type="checkbox"/>	RejectedRateMax	Dword64	0 (0x0)
<input type="checkbox"/>	RequestCounter	Dword64	0 (0x0)
<input type="checkbox"/>	RequestRateAvg	Dword64	0 (0x0)
<input type="checkbox"/>	RequestRateMax	Dword64	0 (0x0)
<input type="checkbox"/>	RequestResponseTimeAvg	Dword	0 (0x0)
<input type="checkbox"/>	RequestResponseTimeMax	Dword	0 (0x0)
<input type="checkbox"/>	ResponseCounter	Dword64	0 (0x0)
<input type="checkbox"/>	ResponseRateAvg	Dword64	0 (0x0)
<input type="checkbox"/>	ResponseRateMax	Dword64	0 (0x0)
<input type="checkbox"/>	WorkingThreadTimeAvg	Dword	0 (0x0)
<input type="checkbox"/>	WorkingThreadTimeMax	Dword	0 (0x0)

**Рис. 167. Статистика по пользователю**

Список переменных аналогичен [списку переменных](#) для входящих сообщений RTSIB.

## 13. Система Subscriber Service Profile

*Subscriber Service Profile* хранит данные по сервисам для конкретного абонента.

### 13.1. Установка Subscriber Service Profile Subsystem

Для установки *Subscriber Service Profile Subsystem* установите SSP DB и SA SSP.

**! Внимание!** Выполняйте установку *Subscriber Service Profile Subsystem* на LW SA Container версии не ниже 3.2.

Порядок установки SSP:

1. Разверните схему SSP DB в СУБД Oracle или PostgreSQL. Имя пользователя создаваемой схемы — произвольное. Рекомендуемое имя пользователя — **sspm**.
2. *Установите* LW SA Container версии не ниже 3.2.
3. *Подключите* SSP DB к LW SA Container. SSP DB транзакционная БД. Если SSP DB развернута в СУБД Oracle, настройте *JMS-соединение*.
4. *Подключите* SA SSP.

После установки *настройте* SSP.

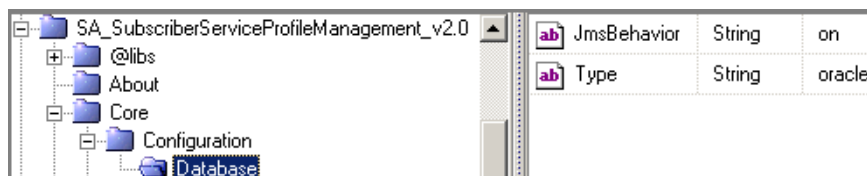
### 13.2. Настройка Subscriber Service Profile Subsystem

Для корректной работы *Subscriber Service Profile Subsystem* настройте соединение с SA SSP компонентов, использующих RTSIB.

Порядок настройки SSP:

1. Задайте MIB-параметры группы CMP\_LWSAContainer\_vX.Y/@artifacts/SA\_SubscriberServiceProfileManagement\_vX.Y/Core/Configuration/Database:
  - *Type*. Тип используемой БД:
    - oracle — используется Oracle DB.
    - postgresql — используется PostgreSQL.
  - *JmsBehavior*. Тип работы с JMS-очередью:
    - on — прием и отправка нотификаций.
    - off — прием и отправка нотификаций отключена.
    - send — отправка нотификаций.





**Рис. 168.** Группа CMP\_LWSAContainer\_vX.Y/@artifacts/SA\_SubscriberServiceProfileManagement\_vX.Y/Core/Configuration/Database

2. В группе CMP\_LWSAContainer\_vX.Y/JMS/ConnectionsList создайте вложенную группу с именем источника данных, заданном в файле server.xml.

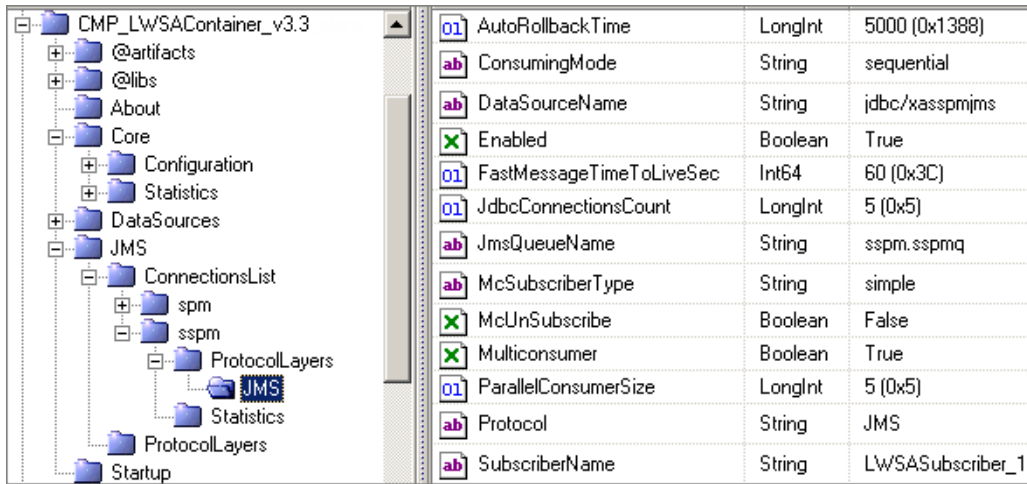
В созданной группе CMP\_LWSAContainer\_vX.Y/JMS/ConnectionsList/<имя источника данных>/ProtocolLayers/JMS задайте значения переменных:

- **DataSourceName.** JNDI-имя для подключения JMS, заданное при настройке [подключения к схеме данных](#).
- **JmsQueueName.** Имя JMS-очереди сообщений. Рекомендуемое значение: 'sspm.sspmq', где spsm — имя схемы данных SSP DB.
- **ConsumingMode.** Режим работы JMS-сообщений:
  - sequential — последовательная обработка сообщений, система не отправляет клиенту новое сообщение до его выхода из метода нотификации.
  - parallel — система вызывает метод нотификации из количества потоков, определенных в переменной *ParallelConsumerSize*. Подтверждение commit или rollback система отправляет на все считанные из очереди на текущий момент времени сообщения.
- **SubscriberName.** Имя durable-JMS-подписчика multiconsumer-очереди, использующего данное JMS-подключение. Значение должно быть уникальным среди других имен подписчиков данной очереди.
- **Enabled.** Статус JMS-подключения:
  - True — работает.
  - False — выключено.
- **ParallelConsumerSize.** Размер пула потоков в режиме 'parallel' переменной *ConsumingMode*. При значении '0' пул не ограничен.
- **FastMessageTimeToLiveSec.** Время жизни «быстрых» сообщений, в секундах. При значении '-1' время жизни не ограничено.
- **JdbcConnectionsCount.** Количество используемых подключений к БД.
- **Multiconsumer.** Тип очереди, к которой выполняется подключение. Рекомендуемое значение: True.
- **McSubscriberType.** Тип JMS-потребителей для multiconsumer-очереди:
  - durable — подписка с гарантированной доставкой получателю.
  - simple — доставка сообщений в режиме радио: если подписчик неактивен, сообщения для него не будут сохранены.

Рекомендуемое значение: 'durable'.

- **McUnSubscribe.** Режим отмены подписки:
  - True — система выполняет отмену подписки при отключении.
  - False — система оставляет подписку под именем, определенным в переменной *SubscriberName*, в очереди при отключении.

- *AutoRollbackTime*. Частота автоматической отправки rollback на входящее JMS-сообщение, в миллисекундах.



**Рис. 169.** Группа *CMP\_LWSAContainer\_vX.Y/JMS/ConnectionsList/<имя источника данных>/Protocollayers/JMS*

3. В значении переменной *DataSourceName* группы *CMP\_LWSAContainer\_vX.Y/Core/Configuration* укажите название XA-пула, заданного при настройке [подключения к схеме данных](#). Например: 'jdbc/xasspm'.
4. Настройте [маршрутизацию](#) для SSP.

## 14. Система Session Management

*Система [Session Management](#) хранит и обрабатывает сессии Platform v3.*

### 14.1. Установка системы Session Management

*Для установки системы [Session Management](#) установите [Session DB](#) и [SA SessionManagement](#).*

Порядок установки:

1. Разверните схему [Session DB](#) в СУБД Oracle или PostgreSQL. Имя пользователя создаваемой схемы — произвольное. Рекомендуемое имя пользователя — **session\_manager**.
2. [Установите](#) LW SA Container.
3. [Подключите](#) [Session DB](#) к LW SA Container. [Session DB](#) транзакционная БД.
4. [Подключите](#) [SA SessionManagement](#).

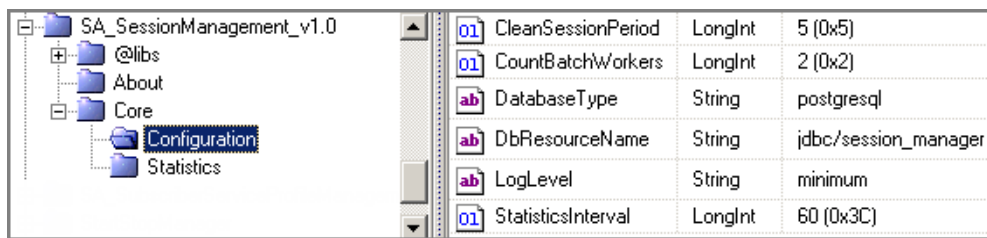
После установки [настройте](#) систему [Session Management](#).

### 14.2. Настройка системы Session Management

*Для корректной работы системы [Session Management](#) настройте соединение с [SA Session Management](#) компонентов, использующих [RTSIB](#).*

Порядок настройки:

1. Задайте MIB-параметры группы `CMP_LWSAContainer_vX.Y/@artifacts/SA_SessionManagement_vX.Y/Core/Configuration`:
  - **DatabaseType**. Тип используемой БД:
    - oracle — используется Oracle DB.
    - postgresql — используется PostgreSQL.
  - **DbResourceName**. Название ресурса пула соединений к [Session DB](#), заданного при [подключении](#) схемы данных. Например: 'jdbc/session\_manager'.
  - **CleanSessionPeriod**. Периодичность удаления устаревших сессий из БД, в минутах.
  - **CountBatchWorkers**. Количество потоков, ведущих пакетную обработку сессий: создание, обновление, удаление.



**Рис. 170. Группа CMP\_LWSAContainer\_vX.Y/@artifacts/SA\_Session Management\_vX.Y/Core/Configuration**

2. В значении переменной *DataSourceName* группы CMP\_LWSAContainer\_vX.Y/Core/Configuration укажите название XA-пула, заданного при **подключении** схемы данных. Например: 'jdbc/xasadbqa'.
3. Настройте **маршрутизацию** системы Session Management.

## 15. Система Operations&Management

Система [Operations&Management](#) выполняет мониторинг распределенных приложений.

**Таблица 4. Перечень компонентов, которые необходимы для функционирования системы Operations&Management**

Система	Версия
ATLAS Core	2.0.0
OMPostptocess	1.0.0
SLES	3.6.0
LW SAContainer	3.6.0
Service Gateway	3.6.0
Kafka	2.6.0
Zookeeper	3.6.1
Logstash-oss (лицензия Apache 2.0)	7.9.2
Elasticsearch-oss (лицензия Apache 2.0)	7.9.2
Grafana	7.3.1

Подсистема OMPostptocess включает в себя компоненты:

- logdecoder 1.5;
- tracedecoder 1.1;
- примеры конфигурационных файлов logstash;
- предустановленные настройки Grafana для отображения логов и трейсов.

### Форматы бинарных записей

**Таблица 5. Формат бинарной записи лога от агента в топике `log-<ВА name>`**

Название поля	Описание поля
Application	Имя сервиса BusinessApplication.
Instance	Идентификатор приложения.
SessionID	Идентификатор пользовательской сессии.
ClassName	Имя класса.
Version	Версия приложения.
LocalIP	IP-адрес локальной стороны.
Time	Время записи в абсолютном формате.
Type	Уровень сообщения: <ul style="list-style-type: none"> <li>▪ 1 — Error;</li> <li>▪ 2 — Warning;</li> <li>▪ 4 — Information;</li> <li>▪ 8 — Fatal;</li> </ul>

Название поля	Описание поля
	<ul style="list-style-type: none"> <li>16 — Debug.</li> </ul>
ClientId	Идентификатор клиента.
DecoderId	Идентификатор декодера.
DecodingScheme	Схема декодирования.
AgentTimeSec	Время получения агентом сообщения, в секундах.
AgentTimeUsec	Время получения агентом сообщения, в микросекундах.

**Таблица 6. Формат бинарной записи трейса от агента в топике `trace-<ВА name>`**

Название поля	Описание поля
Application	Имя сервиса BusinessApplication.
Flags	MIB-флаги.
MessageType	Тип сообщения: <ul style="list-style-type: none"> <li>outgoing request</li> <li>incoming response.</li> </ul>
SessionID	Идентификатор пользовательской сессии.
invokeChain	Цепочка вызовов. Пример значения: '1,2,1,3'.
ActorName	MIB-имя действующего компонента: SA, SE, BP. Если MIB-имя определить невозможно, в поле указывается MIB-имя действующей системы: <ul style="list-style-type: none"> <li>SLES — при входящих запросах;</li> <li>SG — для всех входящих и исходящих запросов и ответов.</li> </ul>
InterfaceName	Имя portType трейсового сообщения.
OperationName	Имя операции трейсового сообщения.
LocalIP	IP-адрес локальной стороны.
RemoteIP	IP-адрес удаленной стороны.
LocalPort	Номер порта локальной стороны.
RemotePort	Номер порта удаленной стороны.
CodingType	Тип кодирования тела сообщения: <ul style="list-style-type: none"> <li>0 — SOAP/XML</li> <li>1 — SOAP/FI</li> <li>2 — JSON.</li> </ul>
Time	Текущее время.
TraceID	Идентификатор сообщения.
Processing	Флаг обработки данных.

Название поля	Описание поля
Environment	Параметры окружения.

## 15.1. Установка и настройка системы ATLAS Agent

Задайте значения MIB-параметров группы *CMP\_Agent\_v2.0/Kafka/Configuration*.

Системы Platform v3 в процессе работы заполняют параметры логирования:

- идентификатор сессии;
- название бизнес-приложения.

Запись трейсов и алармов в java-приложениях выполняется с помощью классов, которые формируются из ATLAS-описания. Для записи используется LDC-компилятор `xml_compiler.bat`.

В ATLAS Agent версии 2.0 формируемые классы TRACE и ALARM дополнены:

- конструкторами с возможностью установки дополнительных общих параметров логирования;
- методами `setTrace` и `setAlarm` для установки дополнительных общих параметров логирования;
- вариантами методов логирования и аларминга для передачи дополнительных общих параметров логирования.

Порядок установки:

1. Установите [ATLAS Core](#).

2. Для подключения к серверу системы Kafka задайте MIB-параметры группы *CMP\_Agent\_v2.0/Kafka/Configuration*:

- `UseKafka`. Признак использования системы Kafka для передачи данных от приложений:
  - True — система Kafka используется.
  - False — система Kafka не используется.
- `Brokers`. Адреса брокеров или кластеров системы Kafka. Значения перечисляются через ','. Формат: `hostname:port`. Пример значения: `'kafka:9092'`.
- `GroupId`. Идентификатор группы consumer системы Kafka.
- `LocalIP`. IP-адрес локальной стороны. Если значение не задано, используется значение по умолчанию для системы.
- `StatisticInterval`. Интервал сбора статистики системы Kafka.
- `UseDisk`. Признак использования диска для записи логов:
  - True — диск используется.
  - False — диск не используется.
- `UseDiskOnKafkaError`. Признак использования диска, если система Kafka недоступна:
  - True — диск используется.
  - False — диск не используется.

Для работы с системой Kafka задайте значение True параметра *UseKafka* и задайте адреса серверов системы Kafka в параметре *Brokers*.

### 15.1.1. Настройка авторизации в Kafka

Чтобы настроить авторизацию в Kafka, создайте хранилище с публичным и приватным ключами, и задайте эти данные в MIB-параметрах по пути `/CMP_Agent_v2.0/Kafka/Configuration/SSL/`.

Порядок настройки:

#### 1. Создайте публичный и приватный ключи:

```
openssl req -new -newkey rsa:4096 -days 365 -x509 -subj "/CN=$KAFKA_SERVER"
-keyout ca-key -out ca-cert -nodes -passin pass:"$PASSWD" -passout pass:"$PASSWD"
```

#### 2. Создайте хранилище ключей для Kafka:

```
keytool -genkey -keyalg RSA -keystore server.keystore.jks -validity 365 -storepass
"$PASSWD" -keypass "$PASSWD" -dname "CN=$KAFKA_CLIENT" -storetype jks
```

#### 3. Создайте подпись:

```
keytool -keystore server.keystore.jks -certreq -file cert-file -storepass "$PASSWD" -
keypass "$PASSWD" -storetype jks
openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days 365
-CAcreateserial -passin pass:"$PASSWD"
```

#### 4. Поместите публичный сертификат для клиента в хранилище `client.truststore.p12`:

```
keytool -keystore client.truststore.p12 -alias CARoot -import -file ca-cert
-storepass "$PASSWD" -keypass "$PASSWD" -noprompt -storetype pkcs12
```

#### 5. Поместите сертификат и подпись в хранилище `server.keystore.jks`:

```
keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert -storepass
"$PASSWD" -keypass "$PASSWD" -noprompt -storetype jks
keytool -keystore server.keystore.jks -import -file cert-signed -storepass "$PASSWD" -
keypass "$PASSWD" -noprompt -storetype jks
```

#### 6. Скопируйте хранилище `client.truststore.p12` на узел с агентом.

#### 7. Задайте путь к хранилищу `client.truststore.p12` в MIB-параметре *TrustStorePKCS12* по пути `/CMP_Agent_v2.0/Kafka/Configuration/SSL/`.

#### 8. Задайте пароль от хранилища `client.truststore.p12` в MIB-параметре *TrustStorePKCS12Password* по пути `/CMP_Agent_v2.0/Kafka/Configuration/SSL/`.

## 15.2. Установка и настройка компонента logDecoder

Для запуска компонента logDecoder выполните команду:

```
./logdecoder brokerIP BAname PathToLogdecoders > /dev/null 2>&1 &
```

где:

- `brokerIP` — сетевое имя или IP-адрес и номер порта сервера системы Kafka;
- `BAname` — имя бизнес-приложения, логи которого обрабатывает система. Логи находятся в топике `log-BAname` системы Kafka;
- `PathToLogdecoders` — абсолютный путь к каталогу, где находятся справочники. Пример значения: `'/opt/Decoders/directories'`. Этот каталог должен содержать



символическую ссылку: `libdecoder_default.so -> /opt/Decoders/shared/libdecoder_default.so.2.3.`

Чтобы запустить компонент `logDecoder` для декодирования логов, которые находятся в топике `log-default`, выполните команду:

```
./logdecoder brokerIP default PathToLogdecoders > /dev/null 2>&1 &
```

## 15.3. Установка и настройка компонента `traceDecoder`

Для запуска компонента `traceDecoder` выполните команду:

```
./tracedecoder brokerIP BAname [log]
```

где:

- `brokerIP` — сетевое имя или IP-адрес и номер порта сервера системы Kafka;
- `BAname` — имя бизнес-приложения, трейсы которого обрабатывает система. Трейсы находятся в топике `trace-BAname` системы Kafka;
- `log` — признак создания файла с отладочной информацией о работе программы в рабочем каталоге. Необязательный параметр. Имя файла: `TraceDecoder_log.txt`.

Настройки компонента хранятся в конфигурационном файле формата JSON. Файл располагается в установочном каталоге компонента. В конфигурационном файле вы можете задать дополнительные параметры или настроить одновременную работу с несколькими профилями. Формат конфигурационного файла:

```
{
  "Broker" : {
    "address" : "<IP-sddress:Port>",
    "msgsize" : "<максимальный размер сообщения>",
    "log" : "true/false"
  },
  "Topics" : [
    {
      "baname" : "NAME1",
      "timeout" : "TIMEOUT1",
      "lazy" : "true/false",
      "restore" : "true/false",
      "recent-period" : "RECENT-PERIOD1"
    },
    {
      "baname" : "NAME2",
      "timeout" : "TIMEOUT2",
      "lazy" : "true/false",
      "restore" : "true/false",
      "recent-period" : "RECENT-PERIOD2"
    }
  ],
}
```

Конфигурационный файл содержит следующие блоки параметров:

- `Broker` — общие настройки работы с сервером Kafka:
  - `address` — сетевое имя или IP-адрес и номер порта сервера Kafka;
  - `msgsize` — максимальный размер сообщения, передаваемого в систему Kafka. Значение указывается в байтах. Значение по умолчанию: `` 16000000``;
  - `SASLProtocol` — протокол безопасности для связи с сервером Kafka. Значение по умолчанию: `'plaintext'`;

- SASLMechanism — процедура аутентификации. Значение по умолчанию: 'GSSAPI';
  - SASLUsername — имя пользователя;
  - SASLPassword — пароль пользователя;
  - SASLKerberosService — имя сервиса Kerberos. Значение по умолчанию: 'kafka';
  - SASLKerberosPrincipal — имя клиента Kerberos. Значение по умолчанию: 'kafkaclient';
  - SASLKerberosKeytab — путь к файлу трансляции. Дополнительный параметр;
  - SSLCaLocation — путь к файлу или каталогу с сертификатом;
  - SSLCertLocation — путь к открытому ключу;
  - SSLKeyLocation — путь к закрытому ключу;
  - SSLKeyPassword — пароль закрытого ключа;
  - log — признак создания файла с отладочной информацией о работе программы в рабочем каталоге. Формат имени файла: `TraceDecoder_log.txt`.
- Zipkin — настройки работы с сервером Kafka, в который выполнялась запись цепочек для использования в системе Zipkin:
    - enable — признак включения функции выдачи цепочек для использования в системе Zipkin. Значение по умолчанию: 'true';
    - address — сетевое имя или IP-адрес и номер порта сервера Kafka, в который выполняется запись цепочек для использования в системе Zipkin. Дополнительный параметр. По умолчанию используется сервер Kafka, который задан в параметре `Broker/address`;
    - msgsize — максимальный размер сообщения, который передается в систему Kafka для использования в системе Zipkin. Дополнительный параметр. По умолчанию используется значение, которое задано в параметре `Broker/msgsize`;
    - SASLProtocol — протокол безопасности для связи с сервером Kafka. Значение по умолчанию: 'plaintext';
    - SASLMechanism — процедура аутентификации. Значение по умолчанию: 'GSSAPI';
    - SASLUsername — имя пользователя;
    - SASLPassword — пароль пользователя;
    - SASLKerberosService — имя сервиса Kerberos. Значение по умолчанию: 'kafka';
    - SASLKerberosPrincipal — имя клиента Kerberos. Значение по умолчанию: 'kafkaclient';
    - SASLKerberosKeytab — путь к файлу трансляции. Дополнительный параметр;
    - SSLCaLocation — путь к файлу или каталогу с сертификатом;
    - SSLCertLocation — путь к открытому ключу;
    - SSLKeyLocation — путь к закрытому ключу;
    - SSLKeyPassword — пароль закрытого ключа.

Эта группа с дополнительными настройками.

- Filter — настройки функции фильтрации набора цепочек, которые передаются на серверы Kafka:
  - headers-only — признак включения режима записи цепочек, которые содержат заголовки трейсов без передаваемых полей данных. Значение по умолчанию: 'false';
  - fault-only — признак включения режима записи цепочек, которые содержат трейсы с признаком FAULT. Значение по умолчанию: 'true';
  - level-limit — максимальный уровень вложенности трейсовых сообщений в формируемых цепочках. Нумерация уровней трейсовых сообщений в цепочках

начинается с '0'. Значение по умолчанию: '-1' — записываются цепочки всех уровней вложенности трейсовых сообщений.

Эта группа с дополнительными настройками.

- Topics — настройки работы с топиками Kafka. Для каждого топика можно задать следующие параметры:
  - VName — имя бизнес-приложения, трейсы которого обрабатываются. Компонент обрабатывает трейсы, которые находятся в топике `trace-VName Kafka`;
  - partitions — список номеров секций топика Kafka, из которых выполняется чтение трейсовых сообщений. Дополнительный параметр;
  - timeout — тайм-аут обработки цепочки трейсов в миллисекундах. Значение по умолчанию: '5000';
  - lazy — признак включения режима отложенной обработки цепочки трейсов. Значение по умолчанию: false;
  - restore — признак включения восстановления значений времени запроса и ответа при отсутствии соответствующих трейсов. Значение по умолчанию: true;
  - recent-period — период блокирования повторной сборки цепочки трейсов, относящихся к одной сессии. Период указывается в секундах. Значение по умолчанию: '60';
  - format — специальный формат поступаемых на обработку трейсовых сообщений. Для трейсов типа AUDIT значение 'audit'. Необязательный параметр.

**Примечание.** Наивысший приоритет задан для параметров, указанных при запуске компонента.

Пример конфигурационного файла:

```
{
  "Broker" : {
    "address" : "198.51.100.11:8535",
    "msgsize" : "15000000",
    "SASLProtocol" : "sasl_plaintext",
    "SASLMechanism" : "PLAIN",
    "SASLUsername" : "user",
    "SASLPassword" : "password",
    "SASLKerberosService" : "kafka",
    "SASLKerberosPrincipal" : "kafkaclient",
    "SASLKerberosKeytab" : "keytab",
    "SSLCaLocation" : "caLocation",
    "SSLCertLocation" : "certLocation",
    "SSLKeyLocation" : "keyLocation",
    "SSLKeyPassword" : "keyPassword",
    "log" : "true"
  },
  "Zipkin" : {
    "enable" : "true",
    "address" : "198.51.100.15:8535",
    "msgsize" : "15000000",
    "SASLProtocol" : "sasl_plaintext",
    "SASLMechanism" : "PLAIN",
    "SASLUsername" : "user",
    "SASLPassword" : "password",
    "SASLKerberosService" : "kafka",
    "SASLKerberosPrincipal" : "kafkaclient",
    "SASLKerberosKeytab" : "keytab",
    "SSLCaLocation" : "caLocation",
    "SSLCertLocation" : "certLocation",
    "SSLKeyLocation" : "keyLocation",
    "SSLKeyPassword" : "keyPassword",
  },
  "Filter" : {
```

```

    "headers-only" : "true",
    "fault-only" : "false",
    "level-limit" : "0"
  },
  "Topics" : [
    {
      "baname" : "default",
      "timeout" : "10000"
    },
    {
      "baname" : "GetBalance",
      "partitions" : "0,1,2",
      "timeout" : "5000",
      "lazy" : "true",
      "restore" : "true",
      "recent-period" : "60"
      "format" : "trace"
    }
  ],
}

```

### 15.3.1. Постобработка цепочки трейсов

Тайм-аут (параметр *timeout*) отсчитывается с момента получения первого сообщения с конкретным идентификатором сессии. Условием завершения сбора цепочки трейсов является наличие принятых пар трейсов «запрос-ответ» по каждому уровню вызова.

Режим отложенной обработки цепочки трейсов задается значением параметра *lazy*:

- **False.** Выполняется активная проверка условия завершения сбора цепочки трейсов с минимальной периодичностью опроса сервера Kafka. После завершения процесса сбора цепочки трейсов выполняется ее обработка и запись исходящих сообщений в [ТОПИК](#) `chain_BAname` сервера Kafka.
- **True.** Активная проверка условия завершения сбора цепочки трейсов не выполняется до окончания периода, заданного параметром *timeout*. Такой режим работы позволяет избежать преждевременного завершения сбора цепочки с обработкой и записью исходящих сообщений — до завершения других соседних цепочек трейсов корневого уровня данной сессии, если они присутствуют.

Режим построения цепочки трейсов и ее дальнейшая обработка определяется значением параметра *restore*:

- **True.** С восстановлением недостающей информации о времени вызовов. Построение цепочки трейсов может выполняться при отсутствии времени вызова и ответа. Это возможно при отсутствии соответствующих трейсов. В этом случае цепочка строится на основе данных, полученных из полей `invokeChain`, `LocalIP` и `RemoteIP`. Недостающие значения времени вызовов рассчитываются аналитически и подставляются в поля исходящего сообщения. Если трейсовые сообщения поступают после окончания тайм-аута (параметр *timeout*), вы можете задать период с момента отправки первого сообщения по конкретной сессии, в течение которого обработка цепочки и отправка последующих исходящих сообщений по этой же сессии будут заблокированы. Это нужно, чтобы избежать в исходящих сообщениях многократного повторения восстановленных участков цепочки по одной и той же сессии в соответствии со значением параметра *recent-period*.
- **False.** Без восстановления недостающей информации о времени вызовов. Если к окончанию тайм-аута условие завершения сбора цепочки не соблюдено, система удаляет эту цепочку и не передает исходящие сообщения в топик `chain_BAname`.

В процессе обработки построенной цепочки трейсов трейсовые сообщения сортируются в порядке их вызова с присвоением им соответствующих уровня вызова

(level) и порядкового номера в последовательности вызовов (sequenceNumber). Для каждого сообщения рассчитывается длительность вызова в миллисекундах (delay).

Если в сообщении включено поле данных (payload), система перемещает его в исходящие сообщения. При этом, если поле данных закодировано в формате FastInfoset, система декодирует данные перед записью исходящего сообщения.

Система добавляет полученную информацию об окружении сессии (Environment) в виде соответствующей группы параметров в каждое исходящее сообщение — в топик chain\_BAname сервера Kafka.

Если размер исходящего сообщения превышает максимальный размер, заданный в параметре *msgsize*, система анализирует значение поля payload исходящего сообщения. Если поле payload исходящего сообщения позволяет уменьшить размер поля payload исходящего сообщения, чтобы полный размер исходящего сообщения не превышал значения, заданного параметром *msgsize* — система уменьшает это значение.

## 15.4. Установка и настройка компонента logstash

Вы можете установить компонента двумя способами:

1. с помощью инструкций на сайте разработчика: <https://www.elastic.co/guide/en/logstash/current/setup-logstash.html>;
2. в Docker-контейнере из репозитория Docker-образов: [https://hub.docker.com/\\_/logstash](https://hub.docker.com/_/logstash).

Для настройки компонента выполните следующие действия:

1. Задайте адрес сервера системы Kafka в файлах конфигурации системы OMPostprocess:
  - chain.conf
  - decoderlog.conf
  - log.conf
  - trace.conf
  - tracedefault.conf
  - logdefault.conf
  - pipelines.yml.
2. Сохраните их в конфигурационном каталоге /usr/share/logstash/config.

### Пример настройки файла decoderlog.conf

```
input {
  kafka {
    bootstrap_servers => "kafka:9092" # например, 192.0.2.11:9092
                                topics_pattern => "decodedlog-.*"
    # подписка на все топики, содержащие декодированные с помощью компонента
    logdecoder сообщения.
    # например, при запуске 2 экземпляров компонента logdecoder: 1) с BAName GetBalance 2)
    default, в индекс decodedlog-2021.03.17 попадут сообщения из топиков decodedlog-default и
    decodedlog-GetBalance
    metadata_max_age_ms => 1 #данная опция необходима для мгновенного объединения групп.
    Решает проблему пропуска первых сообщений при разворачивании тестового окружения в
    docker контейнерах.
    decorate_events => true #опция необходима для добавлении метаинформации от kafka (topic,
    offset, e.t.c). Добавление метаинформации от kafka - не обязательно.
```

```

        auto_offset_reset => earliest #решает проблему нестабильности
тестового окружения в докер контейнерах. После решения https://jira.bercut.com/browse/
PLV-1662 - можно убрать.
    }
}

filter {
  json{
    source => "message"
  }
  date {
    match => ["timestamp", "yyyy.MM.dd HH:mm:ss.SSS"]
    target => "@timestamp"
    remove_field => "timestamp" #для корректного отображения информации в
grafana, необходимо удалить timestamp, сгенерированный kafka и использовать
timestamp из декодированного сообщения.
  }
  mutate {
    copy => { "[@metadata][kafka]" => "kafka" } # добавляет метаинформацию от kafka в сообщение
(не обязательно)
  }
}

output {
  elasticsearch {
    hosts => ["elasticsearch:9200"] # ip:порт elastic
    index => "decodedlog-%{+YYYY.MM.dd}" # название index-а в elastic
    ilm_enabled => false # опция необходима для старта oss(open source) docker контейнера с
logstash. Так как ilm - не является open source компонентом.
  }
}

```

ATLAS Agent отправляет сообщения, которые не относятся к определенному бизнес-приложению, в топики: trace-default и log-default. Для хранения большого количества сообщений в log-default используйте конфигурацию:

```

input {
  kafka {
    bootstrap_servers => "kafka:9092"
    topics_pattern => "log-default.*"
    metadata_max_age_ms => 1
    auto_offset_reset => earliest
  }
}

filter {
}

output {
  null{}
}

```

## 15.5. Установка и настройка компонента Zookeeper

Компонент Zookeeper является централизованным хранилищем YAML-описаний бизнес-приложений и конфигураций компонентов.

Компонент Zookeeper хранит метаинформацию для функционирования автоконфигурации компонентов и маршрутов на шине RTSIB. К Zookeeper подключаются ATLAS Agent, которые входят в группу серверов бизнес-приложения.

YAML-описание создает разработчик системы.

Задайте параметры соединения с компонентом Zookeeper в MIB-группе CMP\_Agent\_v2.0/Discovery:

- **Enable.** Признак включения работы с компонентом Zookeeper.

- *ZooKeeperAddr*. IP-адрес и номер порта сервера, на котором установлен компонент Zookeeper.

При запуске ATLAS Agent считывает дерево BusinessApplications в компоненте Zookeeper. После считывания выполняется разбор YAML-описаний. ATLAS Agent сохраняет в своей памяти модель бизнес-приложения.

Структура дерева BusinessApplications:

```
BusinessApplications/<имя бизнес-приложения>/Components/<имя компонента>
```

YAML-описание развертывания бизнес-приложения хранится в переменной *Description* ветки BusinessApplications/<имя бизнес-приложения> для каждого бизнес-приложения. YAML-описание содержит список локальных для каждого хоста имен систем SLES, LW SA Container и Service Gateway, которые заданы внешней системой при развертывании бизнес-приложения.

YAML-описание компонента (uses\provides) хранится в переменной *Description* ветки BusinessApplications/<имя бизнес-приложения>/Components/<имя компонента> для каждого компонента.

ATLAS Agent подписывается на изменение ветки BusinessApplications в компоненте Zookeeper. При удалении бизнес-приложения никаких действий не выполняется. При добавлении или изменении бизнес-приложения выполняются действия, аналогичные тем, которые выполняются при запуске ATLAS Agent.

### 15.5.1. Подписка на добавление, изменение и удаление portType

После старта и получения конфигурации Business Application ATLAS Agent подписывается на изменение MIB-группы Core/Statistics/DeployedArtifacts для всех приложений, которые есть в YAML-описании Business Application.

При добавлении, изменении или удалении вложенной группы в Core/Statistics/DeployedArtifacts ATLAS Agent создает, изменяет или удаляет в Apache ZooKeeper ветку соответствующего интерфейса: domains/<имя домена из yaml-описания>/<название артефакта>/<название интерфейса (portType)>. Название интерфейса извлекается из MIB-группы Core/Statistics/DeployedArtifacts/<название артефакта>/ServiceComponents/<имя компонента>/ImplementedPortTypes/<название интерфейса (portType)>.

Имя домена хранится в секции deployment\_descriptor BusinessApplications/<название артефакта>/Description в yaml-описании.

Создайте в ветке domains/<имя домена из yaml-описания>/<название артефакта>/<название интерфейса (portType)> переменную с именем <название приложения>\_<host> из JSON. Например: 'CMP\_SLES\_v3\_4\_192\_0\_2\_1'.

В YAML-описании deployment\_descriptor BusinessApplications/<название артефакта>/Description в секции use/filter/routes хранится значение **transportType** соответствующего portType.

Значение переменной при создании или изменении ветки содержит JSON-код формата:

```
{"app": "<название приложения>", "host": "<имя хоста>", "transport": Object }
```

Структура элемента Object зависит от типа транспорта:

- тип 'tcp'. Структура элемента Object:

```
Object = {
```

```

    "tcp": {
      "dataServer": "192.168.0.1: 3021",
      "serviceServer": "192.168.0.1: 3022",
      "connectionsCount": 10
    }
  }
}

```

Параметры для DataServer и ServiceServer берутся из MIB-групп @libs/MessageBus/DataServer/ProtocolLayers/Tcp/ServerAddress и @libs/MessageBus/ServiceServer/ProtocolLayers/Tcp/ServerAddress.

- тип 'Local'. Структура элемента Object:

```
Object = {"local":{}}
```

- тип 'Http'. Структура элемента Object:

```

Object = {
  "http": {
    "httpURI": "192.168.0.1: 8080",
    "connectionsCount": 10
  }
}

```

### 15.5.2. Конфигурация маршрутов на шине RTSIB

При добавлении или изменении интерфейса и дерева в Apache ZooKeeper ATLAS Agent выполняет следующие действия:

1. Проверяет по внутренней модели, используется ли интерфейс в секции uses в рамках бизнес-приложения.
2. Считывает JSON-код узла, в котором есть информация об организации маршрута к этому интерфейсу.

### 15.5.3. Соединения

Создание соединений зависит от типа транспорта:

- тип 'Tcp'. ATLAS Agent создает или пересоздает в MIB-группе @libs/MessageBus/ExternalConnections каждого приложения группу с именем <название приложения>\_<host> из JSON. Например: 'CMP\_SLES\_v3\_4\_192\_0\_2\_1'. Далее ATLAS Agent создает соответствующие соединения. Параметры для DataServer и ServiceServer берутся из JSON.
- тип 'Local'. Соединения не создаются.
- тип 'Http'. ATLAS Agent создает или пересоздает в MIB-группе @libs/MessageBus/ExternalConnections каждого приложения группу с именем <название приложения>\_<host> из JSON. Далее ATLAS Agent создает соответствующие соединения. Параметры для HttpURI берутся из JSON.

### 15.5.4. Маршруты

Создание маршрутов зависит от типа транспорта:

- тип 'Tcp':
  - ATLAS Agent создает или пересоздает в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType> каждого приложения дерево MIB-настроек;



- ATLAS Agent создает в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable вложенную группу с именем, которое было задано в группе @libs/MessageBus/ExternalConnections.
- в созданной группе задаются значения MIB-параметров:
  - *ConnectionName*=<Имя созданной группы>.
  - *ConnectionType*='tcp'.
- тип 'Local':
  - ATLAS Agent создает или пересоздает в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType> каждого приложения дерево MIB-настроек;
  - ATLAS Agent создает в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable вложенную группу с именем 'local'.
  - в созданной группе задаются значения MIB-параметров:
    - *ConnectionName*='local'.
    - *ConnectionType*='local'.
- тип 'Http':
  - ATLAS Agent создает или пересоздает в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType> каждого приложения дерево MIB-настроек;
  - ATLAS Agent создает в MIB-группе @libs/MessageBus/Core/Configuration/Routing/PortTypeRouting/<имя portType>/ScalingTable вложенную группу с именем, которое было задано в группе @libs/MessageBus/HttpExternalConnections.
  - в созданной группе задаются значения MIB-параметров:
    - *ConnectionName*=<Имя созданной группы>.
    - *ConnectionType*='http'.

Для создания маршрута с условием используйте секцию `filters` в YAML-описании.

### 15.5.5. Пример YAML-описания

```
{
# -- service deployment
# -- contains setting for get-balance service package
# -- this package consists of BP_GetBalance, SaS_carem, BP_CareMData components

# -- section [use] describes external interface usage

"//": "BP_GetBalance_v3",
"//": "GetBalance.bpel",
"//": "[provide]",
"//": "{http://www.bercut.com/wsd/GetBalance}GetBalancePortType",
"//": "[use]",
"//": "getSubscriberPromoText:: {http://www.bercut.com/cc/sa/advertisingmanagement}
AdvertisingManagement",
"//": "getSubscriberBalance:: {http://www.bercut.com/cc/sa/balancemanagement}
BalanceManagement",
"//": "getContentMessage:: {http://www.bercut.com/cc/sa/contentmanagement}
ContentManagement",
"//": "roundDecimal:: {http://www.bercut.com/specs/common/Utils}Utils",
"//": "GetSubscriberInfo:: {http://www.bercut.com/wsd/SubscriberAndServiceInfo}
SubscriberAndServiceInfoPortType",
```



```

{
  "interface": "{http://www.bercut.com/cc/sa/accountmanagement}AccountManagement",
  "filters": [
    {
      "name": "carem-data-db",
      "criteria": "*",
      "operations": [
        "getSubscriber"
      ],
      "routes": [
        {
          "name" : "invoice",
          "Protocol": "rtsib",
          "Algo": "round-robbin/response-weighted/any",
          "Domain": [
            "ssp.cache.com"
          ],
          "Connections": 5
        }
      ]
    }
  ],
  {
    "interface": "{http://www.bercut.com/cc/sa/contentmanagement}ContentManagement",
    "filters": [
      {
        "name": "invoice",
        "operations": [
          "getContentMessage"
        ],
        "routes": [
          {
            "name" : "cache",
            "Protocol": "rtsib",

            "Algo": "round-robbin/response-weighted/any",

            "Domain": [
              "ssp.cache.com"
            ],
            "Connections": 5
          }
        ]
      }
    ],
    {
      "interface": "{http://www.bercut.com/cc/sa/balancemanagement}BalanceManagement",
      "filters": [
        {
          "name": "invoice3",
          "operations": [
            "getSubscriberBalance"
          ],
          "routes": [
            {
              "name" : "cache",
              "Protocol": "rtsib",

              "Algo": "round-robbin/response-weighted/any",

              "Domain": [
                "ssp.cache.com"
              ],
              "Connections": 5
            }
          ]
        }
      ]
    }
  ]
}

```

## 16. Платформа Tarantool

*Tarantool — платформа резидентных вычислений (in-memory computing) с гибкой схемой данных для эффективного создания высоконагруженных приложений.*

В Platform v3 реализованы следующие компоненты:

- Tarantool RTSIB Driver;
- Tarantool SOAP Adapter.

Компоненты предназначены для технических специалистов, которые занимаются разработкой, тестированием и сопровождением систем Bercut.

### 16.1. Tarantool RTSIB Driver

*Tarantool RTSIB Driver является расширением сервера приложений Tarantool и организует взаимодействие Lua-приложений с интеграционной шиной RTSIB.*

Tarantool RTSIB Driver может работать в следующих режимах:

- В режиме сервера. Обеспечивает прием входящих запросов на portType;
- В режиме клиента. Обеспечивает отправку сообщений на удаленный серверный portType.

#### **Обработка исходящего запроса на стороне клиента**

Сценарий состоит из следующих шагов:

1. Диспетчер потоков Tarantool (*fiber*) вызывает функцию `process_events()`. Эта функция обслуживает поток входящих сообщений и событий интеграционной шины. Fiber ожидает наступления события в системном вызове `coio_wait()`.
2. Пользовательская программа на fiber вызывает функцию `client:invoke()`, в которой передает таблицу с заполненным ответным сообщением. Формат сообщения совпадает с форматом операции удаленного portType.
3. Библиотека преобразовывает таблицы Lua к типу `unidt::UDT_Type` и передает ответ в шину. Для этого используется функция `SendRequest()` библиотеки `luartsib`. После преобразования библиотека переходит в режим ожидания по условной переменной `fiber`.
4. При получении входящего сообщения из потока `librtsib` система вызывает функцию-обработчик `OnResponse()`. Функция ставит входное сообщение в очередь и формирует событие с помощью системного вызова `eventfd_write()`.
5. Ядро Tarantool инициирует fiber. Fiber выбирает из очереди входящий ответ для клиента и находит условную переменную, которая соответствует отправленному ранее запросу. Fiber диспетчера отправляет сигнальное сообщение ожидающему `fiber`.
6. Ядро Tarantool инициирует fiber, который находился в режиме ожидания. Этот fiber выбирает ответ из очереди, преобразовывает его из внутреннего типа `unidt::UDT_Type` в таблицу Lua и возвращает управление пользовательской программе.
7. Пользовательская программа получает ответ сервера. Ответ возвращается в значении функции `client:invoke()`.

## Обработка входящего запроса на стороне сервера

Сценарий состоит из следующих шагов:

1. Fiber периодически вызывает функцию `poll_requests()`. Эта функция проверяет наличие входящих сообщений.
2. При получении входящего сообщения из сети система вызывает функцию-обработчик `OnRequest()` из потока `librtsib`. Функция ставит входное сообщение в очередь и формирует событие с помощью системного вызова `eventfd_write()`.
3. Ядро Tarantool иницирует fiber. Fiber выбирает из очереди все накопившиеся сообщения от удаленного клиента и возвращает их в Lua. Диспетчер запускает для каждого сообщения свой fiber.
4. Fiber обработчика сообщений вызывает функцию `request:get()`. Функция преобразовывает входящее сообщение из внутреннего типа `unidt::UDT_Type` в таблицу Lua.
5. Код пользовательской программы предоставляет функцию-обработчик входящего запроса. Функция разбирает сообщение и возвращает ответное сообщение с помощью `request:send_response()`.
6. Fiber обработчика сообщений преобразовывает таблицы Lua в тип `unidt::UDT_Type` и отправляет ответ в шину. Ответ отправляется с помощью функции `SendResponse()` библиотеки `luartsib`.

## Доступ к HTTP Headers

1. Система записывает XSD-описание для передаваемых заголовков и добавляет его в репозиторий. Пример XSD-описания:

```
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bercut.com/schema/http-headers">
  <xs:element name="SomeHeader1" type="xs:string"/>
  <xs:element name="SomeHeader2" type="xs:string"/>
</xs:schema>
```

2. Система разбирает SOAP Header (sequence-of-any) с помощью XSD-описания из репозитория: ищет элементы по `qName`.
3. Найденные элементы добавляются в таблицу `LuaTable` SOAP-сообщения.

Скрипт lua имеет доступ к SOAP Header.

RTSIB SDK выполняет следующие функции при разборе SOAP Header:

1. Помещает в последовательность `sequence-of-any` неизвестные поля SOAP Header.
2. Перемещает заголовки для доступа к ним.

## Пример

Скрипт lua для доступа к HTTP Header при распечатке `LuaTable`-запроса в функции-обработчике fiber для серверного `portType` `{http://www.bercut.com/wsdl/GetPersonalRate}GetPersonalRatePortType` и операции `GetPersonalRate`:

```
local onCreateGetPersonalRatePortTypeFiber = function(GetPersonalRatePortTypeReqMsg)

  function dump(o)
    if type(o) == 'table' then
      local s = '{ '
      for k,v in pairs(o) do
        if type(k) ~= 'number' then k = "'..k..'" end
        s = s .. '['..k..'] = ' .. dump(v) .. ','
      end
      return s .. '}'
    else
      return tostring(o)
    end
  end

end
```

```

        print("%s", dump(GetPersonalRatePortTypeReqMsg))
    end

    { ["data"] = { ["GetPersonalRate"] = { [""] = { ["parameters"] = { ["msisdn"] = ?, } , } , } ,
    ["extensions"] = { ["From"] =
    { ["Address"] = rtsib://203.0.113.5:3116/6b4a3960-6557-11ed-8ebd-005056b784fe, } , ["To"] =
    rtsib://198.51.100.11:5016, ["PrincipalContext"] =
    { ["UserPrincipal"] = SA_KafkaConsumer, ["SystemPrincipal"] = LWSA_Kafka, ["@target_ns"] =
    http://someHeaderScheme.url, ["headerElement"] =
    { ["header1"] = headerValue, } , } , }

```

### 16.1.1. Установка Tarantool RTSIB Driver

Порядок установки:

1. Перед запуском Tarantool в каталоге с файлами разделяемых библиотек выполните следующие команды:

```

$ ln -sf librtsib.so.3.0 librtsib.so
$ ln -sf libluartsib.so.1.0 rtsib.so

```

2. Создайте в MIB дерево настроек для интеграционной шины. Имя приложения должно совпадать с параметром **appName**, который передается функции-конструктору `rtsib.instance.new(appName)`. Пример: `appName = 'Tarantool'`, MIB-дерево — `'CMP_MessageBusApp_v3.0_Tarantool'`. При использовании функции-конструктора `rtsib.instance.new(appName, parameters)` дополнительные параметры, например, версия и имя приложения, сохраняются в MIB-параметрах в группе `Tarantool/About`. Пример сервера на базе Tarantool SOAP Adapter с использованием единого callback и дополнительными MIB-параметрами приведен в разделе — [16.2.2](#).

### 16.1.2. Интерфейс для Lua

**Таблица 7. Таблица rtsib**

Имя функции или объекта	Аргументы	Возвращаемое значение	Описание
<code>poll_events()</code>	нет	<code>event[]</code>	<p>Функция обслуживания клиентских событий на интеграционной шине. Возвращает массив событий. Массив может быть пустым, если событий нет.</p> <p>Функция периодически вызывается в fiber Tarantool.</p> <p>Пример:</p> <pre> while true do     fiber.testcancel()     local events =     rtsib.poll_events()     for i=1,#events do         local event = events[i]         event:process()     end     fiber.yield() end </pre>
<code>poll_requests()</code>	нет	<code>request[]</code>	<p>Функция опроса (pooling) сервера на предмет наличия входящих запросов. Возвращает массив</p>

Имя функции или объекта	Аргументы	Возвращаемое значение	Описание
			сообщений. Массив может быть пустым, если сообщений нет. Функция периодически вызывается в fiber Tarantool. Пример: <pre>while true do   fiber.testcancel()   local msgs =   rtsib.poll_requests()   for i=1,#msgs do     fiber.create(function(msg)       OrderFiber(msg:get())     end, msgs[i])   end   fiber.yield() end</pre>
instance			Таблица соответствует экземпляру приложения.
server			Таблица соответствует экземпляру сервера.
client			Таблица соответствует экземпляру клиента.
request			Таблица соответствует входящему сообщению — запросу на серверный portType.
event			Таблица соответствует событию клиента. Допустимые события: <ul style="list-style-type: none"> <li>■ получение ответа на исходящий запрос;</li> <li>■ ошибка исходящего запроса;</li> <li>■ тайм-аут ожидания ответа на исходящий запрос;</li> <li>■ доступность portType;</li> <li>■ недоступность portType;</li> <li>■ перегрузка очереди сообщений;</li> <li>■ снятие перегрузки очереди сообщений.</li> </ul>

**Таблица 8. Таблица instance**

Имя функции	Аргументы	Возвращаемое значение	Описание
new	appName: string	instance: table	Создание экземпляра приложения для работы с интеграционной шиной RTSIB. Во входных параметрах передается имя приложения для подключения к MIB-настройкам. В MIB-дереве

Имя функции	Аргументы	Возвращаемое значение	Описание
			должна присутствовать группа: CMP_MessageBusApp_v3.0_appName.
new	appName: string, parameters: table	instance: table	Создание экземпляра приложения для работы с интеграционной шиной RTSIB с дополнительными параметрами. Во входных параметрах передается имя приложения для подключения к MIB-настройкам и таблица с дополнительными переменными для их отображения в MIB-группе appName/About . В MIB-дереве должна присутствовать группа: CMP_MessageBusApp_v3.0_appName.
start	нет	result: boolean	Инициализация экземпляра приложения, подключение к MIB. При успешном завершении возвращается значение 'true'.
create_server	portType: string	result: table	Создание экземпляра сервера для обслуживания portType из WSDL-описания сервиса.
create_client	portType: string	result: table	Создание экземпляра клиента для сетевого подключения к portType из WSDL-описания сервиса.
stop	нет	нет	Останов экземпляра приложения, отключение от MIB, освобождение ресурсов.

Таблица 9. Таблица server

Имя функции	Аргументы	Возвращаемое значение	Описание
new	portType: string	instance: table	Создание экземпляра сервера для обслуживания portType из WSDL-описанием сервиса.  Первый входной аргумент функции — имя обслуживаемого portType. Путь и название соответствующего WSDL-описания должны быть заданы в настройках шины, в MIB-группе CMP_MessageBusApp_v3.0_Tarantool/@libs/LIB_WsdRepository_v1.0/Core/Configuration/SoapServices/<название продукта>/FilePath.
start	нет	result: boolean	Запуск сервиса для прослушивания SOAP/TCP-порта и приема сообщений.



Имя функции	Аргументы	Возвращаемое значение	Описание
			При успешном завершении возвращается значение 'true'.
stop	нет	нет	Останов экземпляра сервера, закрытие соединений и TCP-порта, освобождение ресурсов.

**Таблица 10. Таблица client**

Имя функции	Аргументы	Возвращаемое значение	Описание
new	portType: string	instance: table	Создание экземпляра клиента для сетевого подключения к portType из WSDL-описания сервиса. На вход функции передается имя обслуживаемого portType. Путь и название соответствующего WSDL-описания должны быть заданы в настройках шины, в MIB-группе CMP_MessageBusApp_v3.0_Tarantool/@libs/LIB_WsdlRepository_v1.0/Core/Configuration/SoapServices/<название продукта>/FilePath.
start	нет	result: boolean	Запуск клиентского сервиса для подключения к удаленному portType. При успешном выполнении возвращается значение 'true'.
invoke	request: table	response: table	Синхронная отправка запроса на удаленный portType. На время ожидания ответа ядро Tarantool передает управление ожидающим fiber. Функция принимает таблицу, которая соответствует запросу. Функция возвращает таблицу, которая соответствует ответному сообщению. Если отправка сообщения невозможна, возвращается значение 'nil'.
stop	нет	нет	Останов экземпляра сервера, закрытие соединений и TCP-порта, освобождение ресурсов.

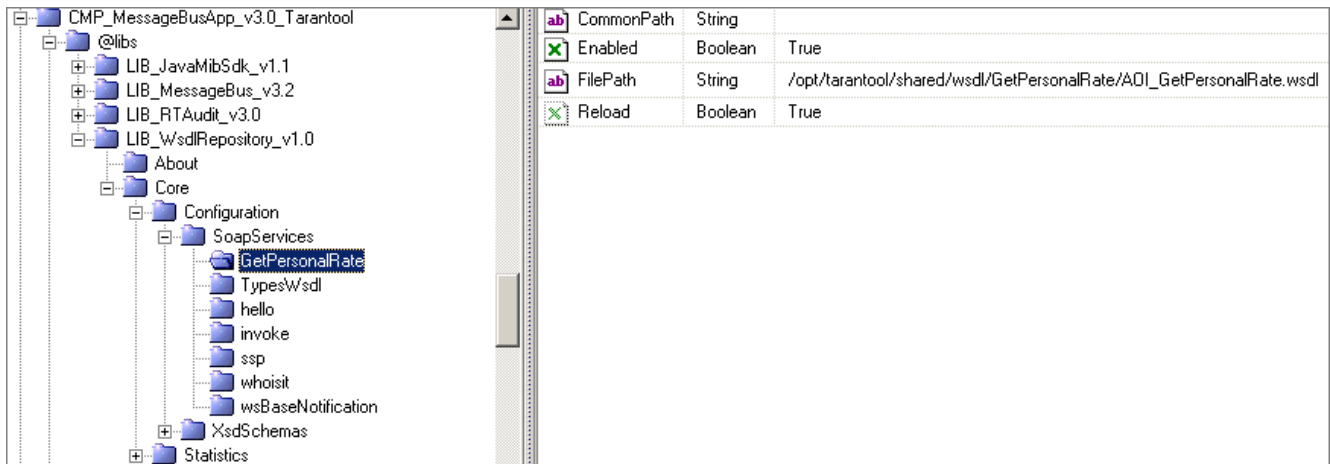
**Таблица 11. Таблица request**

Имя функции	Аргументы	Возвращаемое значение	Описание
get	нет	msg: table	Возвращает таблицу, которая соответствует принятому запросу.

Имя функции	Аргументы	Возвращаемое значение	Описание
send_response	msg: table	нет	Отправка ответа на запрос.

**Таблица 12. Таблица event**

Имя функции или свойства	Аргументы	Возвращаемое значение	Описание
process	нет	нет	Вызов обработчика, который зарегистрирован для данного типа события. Код обработчика заранее зарегистрирован в библиотеке.
eventType	—	number	Получение идентификатора, соответствующего типу события.
portType	—	string	Получение portType, для которого произошло событие.



**Рис. 171. Пример настройки WSDL-описания**

### 16.1.3. Примеры использования Tarantool RTSIB Driver

#### Обслуживание серверного portType

rtsib\_simple\_server.lua

```
#!/usr/bin/env tarantool

data_dir = "server"

box.cfg({
  log_level = 5
  , listen = 3131
  -- , log = 'rtsib_simple_server.txt'
  , wal_dir = data_dir
  , memtx_dir = data_dir
  , vinyl_dir = data_dir
})

box.once('init', function()
  box.schema.user.grant("guest", 'read,write,execute,create,drop', 'universe')
end)

log = require 'log'
fiber = require 'fiber'
```

```

rtsib = require 'rtsib'

orderNum = 0
appName = 'OrderManagement'
portType = '{http://www.bercut.com/spec/wsdl/ProductOrdering}ProductOrdering'

local onCreateProductOrder = function(productOrderingReqMsg)
    orderNum = orderNum + 1
    local productOrderingRespMsg = {
        data = {
            createProductOrder = {
                createProductOrderResponse = {
                    param = {
                        orderId = "PORD-TRNL-" .. tostring(orderNum)
                    }
                }
            }
        }
    }
    return productOrderingRespMsg
end

--- функция обслуживает входящий поток сообщений, создавая фибер для каждого
отдельного сообщения
function serverPollRequests()
    fiber.self():name('rtsib_poll_messages_f')
    log.info("run")

    while true do
        fiber.testcancel()
        log.debug("poll requests")
        local msgs = rtsib.poll_requests()
        for i=1,#msgs do
            fiber.create(function(msg)
                msg:send_response(onCreateProductOrder(msg:get()))
            end, msgs[i])
        end
        fiber.yield()
    end

    log.info("stop")
end

instance = rtsib.instance.new(appName)
if not instance:start() then
    os.exit()
end
log.info(tostring(instance)..": start")

server = rtsib.server.new(portType)
if not server:start() then
    os.exit()
end
log.info(tostring(server)..": start")

disp_fiber = fiber.create(serverPollRequests)

require('console').start()

server:stop()
instance:stop()

os.execute('sleep 1')
os.exit()

```

## Отправка запроса на клиентский portType

### rtsib\_simple\_client.lua

```

#!/usr/bin/env tarantool

data_dir = "client"

```

```

box.cfg({
  log_level    = 5
  , log        = 'rtsib_simple_client.txt'
  , wal_dir    = data_dir
  , memtx_dir  = data_dir
  , vinyl_dir  = data_dir
})

box.once('init', function()
  box.schema.user.grant("guest", 'read,write,execute,create,drop', 'universe')
end)

fiber = require 'fiber'
rtsib = require 'rtsib'

appName = 'TarantoolTestClient'
portType = '{http://www.bercut.com/spec/wsdl/ProductOrdering}ProductOrdering'

instance = rtsib.instance.new(appName)
rc = instance:start()
assert(rc == 0, "rtsib instance can not started")

function pollClientEvents(handlers)
  fiber.self():name('rtsib_poll_events_f')
  fiber.self():set_joinable(true)

  log.info("run")

  while true do
    fiber.testcancel()
    log.debug("poll events")
    local events = rtsib.poll_events()
    for i=1,#events do
      local event = events[i]
      log.debug("process " .. tostring(event))
      event:process()
      local handler = handlers[event.eventType]
      if handler then
        handler(event.portType)
      end
    end
    fiber.yield()
  end

  log.info("stop")
end

client = rtsib.client.new(portType)
handlers = {}
client_fiber = fiber.create(pollClientEvents, handlers)
if not client:start() then
  os.exit()
end

productOrderingReqMsg = require('ProductOrdering').createProductOrderRequest
productOrderingRespMsg = client:invoke(productOrderingReqMsg)
if productOrderingResponse then
  print(string.format("result: %s",
    productOrderingResponse.data.createProductOrder.createProductOrderResponse.param.orderId))
else
  print("Error happens")
end

os.execute('sleep 1')
client:stop()
instance:stop()

client_fiber:cancel()
client_fiber:join()

os.exit()

```

## 16.2. Tarantool SOAP Adapter

*Tarantool SOAP Adapter* является расширением сервера приложений *Tarantool* и организует взаимодействие Lua-приложений с интеграционной шиной *RTSIB* с помощью *SOAP*-сообщений.

Tarantool SOAP Adapter предоставляет следующие интерфейсы:

- **SOAPListener**. Интерфейс сервера для приема входящих запросов на локальный portType;
- **SOAPConnector**. Интерфейс клиента для отправки запросов на сетевой portType.

Tarantool SOAP Adapter реализован как надстройка над библиотекой Tarantool RTSIB Driver.

### 16.2.1. SOAP-интерфейсы

#### **SOAPInstance**

Интерфейс экземпляра адаптера.

Имя операции	Аргументы	Возвращаемое значение	Описание
SOAPInstance()	appName: string	SOAPInstance: table	Создание экземпляра адаптера с указанным именем. Допускается создание одного экземпляра в рамках одного экземпляра приложения Tarantool. Возвращает созданный объект.
SOAPInstanceEx()	appName: string parameters: table	SOAPInstance: table	Создание экземпляра адаптера с указанным именем и таблицы с дополнительными параметрами в MIB-группе /appName/About. Возвращает созданный объект. <b>Пример:</b> Из lua передаются параметры: parameters = { Description = 'SomeService', Version = '1.2.3.100', SomeInfo = 'Any info' } В MIB-дереве создаются параметры: /appName/About/Description = 'SomeService' /appName/About/Version = '1.2.3.100'

Имя операции	Аргументы	Возвращаемое значение	Описание
			<p>/appName/About/SomeInfo = 'Any info'</p> <p>Допускается создание одного экземпляра в рамках одного приложения Tarantool.</p>
openListener()	portType: string onRequestCallback: function	SOAPListener: table	<p>Создание нового или получение существующего экземпляра серверной точки доступа SOAP.</p> <p>Аргумент portType задает имя локального portType, который задан в настройках интеграционной шины.</p> <p>Аргумент onRequestCallback задает функцию с сигнатурой: table onRequestCallback (table). Функция вызывается в отдельном fiber при поступлении входящего запроса. Функция получает на вход таблицу, которая соответствует входящему запросу. Функция возвращает ответную таблицей, структура которой соответствует структуре выбранного протокола.</p> <p>Функция возвращает объект SOAPListener.</p>
openMultiple Listener()	portType: string onRequestCallback: function	SOAPListener: table	<p>Создание нового или получение существующего экземпляра серверной точки доступа SOAP. для получения в onRequestCallback значения portType дополнительным аргументом.</p> <p>Для нескольких серверных интерфейсов можно использовать единую callBack-функцию.</p> <p>Аргумент portType задает имя локального portType, который должен быть задан в MIB-настройках RTSIB.</p> <p>Аргумент onRequestCallback задает функцию со следующей сигнатурой: table onRequestCallback (table). Эта функция вызывается в отдельном fiber при поступлении входящего запроса. Функция получает на вход таблицу, которая соответствует входящему запросу. Функция возвращает</p>

Имя операции	Аргументы	Возвращаемое значение	Описание
			ответную таблицу, структура которой соответствует структуре выбранного протокола. Функция возвращает объект SOAPListener.
openConnector()	portType	SOAPConnector: table	Создание нового или получение существующего экземпляра клиентской точки доступа SOAP. Аргумент portType задает имя сетевого portType, который задан в настройках интеграционной шины. Функция возвращает объект SOAPConnector.
close()	нет	нет	Закрытие всех ранее созданных точек доступа, остановка служебных fiber и отключение приложения в системе ATLAS.

### SOAPListener

Интерфейс серверной точки доступа.

Имя операции	Аргументы	Возвращаемое значение	Описание
close()	нет	нет	Прекращение приема сообщений для portType.

### SOAPConnector

Интерфейс клиентской точки доступа.

Имя операции	Аргументы	Возвращаемое значение	Описание
invoke	request: table	response: table	Отправка SOAP-запроса и ожидание получения ответа от сетевого portType. Функция выполняется асинхронно с ожиданием вызывающего fiber на условной переменной.
close	нет	нет	Отключение от сетевого portType.

## 16.2.2. Примеры использования Tarantool SOAP Adapter

### Сервер на базе Tarantool SOAP Adapter

#### server.lua

```

---
--- Generated by EmmyLua (https://github.com/EmmyLua)
--- Created by bercut.

```

```

--- DateTime: 19.12.2020 10:08
---
data_dir = "server"
box.cfg({
  log_level = 5
  , listen = 3131
--  , log = 'server.txt'
  , wal_dir = data_dir
  , memtx_dir = data_dir
  , vinyl_dir = data_dir
})
box.once('init', function()
  box.schema.user.grant("guest", 'read,write,execute,create,drop', 'universe')
end)

log = require('log')
fiber = require('fiber')
soap = require('SOAPInstance')

orderNum = 0

appName = 'OrderManagement'
portType = '{http://www.bercut.com/spec/wsdl/ProductOrdering}ProductOrdering'

local onCreateProductOrderFiber = function(productOrderingReqMsg)
  orderNum = orderNum + 1
  local productOrderingRespMsg = {
    data = {
      createProductOrder = {
        createProductOrderResponse = {
          param = {
            orderId = "PORD-SOAP-" .. tostring(orderNum)
          }
        }
      }
    }
  }
  return productOrderingRespMsg
end

instance = soap.SOAPInstance(appName)
server = instance:openListener(portType, onCreateProductOrderFiber)

require('console').start()

server:close()
instance:close()

require('os').exit()

```

## Клиент на базе Tarantool SOAP Adapter с использованием единого callBack и дополнительными MIB-параметрами

### server.lua

```

---
--- Generated by EmmyLua(https://github.com/EmmyLua)
--- Created by bercut.
--- DateTime: 19.12.2020 10:08
---
data_dir = "server"
box.cfg({
  log_level = 5
  , listen = 3131
--  , log = 'server.txt'
  , wal_dir = data_dir
  , memtx_dir = data_dir
  , vinyl_dir = data_dir
})
box.once('init', function()
  box.schema.user.grant("guest", 'read,write,execute,create,drop', 'universe')
end)

log = require('log')

```



```

fiber = require('fiber')
soap = require('SOAPInstance')

orderNum = 0

appName = 'OrderManagement'
portTypePO = '{http://www.bercut.com/spec/wsdl/ProductOrdering}ProductOrdering'
portTypeAM = '{http://www.bercut.com/spec/wsdl/AccountManagement}AccountManagement'

local onIncomingRequest = function(reqMsg, porttype)

    if porttype == portTypePO
    then
        orderNum = orderNum + 1
        local productOrderingRespMsg = {
            data = {
                createProductOrder = {
                    createProductOrderResponse = {
                        param = {
                            orderId = "PORD-SOAP-" .. tostring(orderNum)
                        }
                    }
                }
            }
        }
        return productOrderingRespMsg
    end
    if porttype == portTypeAM
    then
        local accountManagementRespMsg = {
            data = {
                GetInfo = {
                    systemFault = {
                        fault = {
                            faultCode = "234",
                            faultString = "error"
                        }
                    }
                }
            }
        }
        return accountManagementRespMsg
    end

    return nil
end

parameters = {
    Description = 'SomeService',
    Version = '1.2.3.100',
    SomeInfo = 'Any info'
}

instance = soap.SOAPInstanceEx(appName, parameters)
serverPO = instance.openMultipleListener(portTypePO, onIncomingRequest)
serverAM = instance.openMultipleListener(portTypeAM, onIncomingRequest)

require('console').start()

serverPO:close()
serverAM:close()
instance:close()

require('os').exit()

```

## Клиент на базе Tarantool SOAP Adapter client.lua

```

---
--- Generated by EmmyLua(https://github.com/EmmyLua)
--- Created by bercut.
--- DateTime: 19.12.2020 10:08
---

```

```

data_dir = "client"

box.cfg({
  log_level = 6
--  , log = 'client.txt'
  , wal_dir = data_dir
  , memtx_dir = data_dir
  , vinyl_dir = data_dir
})

box.once('init', function()
  box.schema.user.grant("guest", 'read,write,execute,create,drop', 'universe')
end)

log = require('log')
fiber = require('fiber')
soap = require('SOAPInstance')

appName = 'ClientTest'
portType = '{http://www.bercut.com/spec/wsdl/ProductOrdering}ProductOrdering'

instance = soap.SOAPInstance(appName)
client = instance.openConnector(portType)

productOrderingRequest = {
  --TODO: здесь надо заполнить тело запроса
}

productOrderingResponse = client:invoke(productOrderingRequest)
if productOrderingResponse then
  print(string.format("result: %s",
    productOrderingResponse.data.createProductOrder.createProductOrderResponse.param.orderId))
else
  print("Error happens")
end

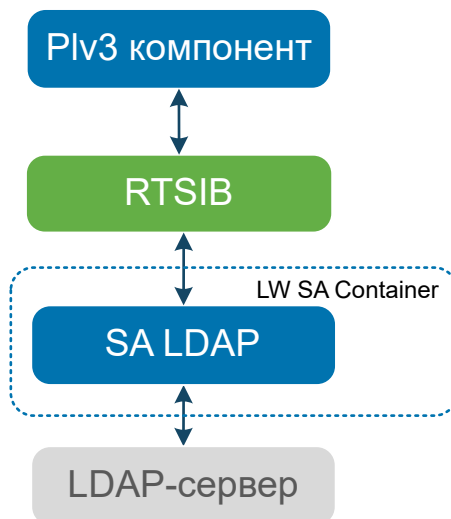
client:close()
instance:close()

require('console').start()
require('os').exit()

```

## 17. Специализированный адаптер SA LDAP

Специализированный адаптер SA LDAP обеспечивает взаимодействие LDAP-сервера и компонентов Platform v3.



**Рис. 172. SA LDAP**

SA LDAP использует следующие операции интерфейса SA DSI LDAP [11]:

- `addEntries`. Создание нескольких записей в списке.
- `addEntry`. Создание записи.
- `addEntryAttribute`. Добавление значения атрибута.
- `deleteEntries`. Удаление нескольких записей.
- `deleteEntry`. Удаление записи.
- `deleteEntryAttribute`. Удаление значения атрибута.
- `modifyEntryAttributeValue`. Изменение значения атрибута.
- `readEntry`. Получение записи.
- `searchEntries`. Поиск записей.
- `userAuthentication`. Проверка данных пользователя на LDAP-сервере.
- `userAvailability`. Проверка наличия пользователя.

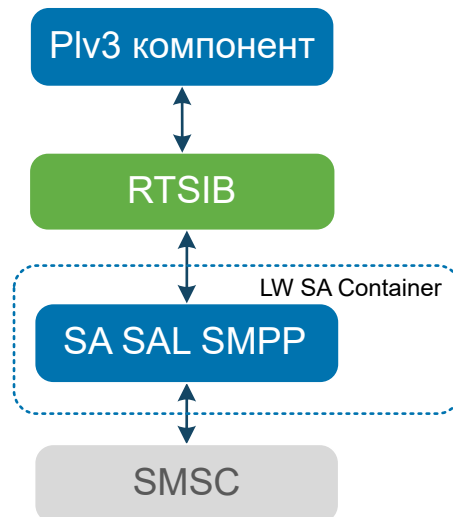
Для установки SA LDAP **подключите** его в среде LW SA Container.

Для настройки SA LDAP выполните следующие действия:

- Задайте MIB-переменные группы `SA_LDAP_vX.Y/Core/Configuration` [12].
- Для операций `userAuthentication` и `userAvailability` задайте параметры:
  - `LdapSearchAttribute`. Имя атрибута в фильтре поиска, которому система присваивает искомое имя пользователя. Пример: `username=Oleg`, имя атрибута здесь: `'username'`.
  - `LdapRoleAttribute`. Атрибут с ролями, которые система извлекает при поиске пользователя. Эти значения возвращает операция `userAuthentication`.

## 18. Специализированный адаптер SA SAL SMPP

Специализированный адаптер SA SAL SMPP обеспечивает взаимодействие SMSC и компонентов Platform v3.



**Рис. 173. SA SAL SMPP**

Функциональные возможности SA SAL SMPP:

- отправка и прием SMS-сообщений от SMSC;
- поддержка соединения с SMSC — отправка запроса EnquireLink;
- повторные попытки соединения с SMSC при ошибках;
- поддержка диалогов по адресам и типам транспорта. Для USSD используются дополнительные поля: `its_session_info` и `ussd_service_op`;
- поддержка RTSIB-сессий для диалогов и отчетов о доставке;
- поддержка следующих режимов работы с SMSC: Transmitter, Receiver, Transceiver;
- синхронное и асинхронное взаимодействие с SMSC;
- поддержка следующих типов исходящих сообщений: `Submit_SM`, `Data SM`;
- прием от SMSC следующих сообщений: `Deliver SM`, `Data SM`;
- возможность отправки мультиадресных сообщений `SUBMIT_MULTI`;
- поддержка следующих типов запроса о статусе доставки сообщения: `QUERY_SM`, `QUERY_LAST_MSGS`, `QUERY_MSG_DETAILS`;
- поддержка отмены сообщения `CANCEL_SM`;
- поддержка отправки и приема конкатенированных сообщений;
- выбор дополнительного поля `message_payload` для передачи данных;
- поддержка следующих типов отчетов о доставке (SMSC Delivery Receipt) в исходящих сообщениях: `Success or failure`, `Failure`;
- формирование общей SMPP-статистики и для каждого типа сообщения;
- установка размера очереди исходящих сообщений;
- установка правил для задания TON и NPI по адресу с помощью регулярного выражения;

- поддержка следующих режимов доставки сообщений (Messaging mode): Default SMSC mode, Store and Forward, Datagram, Forward (Transaction);
- поддержка режима автоматического выбора кодировки и длины сообщения (с учетом конкатенации). Кодировка настраивается в MIB;
- управление потоком при ошибках на SMSC: Throttling Error, Message Queue Full. Возможность повторной отправки сообщений при неуспешной попытке отправки.

Для установки SA SAL SMPP *подключите* его в среде LW SA Container.

SA SAL SMPP использует следующие операции интерфейсов AOI\_SAL\_Messaging и AOI\_SAL\_ActivationManager [11]:

- sendMessage. Отправка сообщения.
- cancelMessage. Отмена сообщения.
- getMessageDeliveryStatus. Получение статуса доставки сообщения.
- onReceiveMessage. Прием сообщения.
- onReceiveMessageDeliveryReceipt. Получение отчета о доставке сообщения.

Для настройки SA SAL SMPP задайте MIB-переменные группы SA\_SMPP\_vX.Y/Core/Configuration [12].

## 18.1. SMPP-соединение

SA SAL SMPP выполняет периодические попытки подключения к SMSC и контролирует SMPP-соединение.

Сценарий установки SMPP-соединения с SMSC:

- SA SAL SMPP пытается установить TCP-соединение с SMSC. Периодичность попыток задается в MIB-параметре *ReconnectDelay* группы SA\_SMPP\_vX.Y/Core/Configuration/SMPP/Connection/Timeouts;
- после установки TCP-соединения SA SAL SMPP отправляет в SMSC сообщение BindTransceiver со следующими параметрами:
  - system\_id
  - system\_type
  - password
- SA SAL SMPP ожидает ответ от SMSC в течение тайм-аута, заданного в MIB-параметре *BindTimeout* группы SA\_SMPP\_vX.Y/Core/Configuration/SMPP/Connection/Timeouts:
  - SMSC вернул успешный ответ — command\_status = 0. SA SAL SMPP объявляет MessagingPortType доступным;
  - при разрыве TCP-соединения SA SAL SMPP объявляет MessagingPortType недоступным и анализирует ошибку. SA SAL SMPP повторно пытается установить соединение с SMSC. Для ошибок, перечисленных в MIB-параметре *PermanentBindFailureCodes* группы SA\_SMPP\_vX.Y/Core/Configuration/SMPP/Connection, повторные попытки установления сессии не выполняются.

SA SAL SMPP может контролировать SMPP-соединение с помощью сообщений EnquireLink. Для принудительной отправки сообщений EnquireLink задайте значение True MIB-параметра *ForceEnquireLink* группы SA\_SMPP\_vX.Y/Core/Configuration/SMPP/Connection. Период отправки сообщений EnquireLink задайте в MIB-параметре

*EnquireLinkTimeout* группы `SA_SMPP_vX.Y/Core/Configuration/SMPP/Connection/Timeouts`.

## 18.2. Режимы взаимодействия с SMSC

SA SAL SMPP взаимодействует с SMSC в асинхронном и синхронном режимах.

SA SAL SMPP может взаимодействовать с SMSC в следующих режимах:

- асинхронный. SA SAL SMPP отправляет SMPP-запрос в SMSC, не дожидаясь поступления SMPP-ответа на предыдущий запрос;
- синхронный. SA SAL SMPP отправляет SMPP-запрос в SMSC только после получения SMPP-ответа на предыдущий запрос или после окончания тайм-аута. Тайм-аут задается в MIB-параметре *SMPPResponseTimeout* группы `SA_SMPP_vX.Y/Core/Configuration/SMPP/Timeouts`.

Для задания режима взаимодействия используйте MIB-переменную *CommunicationType* группы `SA_SMPP_vX.Y/Core/Configuration/SMPP/Connection`.

### Асинхронный режим взаимодействия

Асинхронный режим взаимодействия используется по умолчанию. Сценарий состоит из следующих шагов:

- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP для первого сообщения;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm` с первым сообщением;
- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP для следующего сообщения;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm` со следующим сообщением;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp` для первого сообщения;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES для первого сообщения;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp` для следующего сообщения;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES для следующего сообщения.

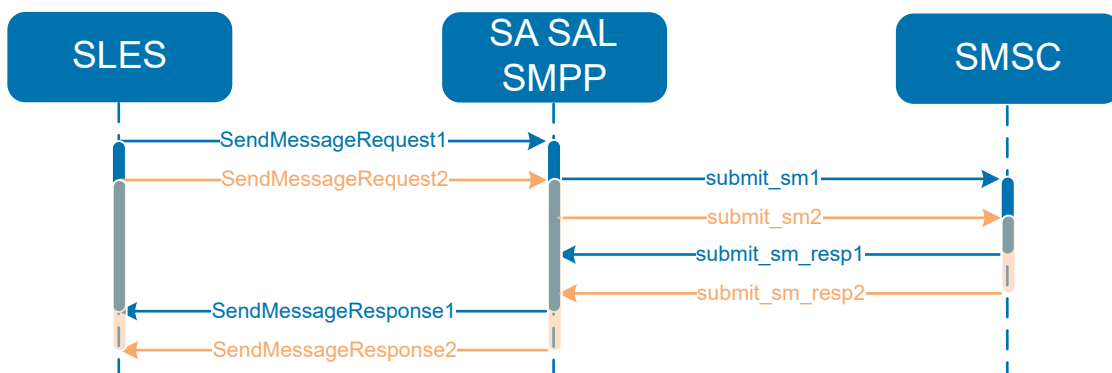
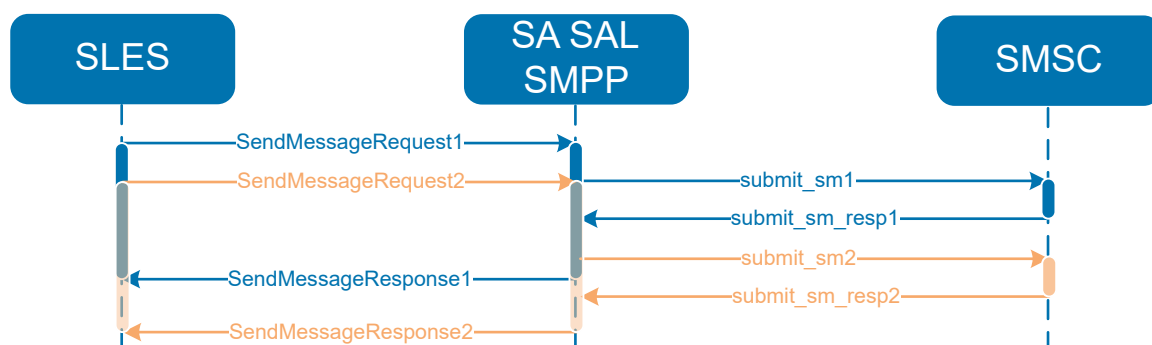


Рис. 174. Асинхронный режим

### Синхронный режим взаимодействия

Сценарий состоит из следующих шагов:

- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP для первого сообщения;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm` с первым сообщением;
- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP для следующего сообщения;
- SA SAL SMPP ожидает ответа от SMSC для первого сообщения;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp` для первого сообщения;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm` со следующим сообщением;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES для первого сообщения;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp` для следующего сообщения;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES для следующего сообщения.



**Рис. 175. Синхронный режим**

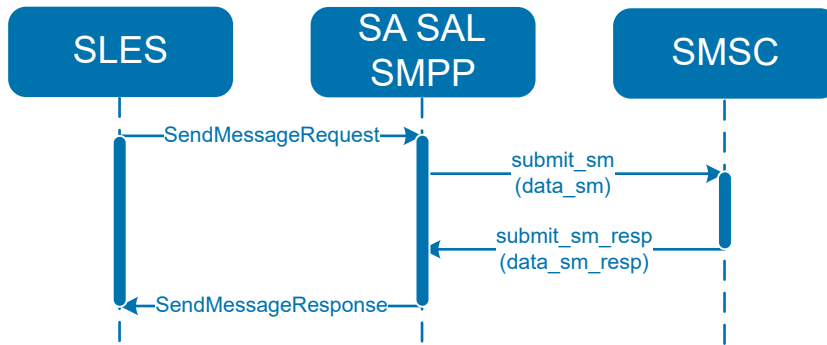
В синхронном режиме происходит накопление сообщений во внутренней очереди SA SAL SMPP при задержке SMPP-ответов. Размер очереди задается в MIB-переменной `OutgoingMessageQueueSize` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP`. При превышении этого значения прием новых сообщений невозможен.

### 18.3. Отправка SMS-сообщений в SMSC

Для отправки SMS-сообщений в SMSC используйте SOAP-операцию `sendMessage` [11].

Сценарий отправки SMS-сообщений в SMSC:

- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm` или `data_sm`. Тип сообщения определяется значением MIB-параметра `PacketType` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP`;
- SA SAL SMPP ожидает ответ от SMSC в течение тайм-аута, заданного в MIB-переменной `SMPPResponseTimeout` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP/Timeouts`;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp` или `data_sm_resp`. Тип сообщения определяется значением MIB-переменной `PacketType` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP`;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES.



**Рис. 176. Отправка SMS-сообщений в SMSC**

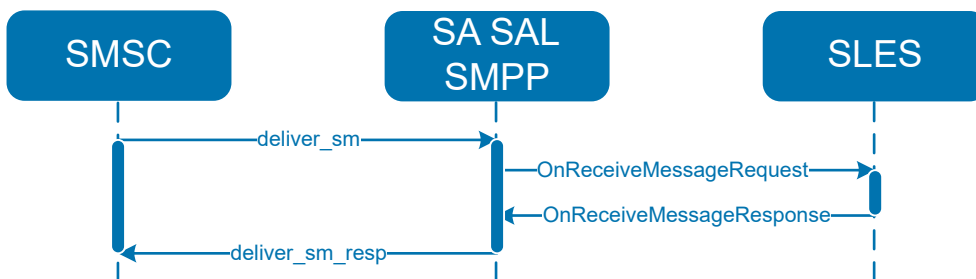
Для отправки сообщения нескольким получателям укажите их адреса в параметре `destAddresses` сообщения `sendMessageRequest`. Тип сообщения, которое в этом случае SA SAL SMPP отправляет в SMSC — `submit_milty_sm`.

### 18.4. Прием SMS-сообщений от SMSC

Для приема SMS-сообщений от SMSC используйте SOAP-операцию `onReceiveMessage` [11].

Сценарий приема SMS-сообщений от SMSC:

- SMSC отправляет в SA SAL SMPP пакет `deliver_sm`;
- SA SAL SMPP отправляет запрос `OnReceiveMessageRequest` в SLES;
- SLES отправляет ответ `OnReceiveMessageResponse` в SA SAL SMPP;
- SA SAL SMPP отправляет в SMSC пакет `deliver_sm_resp`.



**Рис. 177. Прием SMS-сообщений от SMSC**

Если пакет `deliver_sm` содержит отчет о доставке, SA SAL SMPP формирует SOAP-запрос `onReceiveMessageDeliveryReceipt` [11].

### 18.5. Получение отчета о доставке

Для получения отчета о доставке сообщения используйте SOAP-операцию `onReceiveMessageDeliveryReceipt` [11].

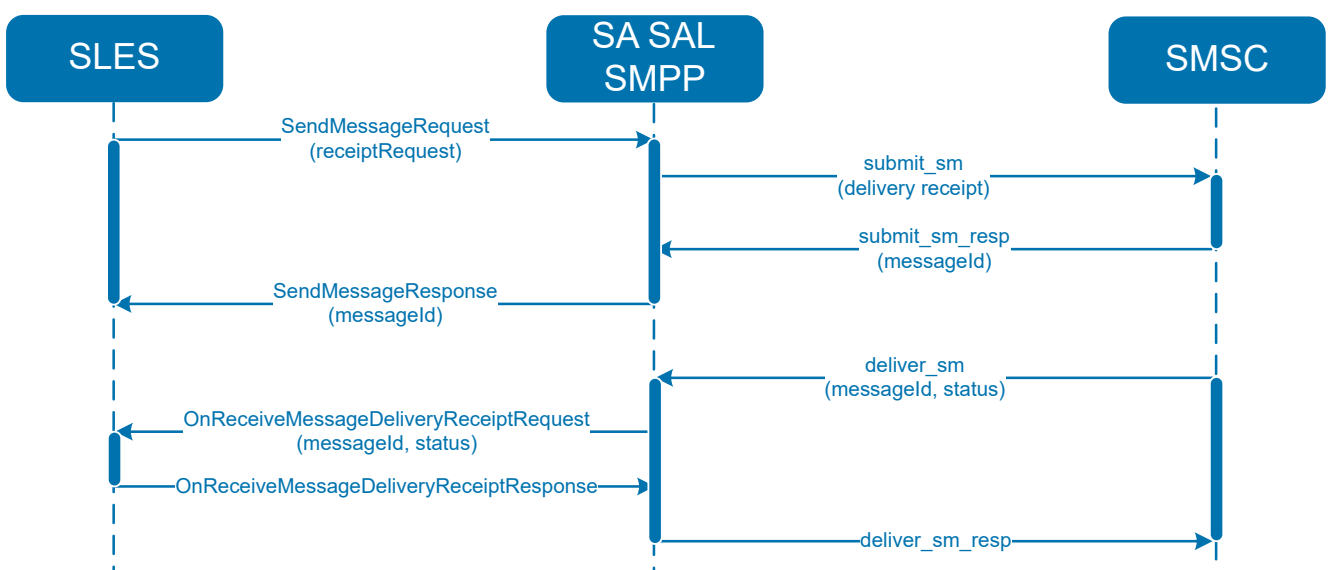
Сценарий получения отчета о доставке:



- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP. Параметр `receiptRequest=True`;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm`;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp`, который содержит идентификатор сообщения;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES. Ответ содержит идентификатор сообщения;
- SLES отправляет запрос `OnReceiveMessageDeliveryReceiptRequest` в SA SAL SMPP;
- SA SAL SMPP отправляет в SMSC пакет `deliver_sm`. SA SAL SMPP ожидает ответ от SMSC в течение тайм-аута, заданного в MIB-параметре `DeliveryReceiptTimeout` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP/Timeouts`. Если `deliver_sm_resp` не поступил после окончания тайм-аута, SA SAL SMPP отправляет повторный запрос `OnReceiveMessageDeliveryReceipt` с новым идентификатором сессии;
- SMSC отправляет в SA SAL SMPP пакет `deliver_sm_resp`;
- SA SAL SMPP отправляет в SLES ответ `OnReceiveMessageDeliveryReceiptResponse` с таким же идентификатором сессии, как и инициирующий запрос `SendMessage`.

В качестве дополнительного параметра ответа `OnReceiveMessageDeliveryReceiptResponse` указывается параметр `SmppMessageState`, в котором передается SMPP-статус. Статус определяется по параметру `message_state` SMPP-сообщения. Допустимые статусы:

- ACCEPTED — `message_state=6`;
- DELETED — `message_state=4`;
- DELIVERED — `message_state=2`;
- EN\_ROUTE — `message_state=1`;
- EXPIRED — `message_state=3`;
- INVALID — `message_state=7`;
- UNDELIVERABLE — `message_state=5`;
- UNKNOWN\_STATE — неизвестный `message_state`.



**Рис. 178. Получение отчета о доставке**

Для длинных сообщений SA SAL SMPP ожидает пакеты deliver\_sm\_resp от всех частей сообщения.

## 18.6. Получение статуса доставки сообщения

Для получения статуса доставки сообщения используйте SOAP-операцию `getMessageDeliveryStatus` [11].

Сценарий получения статуса доставки сообщения:

- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP. Параметр `receiptRequest=True`;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm`;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp`, который содержит идентификатор сообщения;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES. Ответ содержит идентификатор сообщения;
- SLES отправляет запрос `GetMessageDeliveryStatusRequest` в SA SAL SMPP. Запрос содержит идентификатор сообщения, для которого нужно получить статус доставки. Для длинного сообщения указывается идентификатор последней части;
- SA SAL SMPP отправляет в SMSC пакет `query_sm`, `query_last_msgs` или `query_msg_details`. Тип пакета определяется значением MIB-параметра `QueryPacketType` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP`;
- SMSC отправляет в SA SAL SMPP пакет `query_sm_resp`;
- SA SAL SMPP отправляет в SLES ответ `GetMessageDeliveryStatusResponse`, который в качестве дополнительного параметра содержит SMPP-статус (`message_state` SMPP-сообщения).

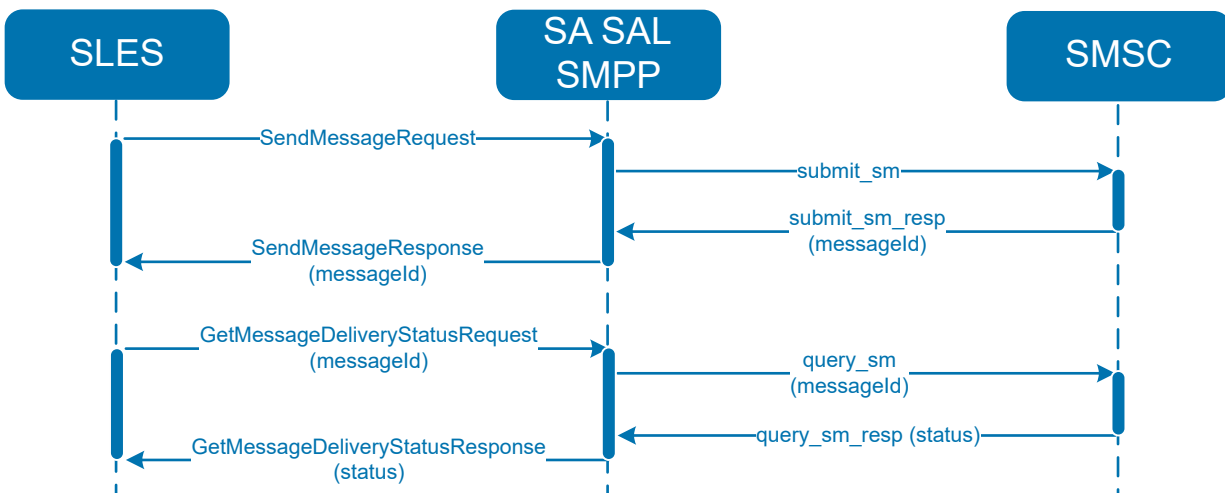


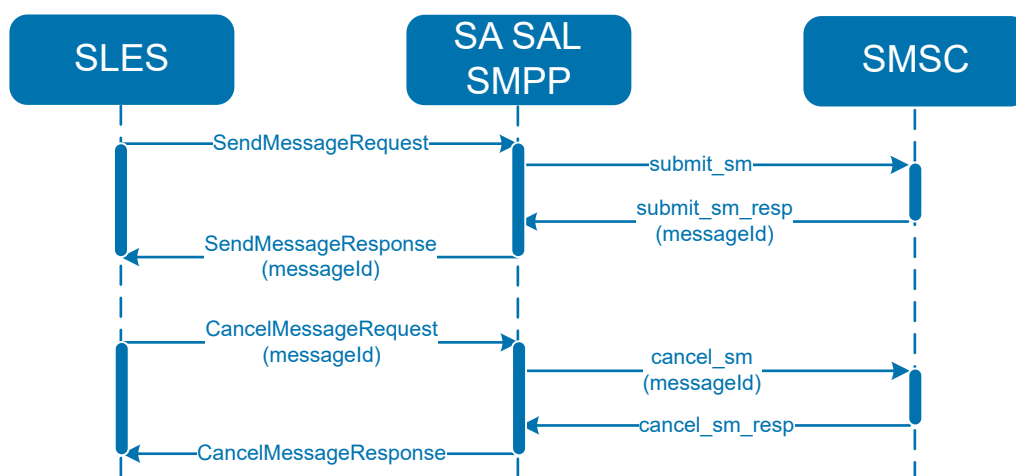
Рис. 179. Получение отчета о доставке

## 18.7. Отмена доставки сообщения

Для отмены доставки сообщения используйте SOAP-операцию `cancelMessage` [11].

Сценарий отмены доставки сообщения:

- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP. Параметр `receiptRequest=True`;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm`;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp`, который содержит идентификатор сообщения;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES. Ответ содержит идентификатор сообщения;
- SLES отправляет запрос `CancelMessageRequest` в SA SAL SMPP. Запрос содержит идентификатор сообщения, доставку которого нужно отменить;
- SA SAL SMPP отправляет в SMSC пакет `cancel_sm`;
- SMSC отправляет в SA SAL SMPP пакет `cancel_sm_resp`;
- SA SAL SMPP отправляет в SLES ответ `CancelMessageResponse`.



**Рис. 180. Получение отчета о доставке**

## 18.8. Обработка длинных сообщений

SA SAL SMPP может разбивать длинные сообщения на части.

Длинным считается сообщение, которое превышает по длине 254 октета. Максимальная длина части сообщения определяется его кодировкой:

- 7bit и 8bit — 153 символа;
- 16bit — 134 символа.

Данные пользователя в пакетах `SUBMIT_SM` и `SUBMIT_MULTY_SM` могут передаваться в полях:

- `SHORT_MESSAGE`. SA SAL SMPP разбивает длинные сообщения на части;
- `MESSAGE_PAYLOAD`. SA SAL SMPP не разбивает длинные сообщения на части.

SA SAL SMPP по умолчанию использует поле `MESSAGE_PAYLOAD`.

Для включения режима разбивки длинных сообщений задайте значение `'SHORT_MESSAGE'` в MIB-параметре `SubmitSmDataField` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP`.

### Отправка длинных сообщений

Сценарий состоит из следующих шагов:

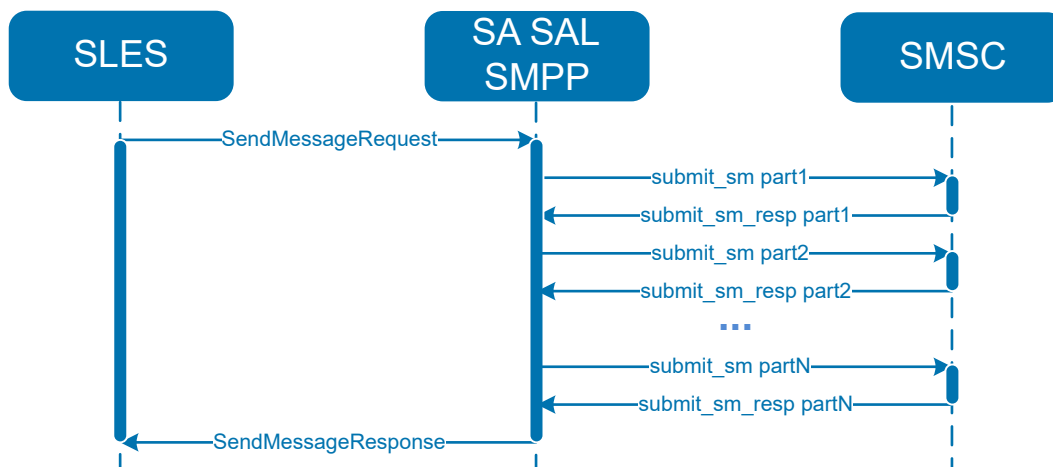
- SLES отправляет запрос SendMessageRequest в SA SAL SMPP;
- SA SAL SMPP разбивает сообщение на части по одному из методов передачи данных:
  - SAR. При использовании этого метода задайте следующую информацию в дополнительных полях:
    - sar\_msg\_ref\_num — идентификатор длинного сообщения (reference number);
    - sar\_total\_segments — общее количество частей сообщения;
    - sar\_segment\_seqnum — номер части.

В поле SHORT\_MESSAGE передаются данные пользователя.

- UDH. Дополнительная информация для последующей склейки сообщения передается вначале поля SHORT\_MESSAGE.

Метод передачи данных задается в MIB-параметре *LargeMessageTransferMethod* группы SA\_SMPP\_vX.Y/Core/Configuration/SMPP.

- SA SAL SMPP отправляет в SMSC пакет submit\_sm с первой частью сообщения;
- SMSC отправляет в SA SAL SMPP пакет submit\_sm\_resp для первой части сообщения;
- SA SAL SMPP отправляет в SMSC пакет submit\_sm со следующей частью сообщения;
- SMSC отправляет в SA SAL SMPP пакет submit\_sm\_resp для следующей части сообщения;
- SA SAL SMPP отправляет в SMSC пакет submit\_sm с последней частью сообщения;
- SMSC отправляет в SA SAL SMPP пакет submit\_sm\_resp для последней части сообщения;
- SA SAL SMPP отправляет ответ SendMessageResponse в SLES после получения ответа об отправке последней части сообщения;



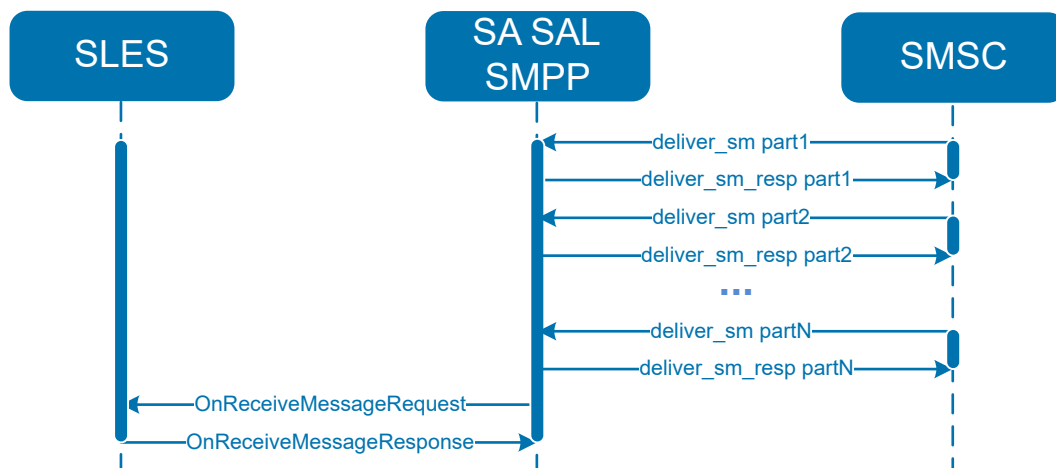
**Рис. 181. Отправка длинных сообщений**

### Прием длинных сообщений

Сценарий состоит из следующих шагов:

- SMSC отправляет в SA SAL SMPP пакет deliver\_sm с первой частью сообщения;
- SA SAL SMPP отправляет в SMSC пакет deliver\_sm\_resp для первой части сообщения;
- SMSC отправляет в SA SAL SMPP пакет deliver\_sm со следующей частью сообщения;
- SA SAL SMPP отправляет в SMSC пакет deliver\_sm\_resp для следующей части сообщения;

- SMSC отправляет в SA SAL SMPP пакет `deliver_sm` с последней частью сообщения;
- SA SAL SMPP отправляет в SMSC пакет `deliver_sm_resp` для последней части сообщения;
- SA SAL SMPP отправляет запрос `OnReceiveMessageRequest` в SLES;
- SLES отправляет ответ `OnReceiveMessageResponse` в SA SAL SMPP.



**Рис. 182. Прием длинных сообщений**

Система ожидает поступление всех частей сообщения в течение тайм-аута, заданного в MIB-переменной `LargeMessageTimeout` группы `SA_SMPP_vX.Y/Core/Configuration/SMPP/Timeouts`.

## 18.9. Режим немедленного ответа

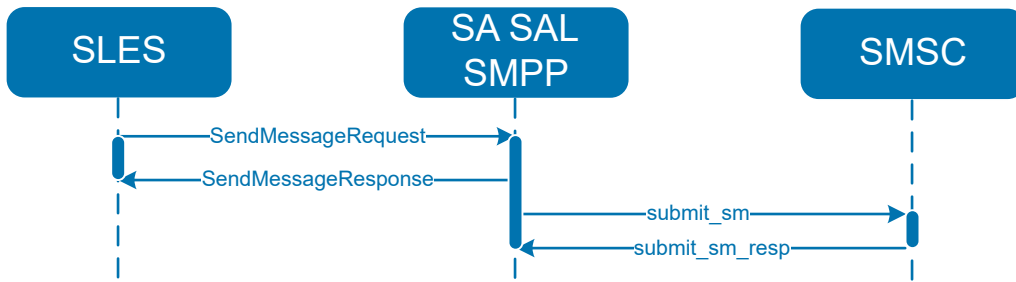
*SA SAL SMPP может отправлять SOAP-ответ, не дожидаясь соответствующего SMPP-пакета, при отправке сообщения и передавать SMPP-пакет, не дожидаясь SOAP-ответа, при получении сообщения.*

### Немедленный ответ при отправке сообщений

Для включения режима немедленного ответа задайте значение `False` в MIB-параметре `WaitSMPPResponse` группы `SA_SMPP_vX.Y/Core/Configuration/SOAP`.

Сценарий состоит из следующих шагов:

- SLES отправляет запрос `SendMessageRequest` в SA SAL SMPP;
- SA SAL SMPP отправляет ответ `SendMessageResponse` в SLES. В поле `messageId` передается значение: `'UNINITIALIZED'`. Запросить статус доставки или отменить такое сообщение не удастся;
- SA SAL SMPP отправляет в SMSC пакет `submit_sm`;
- SMSC отправляет в SA SAL SMPP пакет `submit_sm_resp`;



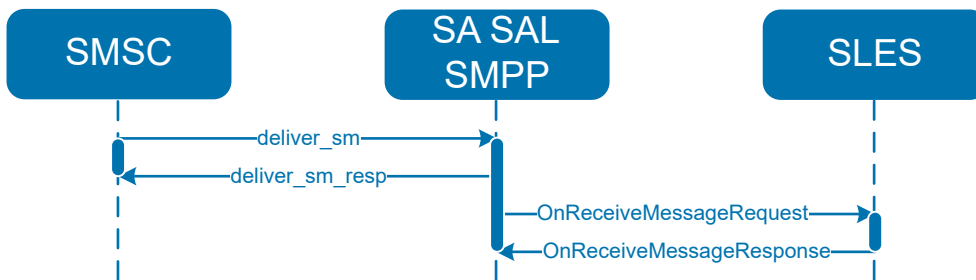
**Рис. 183. Немедленный ответ при отправке сообщений**

### Немедленный ответ при получении сообщений

Для включения режима немедленного ответа задайте значение False в MIB-переменной *WaitSOAPResponse* группы SA\_SMPP\_vX.Y/Core/Configuration/SOAP.

Сценарий состоит из следующих шагов:

- SMSC отправляет в SA SAL SMPP пакет *deliver\_sm*;
- SA SAL SMPP отправляет в SMSC пакет *deliver\_sm\_resp*;
- SA SAL SMPP отправляет запрос *OnReceiveMessageRequest* в SLES;
- SLES отправляет ответ *OnReceiveMessageResponse* в SA SAL SMPP.



**Рис. 184. Немедленный ответ при получении сообщений**

## 18.10. Поддержка диалогов в рамках одной сессии

SA SAL SMPP поддерживает взаимодействие с пользователем в рамках одной сессии. Пользователь обслуживается один экземпляром бизнес-процесса.

Для включения этой возможности задайте значение True в MIB-параметре *DialogSupport* группы SA\_SMPP\_vX.Y/Core/Configuration. Для задания тайм-аута ожидания ответа от пользователя используйте MIB-переменную *DialogTimeout* группы SA\_SMPP\_vX.Y/Core/Configuration.

Для SMS-диалогов условием для такого взаимодействия является совпадение адресов отправителя и получателя.

Для USSD-диалогов установите значение 'USSD' в MIB-переменной *ServiceType* группы SA\_SMPP\_vX.Y/Core/Configuration/SMPP/Connection. Для определения начала, продолжения и завершения USSD-сессии SA SAL SMPP использует следующие дополнительные поля SMPP-сообщений:

- *its\_session\_info*
- *ussd\_service\_op*

USSD-сессия определяется по адресам отправителя и получателя.

## 18.11. Кодировка SMPP-сообщений

SA SAL SMPP определяет кодировку сообщений по содержанию сообщения и настройкам MIB-группы *SA\_SMPP\_vX.Y/Core/Configuration/SMPP*.

Для выбора кодировки SA SAL SMPP анализирует коды символов сообщения:

- при наличии символов с кодами больше '255' используется кодировка: '16bit'. Тип кодировки задается в MIB-параметре *Default16bitEncoding*. Допустимые значения:
  - UCS2 (ISO/IEC-10646). Значение по умолчанию;
  - UTF-16BE (Big Endian);
  - UTF-16LE (Little Endian).
- при наличии символов с кодами в диапазоне от '127' до '255' или с кодами меньше '127' и значении MIB-параметра *Use7bitEncoding=False* используется кодировка: '8bit'. Тип кодировки задается в MIB-параметре *Default8bitEncoding*. Допустимые значения:
  - Latin-1 (ISO-8859-1). Значение по умолчанию;
  - IA5 (CCITT T.50)/ASCII (ANSI X3.4);
  - HP-Roman8 character set.
- при наличии символов с кодами меньше '127' и значении MIB-параметра *Use7bitEncoding=True* используется кодировка: '7bit'. Тип кодировки: 'GSM 7 bit SMSC Default Alphabet'.

SA SAL SMPP не поддерживает другие кодировки.

## 18.12. Указание SMPP-параметров в SOAP-запросе

Для задания SMPP-параметров используйте параметр *networkSpecificParams* в SOAP-запросе *sendMessage* [11].

В структуре *networkSpecificParams* задайте:

- имя параметра в теге *name*;
- значение параметра в двоичном виде в теге *value*.

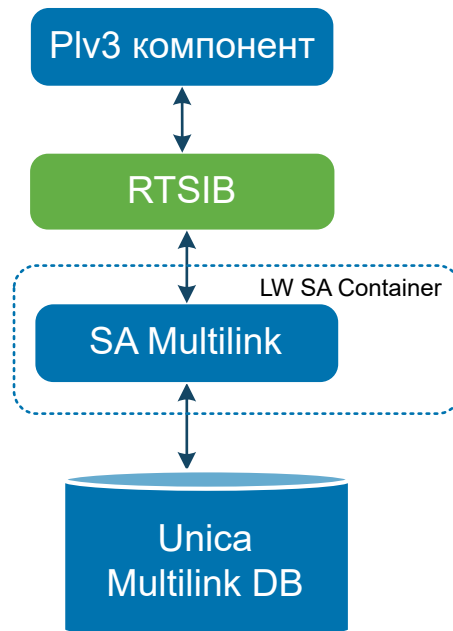
Пример:

```
<networkSpecificParams>
  <com:param>
    <com:name>protocol_id</com:name>
    <com:value>1000000</com:value>
  </com:param>
</networkSpecificParams>
```

**i Примечание.** Если значение параметра *esm\_class* не задано принудительно, система использует значение, указанное в MIB-параметре *MessageMode* группы *SA\_SMPP\_vX.Y/Core/Configuration/SMPP*.

## 19. Специализированный адаптер SA Multilink

Специализированный адаптер SA Multilink обеспечивает взаимодействие Unica Multilink DB и компонентов Platform v3.



**Рис. 185. SA Multilink**

Функциональные возможности SA Multilink:

- загрузка сообщений в Unica Multilink DB;
- JMS-нотификация компонента Platform v3.

Для установки SA Multilink *подключите* его в среде LW SA Container.

SA Multilink использует следующие операции интерфейсов AOI\_SAL\_Messaging и AOI\_SAL\_ActivationManager [11]:

- sendMessage. Отправка сообщения.
- cancelMessage. Отмена сообщения.
- getMessageDeliveryStatus. Получение статуса доставки сообщения.
- onReceiveMessage. Прием сообщения.
- onReceiveMessageDeliveryReceipt. Получение отчета о доставке сообщения.

Для настройки SA Multilink выполните следующие действия:

- Задайте MIB-переменные группы SA\_Multilink\_vX.Y/Core/Configuration [12].
- Задайте следующие значения системных параметров Unica Multilink DB [14]:
  - DisableAddToSaQueue=N.
  - AllowStateEventsHandlers=Y.
  - OnlyStateEventsHandlers=Y
- Настройте *JMS-подключение* для Unica Multilink DB.



## 20. Установка и настройка компонентов платформы Unica для CareM v3

В продуктах, которые реализованы на CareM v3, логика работы описывается в профилях Unica. Дополнительно настройте компоненты платформы Unica.

Порядок настройки компонентов платформы Unica:

1. [Установка и запуск](#) Unica SLR.
2. [Развертывание](#) Unica-профилей.
3. [Настройка](#) взаимодействия Service Gateway и Unica SLR.

### 20.1. Установка и запуск Unica SLR

Unica SLR реализует сценарии обслуживания и унифицированное взаимодействие с внешними системами и агентами Unica.

Порядок установки Unica SLR [5]:

1. Переключитесь на пользователя *bercut*. Все последующие действия выполняйте от имени этого пользователя.
2. Создайте каталог установки Unica SLR.
3. Скопируйте каталог дистрибутива в созданный каталог.
4. Распакуйте архив установочного дистрибутива.
5. Установите права на запуск, запись и чтение созданному каталогу.

```
chmod -R 755 /opt/SLR
```

6. Перейдите в каталог `bin`.
7. Отредактируйте следующие элементы скрипт-файла `run.sh`:
  - MIB\_NAME. Имя MIB-группы системы.
  - PRODUCT\_HOME. Путь к установочным файлам — каталогу, созданному на шаге 2.
  - JAVA\_HOME. Здесь задается путь к каталогу, где находится java.

8. Выполните команду:

```
./run.sh
```

Система автоматически создаст дерево в приложении ATLAS MIB Explorer.

9. Проверьте в MIB, что Unica SLR доступен. Параметр *Active* группы `Unica_SLR/Status & Control` должна иметь значение '1'.

**i Примечание.** При первом старте до введения значения лицензии, система будет работать в течение нескольких секунд.

10. Задайте лицензию в MIB-параметре *Signature* группы `Unica_SLR/Configuration/Licensing` [5].
11. Проверьте трейс в приложении ATLAS MIB Explorer на отсутствие ошибок от системы Unica SLR.

Настройте соединение с Unica-агентами и внешними системами [5].

## 20.2. Развертывание Unica-профилей

Для создания и развертывания Unica-профилей скопируйте профили вручную или с помощью Unica Visual Builder Light Web [7].

Для работы с Unica-профилями проверьте наличие:

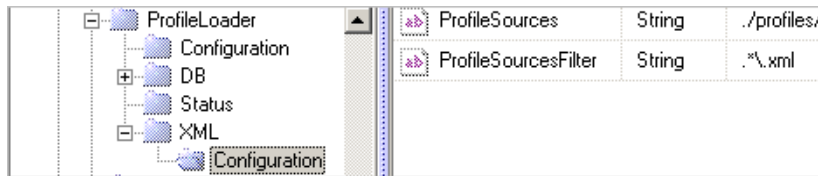
- Установленной и настроенной системы Unica Visual Builder Light Web при его использовании.
- Установленной и настроенной системы Unica SLR.

Порядок создания или развертывания Unica-профиля:

1. Сгенерируйте сценарий [7].
2. Примените сценарий к рабочей версии системы [7].

**Примечание.** Пункты 1 и 2 выполните, если используется Unica Visual Builder Light Web.

3. XML-файлы Unica-профилей выложите в каталог `\profiles\vbl` каталога установки Unica SLR.
4. Укажите путь к Unica-профилям в MIB-параметре `ProfileSources` группы `Unica_SLR/ProfileLoader/XML/Configuration`. В примере значение переменной: `./profiles/`.



**Рис. 186.** Группа `Unica_SLR/ProfileLoader/XML/Configuration`

**Примечание.** Если значение переменной `ProfileSources` было изменено, перезапустите Unica SLR.

*Настройте* Service Gateway.

## 20.3. Настройка взаимодействия Service Gateway и Unica SLR

Для настройки взаимодействия компонентов платформы Unica с компонентами Platform v3 отредактируйте файл `ServiceGatewaySystem.xml`.

Для этого необходимо:

1. *Настройте* Service Gateway.
2. Перейдите в каталог `profiles\vbl\system` каталога установки Unica SLR.  
`cd /opt/SLR/profiles/vbl/system`
3. Отредактируйте следующие элементы файла `ServiceGatewaySystem.xml`:

- **connectionstring** — строка соединения с Service Gateway. Формат: `http://<IP-  
address сервера>:<номер порта>/`.
- **username** — имя пользователя для Service Gateway.
- **password** — пароль пользователя к Service Gateway.

## 21. Расширение системы

*Для расширения системы установите дополнительные экземпляры компонентов.*

Установка дополнительных экземпляров SLES, Service Gateway и LW SA Container описана в разделах [9.1](#), [12.1](#) и [10.1](#) соответственно.

Для запуска дополнительных экземпляров Unica SLR укажите в скрипте `run.sh` отличное от уже установленных экземпляров MIB-имя в параметре **MIB\_NAME**. Затем запустите Unica SLR, [20.1](#).

## 22. Резервирование и масштабирование

Каждый компонент Platform v3 масштабируется и резервируется за счет подключения к общей шине RTSIB. Общая схема подключения: ' $N + m$ ', где ' $N$ ' — количество основных экземпляров и ' $m$ ' — количество резервных экземпляров.

Резервирование и масштабирование осуществляется согласно [диаграмме развертывания](#) системы.

Типовым подключением SA к внешней системе является следующая схема: один экземпляр внешнего компонента подключен к одному экземпляру SA. Например, подключается экземпляр SMSC, работающий с определенным диапазоном номеров. В случае, если SMSC не резервирован, возможно подключение дополнительного резервного SA к данному SMSC.

Типовым подключением внешних приложений к Service Gateway является следующая схема: несколько внешних компонентов, логически объединяемых в одно внешнее приложение и использующих один набор сервисов и бизнес-процессов Platform v3, подключаются к одному экземпляру Service Gateway. Например, подключаются RBT, компоненты Front и Back Office. Типовой схемой резервирования Service Gateway является HA cluster.

Типовая схема подключения SMS-, USSD- и Voice-, Video-порталов в части использования сервисов и бизнес-процессов Platform v3 соответствует схеме подключения внешних приложений.

### **i** **Примечание.**

*Резервирование и масштабирование системы имеет некоторые особенности:*

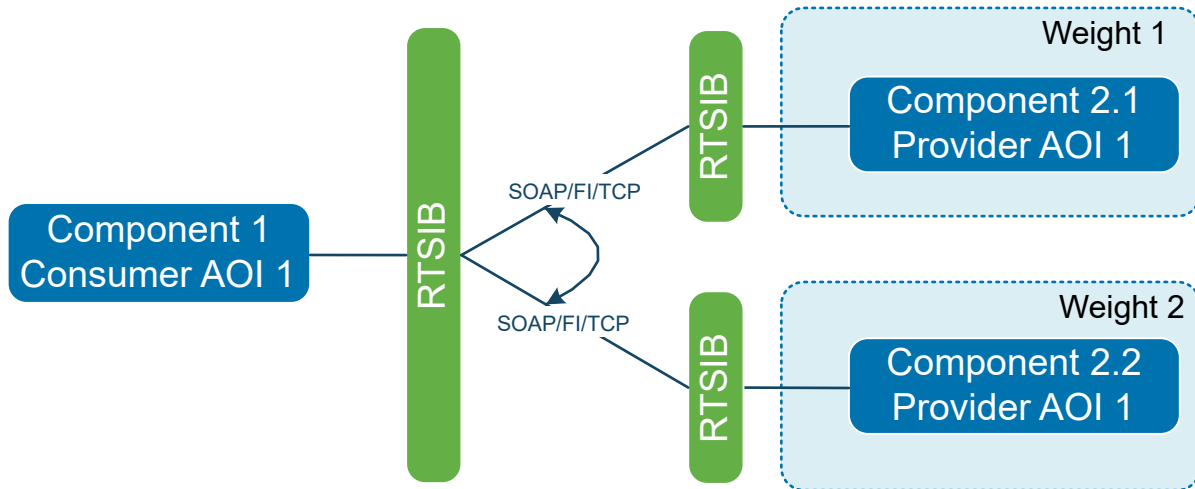
- *Convergent Media, Execution и Showcase Servers масштабируются по схеме ' $N + 1$ '.*
- *Для Access Server (DMZ) и Access Server (Trusted Domain) в большинстве случаев достаточно только резервирования.*
- *Для File Resource, Audit & Report и Managment Server используется только резервирование.*
- *Development Server не требует масштабирования или резервирования.*
- *Используются операционные системы RedHat, CentOS (виртуализация KVM) или Solaris (виртуализация Solaris Containers).*

### 22.1. Балансировка нагрузки и резервирование шины RTSIB

*Балансировка нагрузки предполагает равномерное распределение нагрузки на экземпляры, а резервирование — переключение на резервный экземпляр при выходе из строя основного.*

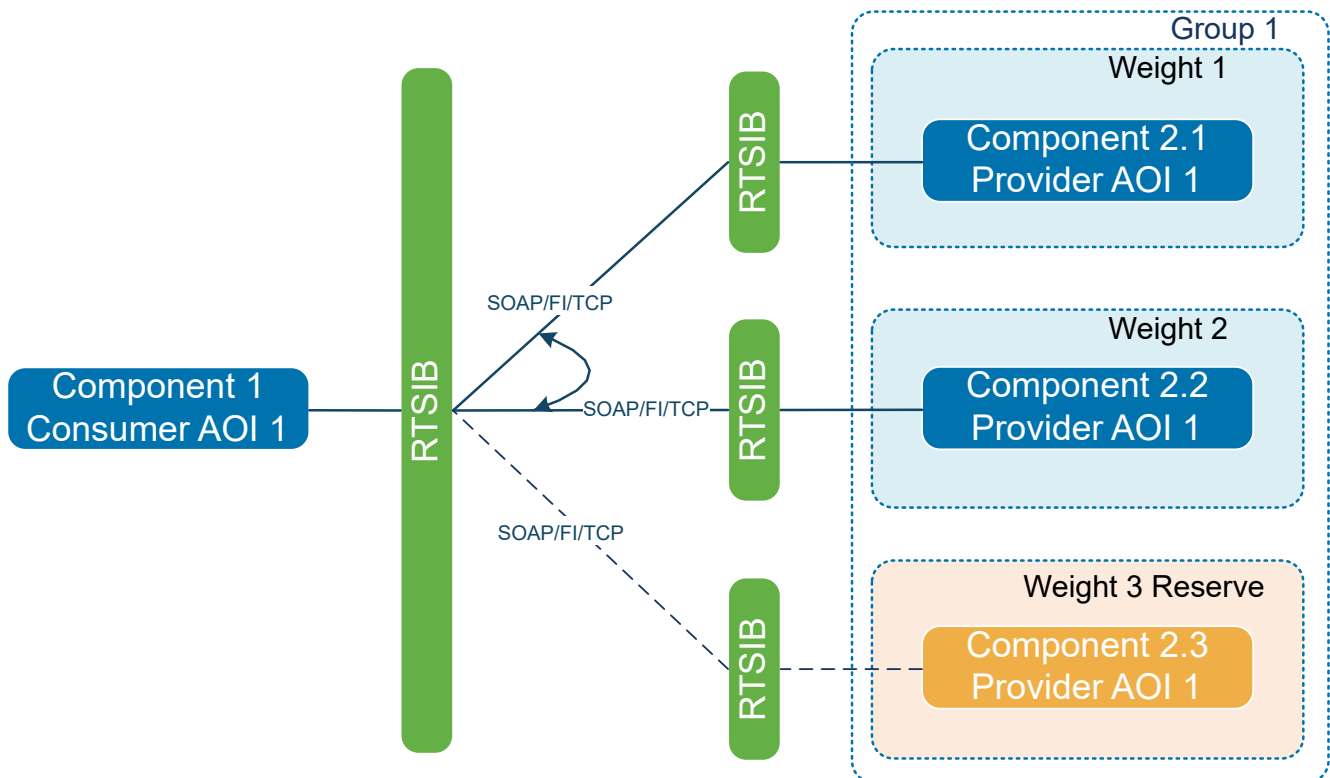
RTSIB обеспечивает распределение иницирующей сессии сообщений между несколькими экземплярами компонентов, реализующих определенный интерфейс. Сообщения, продолжающие сессию, отправляются по маршруту иницирующего сообщения. По умолчанию осуществляется равномерное распределение нагрузки, перераспределение нагрузки может осуществляться назначением веса маршруту.

Балансировка нагрузки позволяет решать как задачи масштабирования, так и резервирования. В одну группу масштабирования или резервирования могут быть объединены компоненты, реализующие один интерфейс. В тоже время компоненты могут быть доступны по разным транспортам.



**Рис. 187. Обеспечение балансировки нагрузки**

Резервирование обеспечивается распределением инициирующих сообщений на резервный экземпляр при отказе одного из основных.



**Рис. 188. Обеспечение резервирования**

## 23. Мониторинг системы

Для мониторинга работы компонентов используется система администрирования и мониторинга ATLAS.

Система ATLAS регистрирует события, возникающие в ходе работы компонентов: информационные, сообщения аварии и сбои в работе.

Также работа контролируемых объектов характеризуется рядом статистических параметров, значения которых также фиксируются системой менеджмента.

### 23.1. Сообщения о работе системы

Все приложения в процессе своей работы генерируют сообщения. Система ATLAS позволяет отслеживать данные события.

Системы и компоненты в процессе своей работы генерируют сообщения, информирующие как о событиях информационного характера (начало сессии, установление соединения и т.п.), так и об авариях и сбоях.

К первому типу сообщений (информационного характера) относятся *трейсовые сообщения*. Трейсовые сообщения характеризуются уровнем детальности, настройку которого администратор может выполнить для каждого из компонентов индивидуально.

**i Примечание.** Описание настройки уровня детальности трейсовых сообщений приводится в данном руководстве.

Трейсовые сообщения не требуют действий администратора и служат только в информационных целях.

Ко второму типу сообщений (сообщения об авариях и сбоях) относятся *алармы*. Алармы характеризуются уровнем критичности в зависимости от важности произошедшего события и необходимости вмешательства администратора в работу компонента. Алармы всегда, в отличие от трейсовых сообщений, записываются в журнал событий.

В зависимости от аварии, вызвавшей появление аларма, администратор должен предпринять те или иные действия для ликвидации аварии или предотвращения возможной аварийной ситуации. В списке алармов для каждого из них приводятся рекомендуемые действия администратора.

Сообщения записываются системой ATLAS в журнал событий и могут быть просмотрены администратором в приложении ATLAS MIB Explorer как в режиме реального времени, так и посредством загрузки журнала событий за определенный интервал (либо последних N событий). Данный инструмент позволяет оценивать состояние и логику работы компонентов, а также предупреждать возможные аварийные ситуации, реагируя на произошедшие события.

Администратору предоставляется возможность фильтрации сообщений по типу источника, номеру хоста, уровню критичности (детальности), параметрам сообщений.

## 23.2. Статистика работы систем и компонентов с помощью ATLAS

Статистическая информация о ходе работы компонентов и систем отображается в приложении ATLAS MIB Explorer.

В процессе работы компонентов и систем происходит обновление статистических параметров, характеризующих те или иные аспекты их функционирования. В приложении ATLAS MIB Explorer в MIB-группах компонентов отображаются следующие статистические данные:

- Статистических переменных состояния (нечислового типа). Обновление таких параметров в MIB осуществляется в режиме реального времени сразу после изменения значения переменной;
- Агрегированные (обработанные) числовые статистические данные. В зависимости от типа агрегации, определенного для переменной, через фиксированный промежуток времени осуществляется вычисление ее значения, результат которого также записывается в MIB. Тип агрегации, определенный для данной переменной, указывается в ее описании.

Возможны следующие виды агрегации:

1. `sum` (сумма). Значения переменной суммируются в течение интервала обновления статистики, после чего полученное суммарное значение присваивается данной переменной в MIB. Вычисление значений следующего интервала времени начинается с нуля.
2. `counter` (счетчик). Значения переменной суммируются в течение интервала обновления статистики, после чего полученная сумма прибавляется к текущему значению переменной. Значения статистики за следующий интервал прибавляются к сумме т. д. Обнуление переменной происходит при старте приложения.
3. `average` (среднее). Осуществляется запись всех значений переменной за интервал времени, а также ведется подсчет количества записанных значений. После окончания интервала вычисляется среднее арифметическое значение переменной, которое и записывается в MIB.
4. `average speed` (средняя скорость). За интервал времени все полученные значения переменной складываются. По окончании интервала обновления статистики полученная сумма делится на длительность интервала, сек. Вычисленное таким образом значение записывается в MIB.
5. `max` (максимум). После окончания интервала времени в MIB записывается максимальное значение переменной из всех полученных за этот интервал.
6. `min` (минимум). После окончания интервала времени в MIB записывается минимальное значение переменной из всех полученных за этот интервал.

Интервал вычисления значений числовых переменных определяется настройками данного компонента.

Одним из показателей работы является текущий статус доступности объектов, который определяется переменной `Status` в группе настроек `.../Statistics/` (для компонентов — `/Core/Statistics/`).

Статистические параметры, отражающие производительность работы, содержатся в группе `.../Statistics/Performance/`. Параметры, характеризующие качество работы, содержатся в группе `.../Statistics/Quality/`.



# Приложение 1. Использование Service Profile Management. Пример работы с сервисом

Установочный пакет *Service Profile Management* включает в себя каталог дистрибутива, а также примеры запросов и ответов.

Для работы с сервисом используйте каталог дистрибутива **service-management-X.Y-multiplatform-release**, содержащий:

- Скрипт-файлы для подключения XSD-файлов — каталог **bin**.
- Скрипт-файлы для установки SP DB — каталог **db**.
- Набор библиотек — каталог **lib**.
- Набор стандартных XSD-файлов — каталог **xsd**.

Для каждого сервиса используйте свой XSD-файл со специфичным словарем типов сервиса и со ссылками на используемые стандартные словари типов. Пример профиля сервиса:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bercut.com/spec/schema/ServiceName"
  xmlns:tns="http://www.bercut.com/spec/schema/ServiceName"
  elementFormDefault="qualified"
  xmlns:sd="http://www.bercut.com/spec/schema/SimpleDefinition"
  xmlns:pd="http://www.bercut.com/spec/schema/ProfileDefinition"
  xmlns:lbsdef="http://www.bercut.com/specs/schemas/ServiceNameDefinitions">
  <xsd:import schemaLocation="SimpleDefinition.xsd"
    namespace="http://www.bercut.com/spec/schema/SimpleDefinition"/>
  <xsd:import schemaLocation="ProfileDefinition.xsd"
    namespace="http://www.bercut.com/spec/schema/ProfileDefinition"/>
  <xsd:import schemaLocation="ServiceNameDefinitions.xsd"
    namespace="http://www.bercut.com/specs/schemas/ServiceNameDefinitions"/>
  <xsd:complexType name="Type1Name">
    <xsd:annotation>
      <xsd:documentation>Type1 description</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element
        name="Element1_1_Name"
        type="tns:Element1_1_Type">
        <xsd:annotation>
          <xsd:documentation>Element1_1 description</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element
        name="Element1_N_Name"
        type="tns:Element1_N_Type">
        <xsd:annotation>
          <xsd:documentation>Element1_N description</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TypeM_Name">
    <xsd:annotation>
      <xsd:documentation>TypeM description</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element
        name="ElementM_1_Name"
        type="tns:ElementM_1_Type">
        <xsd:annotation>
          <xsd:documentation>ElementM_1 description</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

```

    <xsd:element
      name="ElementM_N_Name"
      type="tns:ElementM_N_Type">
      <xsd:annotation>
        <xsd:documentation>ElementM_N description</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="SimpleType">
  <xsd:annotation>
    <xsd:documentation>SimpleType description</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="restriction1_type">
    <xsd:enumeration value="enumeration1_value">
      <xsd:annotation>
        <xsd:documentation>enumeration1_description</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="restrictionN_value">
      <xsd:annotation>
        <xsd:documentation>restrictionN_description</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element
  name="ServiceNameProfile"
  type="tns:ServiceNameProfile"
  pd:typedef="ServiceType"
  pd:contextref="tns:ServiceContext">
</xsd:element>
</xsd:schema>

```

Перед началом использования выполните действия:

- Установите SP DB с помощью скрипт-файлов из каталога дистрибутива.
- Подключите стандартные XSD-файлы, входящие в установочный пакет. XSD-файлы расположены в каталоге `xsd`. Для подключения запустите файл `deploy_common_xsd.sh`, расположенный в каталоге `bin`. Файл `deploy_common_xsd.sh` содержит стандартные XSD. Для подключения всех XSD-файлов предварительно заархивируйте их в ZIP-архив и выполните команду:

```
./deploy.sh -c deploy -u http://IP-address:Port/ServiceProfile/SchemaPortType -z ../xsd/название_архива.zip
```

Для подключения одного стандартного XSD-файла выполните команду:

```
./deploy.sh -c deploy -x ../xsd/название_xsd-файла.xsd -n название_типа_сервиса
```

Если IP-адрес сервера сервиса отличается от IP-адреса сервера, на котором запускается скрипт, или номер порта отличается от стандартного значения **8080**, команда выглядит так:

```
./deploy.sh -c deploy -u http://IP-address:Port/ServiceProfile/SchemaPortType -x ../xsd/название_xsd-файла.xsd -n название_типа_сервиса
```

- Подключите XSD-файл сервиса. Действия по подключению аналогичны описанным в предыдущем пункте.

В примере работы с сервисом в рамках Service Profile Management рассмотрены следующие операции:

- Создание сервиса — `createService`.
- Чтение профиля сервиса — `readService`. Чтение профиля выполняется целиком или для конкретного пользователя.

- Редактирование профиля — updateService.

## Создание сервиса

Запрос на создание сервиса createServiceRequest:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:createServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <sim:serviceName>ServiceNameProfile</sim:serviceName>
      <sim:serviceType>ServiceName</sim:serviceType>
      <sim:serviceStatus>ServiceStatus</sim:serviceStatus>
      <sim:tag>ServiceName</sim:tag>
      <ser:serviceContent>
<ns0:ServiceNameProfile
  xmlns:ns0="http://www.bercut.com/spec/schema/ServiceNameProfile">
  <ns0:serviceParams>
    <ns0:userList>
      <ns0:user>
        <ns0:priority>User1_Priority</ns0:priority>
        <ns0:Parameter1_Name for User1>Parameter1 Value</ns0:Parameter1_Name for User1>
        ...
        <ns0:ParameterM_Name for User1>ParameterM Value</ns0:ParameterM_Name for User1>
        <ns0:rule>
          <ns0:login>User1_Login</ns0:login>
        </ns0:rule>
      </ns0:user>
      ...
      <ns0:user>
        <ns0:priority>UserN_Priority</ns0:priority>
        <ns0:Parameter1_Name for UserN>Parameter1 Value</ns0:Parameter1_Name for UserN>
        ...
        <ns0:ParameterM_Name for UserN>ParameterM Value</ns0:ParameterM_Name for UserN>
        <ns0:rule>
          <ns0:login>UserN_Login</ns0:login>
        </ns0:rule>
      </ns0:user>
    </ns0:userList>
  </ns0:serviceParams>
</ns0:ServiceNameProfile>
</ser:serviceContent>
  </ser:createServiceRequestParams>
</soapenv:Body>
</soapenv:Envelope>
```

Ответ на операцию создания сервиса createServiceResponse:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber-addr="http://www.bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
  xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:createServiceResponseParams>
      <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
      <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
      <ber-sd:serviceType>ServiceNameProfile</ber-sd:serviceType>
      <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
      <ber-sd:description>ServiceName</ber-sd:description>
      <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
    </ber-sp:createServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## Чтение профиля сервиса

Запрос на чтение всего профиля сервиса readServiceRequest:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition"
xmlns:com="http://www.bercut.com/spec/schema/ComplexDefinition"
xmlns:ns0="http://xml.netbeans.org/schema/ServiceNameProfile">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:readServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <com:serviceIds>
        <sim:serviceName>ServiceName</sim:serviceName>
      </com:serviceIds>
      <sim:contractIds>all</sim:contractIds>
      <ser:parameterContext>
        <ser:serviceContext>
          <serviceContext>
            <Login>UserLogin</Login>
          </serviceContext>
        </ser:serviceContext>
        <sim:xpath>/serviceParams/userList</sim:xpath>
      </ser:parameterContext>
    </ser:readServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>
```

Ответ на операцию чтения всего профиля сервиса readServiceResponse:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  <SOAP-ENV:Header xmlns:ber-addr="http://bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-ns0="http://www.bercut.com/spec/schema/ServiceNameProfile"
    xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
    xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:readServiceResponseParams>
      <ber-sp:serviceDescription>
        <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
        <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
        <ber-sd:serviceTypeName>ServiceNameProfile</ber-sd:serviceTypeName>
        <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
        <ber-sd:description>ServiceName</ber-sd:description>
        <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
      </ber-sp:serviceDescription>
      <ber-sp:serviceContent>
        <ber-ns0:ServiceNameProfile>
          <ber-ns0:serviceParams>
            <ber-ns0:userList>
              <ber-ns0:user>
                <ber-ns0:priority>User1_Priority</ber-ns0:priority>
                <ber-ns0:Parameter1_Name for User1>Parameter1 Value</ber-
ns0:Parameter1_Name for User1>
                ...
                <ber-ns0:ParameterM_Name for User1>ParameterM Value</ber-
ns0:ParameterM_Name for User1>
              </ber-ns0:user>
              <ber-ns0:rule>
                <ber-ns0:login>User1_Login</ber-ns0:login>
              </ber-ns0:rule>
            </ber-ns0:userList>
          </ber-ns0:serviceParams>
          ...
          <ber-ns0:user>
            <ber-ns0:priority>UserN_Priority</ber-ns0:priority>
            <ber-ns0:Parameter1_Name for UserN>Parameter1 Value</ber-
ns0:Parameter1_Name for UserN>
            ...
            <ber-ns0:ParameterM_Name for User1>ParameterM Value</ber-
ns0:ParameterM_Name for User1>
          </ber-ns0:user>
        </ber-ns0:ServiceNameProfile>
      </ber-sp:serviceContent>
    </ber-sp:readServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        <ber-ns0:login>UserN_Login</ber-ns0:login>
      </ber-ns0:rule>
    </ber-ns0:user>
  </ber-ns0:userList>
</ber-ns0:serviceParams>
</ber-ns0:ServiceNameProfile>
</ber-sp:serviceContent>
</ber-sp:readServiceResponseParams>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Запрос на чтение параметров пользователя в рамках сервиса readServiceRequest:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
xmlns:com="http://www.bercut.com/spec/schema/ComplexDefinition"
xmlns:ns0="http://xml.netbeans.org/schema/ServiceNameProfile">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:readServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <com:serviceIds>
        <sim:serviceName>ServiceName</sim:serviceName>
      </com:serviceIds>
      <sim:contractIds>all</sim:contractIds>
      <ser:parameterContext>
        <ser:serviceContext>
          <ns0:serviceContext>
            <Login>UserLogin</Login>
          </ns0:serviceContext>
        </ser:serviceContext>
        <sim:xpath>/serviceParams/userList</sim:xpath>
      </ser:parameterContext>
    </ser:readServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>

```

### Ответ на операцию чтения параметров пользователя в рамках сервиса readServiceResponse:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
<SOAP-ENV:Header xmlns:ber-addr="http://bercut.com/addressing">
  <ber-addr:From>
    <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
  </ber-addr:From>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:ber-ns0="http://www.bercut.com/spec/schema/ServiceNameProfile"
xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
  <ber-sp:readServiceResponseParams>
    <ber-sp:serviceDescription>
      <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
      <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
      <ber-sd:serviceTypeName>ServiceNameProfile</ber-sd:serviceTypeName>
      <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
      <ber-sd:description>ServiceName</ber-sd:description>
      <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
    </ber-sp:serviceDescription>
    <ber-sp:serviceContent>
      <ber-ns0:userList>
        <ber-ns0:user>
          <ber-ns0:priority>User_Priority</ber-ns0:priority>
          <ber-ns0:Parameter1_Name for User>Parameter1 Value</ber-ns0:Parameter1_Name
for User>
          ...
          <ber-ns0:ParameterM_Name for User>ParameterM Value</ber-ns0:ParameterM_Name
for User>
          <ber-ns0:rule>
            <ber-ns0:login>User_Login</ber-ns0:login>
          </ber-ns0:rule>
        </ber-ns0:user>
      </ber-ns0:userList>

```

```

    </ber-sp:serviceContent>
  </ber-sp:readServiceResponseParams>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Редактирование сервиса

### Запрос на редактирование сервиса updateServiceRequest:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
xmlns:com="http://www.bercut.com/spec/schema/ComplexDefinition">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:updateServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <com:serviceIds>
        <sim:serviceName>ServiceName</sim:serviceName>
      </com:serviceIds>
      <sim:serviceStatus>ServiceStatus</sim:serviceStatus>
      <ser:serviceParameter>
        <sim:xpath>serviceParams/userList</sim:xpath>
        <ser:serviceContent>
          <userList>
            <user>
              <priority>User1_Priority</priority>
              <Parameter1_Name for User1>Parameter1 Value</Parameter1_Name for User1>
              ...
              <ParameterM_Name for User1>ParameterM Value</ParameterM_Name for User1>
              <rule>
                <login>User1_Login</login>
              </rule>
            </user>
            ...
            <user>
              <priority>UserN_Priority</priority>
              <Parameter1_Name for UserN>Parameter1 Value</Parameter1_Name for UserN>
              ...
              <ParameterM_Name for UserN>ParameterM Value</ParameterM_Name for UserN>
              <rule>
                <login>UserN_Login</login>
              </rule>
            </user>
          </userList>
        </ser:serviceContent>
      </ser:serviceParameter>
    </ser:updateServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>

```

### Ответ на операцию редактирования сервиса updateServiceResponse:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber-addr="http://bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:updateServiceResponseParams>
      <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
      <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
      <ber-sd:serviceTypeName>ServiceNameProfile</ber-sd:serviceTypeName>
      <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
      <ber-sd:description>ServiceName</ber-sd:description>
      <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
    </ber-sp:updateServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Приложение 2. Недопустимые символы в строковых данных

Список недопустимых кодов символов в SOAP/XML-запросах, которые приводят к ошибкам в сервисах Platform v3:

- 0x00
- 0x01
- 0x02
- 0x03
- 0x04
- 0x05
- 0x06
- 0x07
- 0x08
- 0x0B
- 0x0C
- 0x0E
- 0x0F
- 0x10
- 0x11
- 0x12
- 0x13
- 0x14
- 0x15
- 0x16
- 0x17
- 0x18
- 0x19
- 0x1A
- 0x1B
- 0x1C
- 0x1D
- 0x1E
- 0x1F.

# История изменений системы

В разделе представлена история изменений системы.

## Изменения с версии 3.2 до 3.3

1. Источники данных на LW SA Container настраиваются с помощью MIB-парметров, а не в XML-файлы.
2. Дополнены статистические параметры шины RTSIB и Service Gateway.
3. Доступ к операциям по бизнес-параметрам в Service Gateway можно выполнять с использованием алиасов.
4. Добавлена поддержка СУБД Oracle timesten.
5. В дистрибутив LW SA Container добавлены драйверы Oracle11, Oracle12, PostgreSQL.
6. Изменены названия MIB-групп:
  - @libs/LIB\_MessageBus\_vX.Y/DataServer/ProtocolLayers/TcpFi на @libs/LIB\_MessageBus\_vX.Y/DataServer/ProtocolLayers/Tcp.
  - @libs/LIB\_MessageBus\_vX.Y/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/TcpFi на @libs/LIB\_MessageBus\_vX.Y/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp
7. Изменен алгоритм работы с необязательными MIB-параметрами, имеющими значение по умолчанию в ATLAS-описании.
8. Кеширование ответов на внешние вызовы.

## Изменения с версии 3.3 до 3.4

1. Изменение версииности компонентов MIB-деревьях.
2. Добавлены статистика и статусы бизнес-процессов по умолчанию для portType и фильтров.
3. Обработка широковещательных сообщений.
4. Маршрутизация по бизнес-параметрам ответа из внешней системы.
5. Перенаправление внешнего идентификатора вызова или сессии в систему аудита для анализа возможных проблем с внешними системами. HTTPServer шины RTSIB поддерживает HTTPHeader X-External-Session-Id.

## Изменения с версии 3.4 до 3.5

1. Автоматическое формирование HTTP Binding.
2. Автоматическое отображение используемых портов в корне MIB-дерева.
3. В SLES добавлен координатор транзакций.
4. В LW SA Container добавлен менеджер транзакций.
5. Использование двух DataSource к разным БД в рамках одной транзакции.

## Изменения с версии 3.5 до 3.6

1. Мониторинг распределенных приложений — система [Operations&Management](#).
2. Настройка трассировки сообщений в Service Gateway, SLES и LW SA Container.



## Источники информации

1. ATLAS. Приложение ATLAS MIB Explorer. Руководство администратора.
2. ATLAS. Общее описание.
3. ATOMS. Подсистема ATOMS System Info Suite. Руководство администратора.
4. ATOMS. Подсистема ATOMS EZ Statistics Suite. Руководство администратора.
5. Unica. Компонент Unica SLR. Руководство администратора.
6. Unica. Разработка услуг в Unica Visual Builder. Руководство по настройке.
7. Unica. Система Unica Visual Builder Light Web. Руководство администратора.
8. Unica. Компонент Unica SMPP Agent. Руководство администратора.
9. IN@Voice. BCSC Interface. Руководство администратора.
10. Unica. BFRS. Руководство администратора.
11. Platform v3. API-команды интерфейсов. Справочник.
12. Platform v3. MIB-параметры. Справочник.
13. Platform v3. API Gateway. Установка и настройка системы. Руководство администратора.
14. Unica Multilink. Схема Unica Multilink DB. Справочник.
15. Service Profile Management. Установка и настройка компонентов. Руководство администратора.
16. Correlation System. Руководство администратора.
17. Correlation System. Схема данных Correlation Management. Справочник.
18. Scheduler Subsystem. Установка и настройка системы. Руководство администратора.
19. RTSIB Audit. Установка и настройка системы. Руководство администратора.
20. SCS. Работа с графическим интерфейсом. Руководство пользователя.
21. <https://www.w3.org/TR/1999/REC-xpath-19991116/>

# История изменений

В разделе представлена история изменений документа.

## Изменения с версии 1.2 до 3.1

Версия документа 1:

1. Изменена структура документа.
2. Описаны новые функциональные возможности Platform v3: работа в транзакционном и сессионном режимах, корреляция.
3. Добавлены подразделы с описанием принципов работы системы Correlation System, планирования заданий, систем Session Management, Service Profile Subsystem, Subscriber Service Profile Subsystem и сервисных систем.
4. Добавлены разделы по установке, запуску и настройке систем: планирования заданий, корреляции, Session Management и аудита.
5. Добавлены описания различий установки, запуска и останова компонентов Platform v3 под управлением ОС Solaris и ОС Linux.
6. Добавлены разделы по настройке транзакционного и сессионного режима работы.
7. В раздел «Подключение SA с помощью Plv3 Management Console» добавлен шаг по установке параметров конфигурации Plv3 Management Console.
8. Дополнен раздел «Настройка источников данных на LW SA Container».
9. Раздел «Настройка Service Gateway» дополнен описанием настроек внешних систем, операций и условий для групп.
10. Добавлены разделы «Настройка Service Profile Subsystem» и «Настройка Subscriber Service Profile Subsystem».
11. Добавлены разделы «Настройка шины RTSIB», «Настройка SLES», «Балансировка нагрузки и резервирование шины RTSIB», «Массовые операции».
12. Переработан раздел «Условия применения» в части соответствия диаграмме развертывания.

## Изменения с версии 3.1 до 3.2

Версия документа 1:

1. Обновлено описание функциональных возможностей и интерфейса Plv3 Management Console.
2. Добавлены описания статистических переменных адаптивной очереди (8.11).
3. Изменено имя файла запуска для системы SLES (9.1).
4. Переработан раздел 9.2.
5. В разделах 9.2 и 10.2 изменено MIB-расположение развертываемых артефактов.
6. Добавлены разделы «Настройка параметров BP и SA», «[Настройка тайм-аутов активности реализации](#)», «[Настройка тайм-аутов ожидания ответа](#)», «[Настройка уровня логирования по бизнес-параметрам](#)» и «[Настройки программы очистки памяти](#)».
7. В настройки [внешних соединений](#) RTSIB добавлено описание переменной UseFastInfoSet.
8. Изменены переменные в разделе «[Настройка внешних соединений](#)».
9. В раздел [Настройка SLES](#) добавлено описание настроек ActivationManager.

## Изменения с версии 3.2 до 3.3

Версия документа 1:

1. Удалены упоминания о Plv3 Management Concole.

2. Изменена информация о настройке источников данных на LW SA Container: используются MIB-параметры, а не XML-файлы (10.3).
3. В раздел 10.3 добавлены рекомендации по обновлению драйверов СУБД перед настройкой источников данных.
4. Добавлена таблица соответствия параметров MIB и Tomcat DataSource (10.3.4).
5. В перечень СУБД, с которыми работает LW SA Container, добавлена СУБД Oracle timesten (10.3).
6. В раздел *Настройка JMS подключений СУБД Oracle* добавлены новые MIB-параметры. Удалена переменная *ServiceName*.
7. Добавлены разделы:
  - *Настройка ограничений трафика во внутренней сети;*
  - *Статистические параметры RTSIB;*
  - *Статистические параметры Service Gateway.*
  - *Настройка локальных соединений.*
  - *Настройка доступа к операциям по бизнес-параметрам.*
  - *История изменений системы.*
8. В раздел *Настройка параметров аутентификации и авторизации в профиле Service Gateway* добавлен блок «Настройка доступа к операциям по бизнес-параметрам».
9. Изменены названия MIB-групп:
  - @libs/LIB\_MessageBus\_vX.Y/DataServer/ProtocolLayers/TcpFi на @libs/LIB\_MessageBus\_vX.Y/DataServer/ProtocolLayers/Tcp (8.5.1).
  - @libs/LIB\_MessageBus\_vX.Y/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/TcpFi на @libs/LIB\_MessageBus\_vX.Y/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp (8.5.3).
10. Дополнено описание настроек Service Profile Subsystem.
11. Дополнен список MIB-настроек Subscriber Service Profile Subsystem (13.2).
12. В раздел «Настройка системы Correlation System» добавлено описание переменной *ForceOverwriteCorrelation*.
13. Дополнен список MIB-настроек Scheduler Subsystem, описаны статистические параметры SA Scheduler.
14. Изменена структура документа.

#### Версия документа 2:

1. Описана функциональная возможность «Кеширование ответов на внешние вызовы» (8.15).
2. Добавлен раздел «Транспортная статистика по бизнес-процессу».

#### Версия документа 3:

1. Запрос 221893. Добавлены диаграммы авторизации в режимах Internal, External и Off для Service Gateway (12.2.1.2).
2. Запрос 222138. Добавлено примечание о перенесении настроек артефактов при обновлении версий (7).
3. Запрос 221590. Добавлено описание команды *undeploy\_all* (9.2).
4. Добавлено описание опций '-s' и '-w' для команды *deployer.sh* (9.2).
5. Запрос 221872. Изменен формат переменной *ServerAddress* для групп @libs/LIB\_MessageBus\_vX.Y/ServiceServer/ProtocolLayers/Tcp, @libs/LIB\_MessageBus\_vX.Y/DataServer/ProtocolLayers/Tcp и @libs/LIB\_MessageBus\_vX.Y/HttpServer/ProtocolLayers/Tcp (8.5.1, 8.5.2).
6. Запрос 222187. Добавлено описание SA LDAP (17).

#### Версия документа 4:

1. Добавлено описание *SA SAL SMPP*.

2. Запрос 222187. Добавлен справочник MIB-переменных для SA SAL SMPP, SA LDAP и SA Multilink.
3. Добавлено описание [SA Multilink](#).

### Изменения с версии 3.3 до 3.4

#### Версия документа 1:

Запросы 221843, 212467 и 224282.

1. В раздел [Настройка SLES](#) добавлены описания групп:
  - CMP\_SLES\_vX.Y/BpelEngine/Configuration/ThreadResources
  - CMP\_SLES\_vX.Y/SlesManagement/Configuration
2. В раздел [Настройка SLES](#) добавлены режимы работы ActivationManager.
3. Добавлено [описание параметра BusinessProcessVersion](#) группы CMP\_SLES\_vX.Y/ActivationManager/Configuration/PortTypes/<имя portType>/Operations/<имя операции>/Filters/<имя фильтра>.
4. Добавлено описание статистики и статусов бизнес-процессов по умолчанию для portType и фильтров (раздел [Настройка SLES](#)).
5. Добавлены разделы:
  - [Версионность компонентов MIB-деревьях](#).
  - [Общие настройки](#).
  - [Обработка широковещательных сообщений](#).
6. В раздел [Настройка маршрутов](#) добавлено описание переменной ReservedType.
7. В раздел [Кеширование ответов на внешние вызовы](#) добавлено описание переменной MaxEstimatedCacheSize.
8. В [архитектуру](#) Platform v3 добавлена система SP.
9. В раздел [Варианты подключения компонентов](#) добавлен перечень используемых протоколов.
10. В раздел [Настройки программы очистки памяти](#) добавлены описания параметров:
  - TypeForceGC
  - WaitQueuesTimeout
  - TimeForceGC
11. В разделы [9.1](#), [12.1](#) и [10.1](#) добавлено описание, как задавать MIB-имя для компонента Platform v3.

#### Версия документа 2:

1. Запрос 220631. В разделы [Настройка транзакционности](#) и [Настройка сессионности](#) добавлены ссылки на статистику бизнес-процессов по транзакционности и сессионности.
2. Запрос 220631. Добавлен раздел [Статистика по развернутым бизнес-процессам](#).
3. Запрос 224282. Добавлен раздел [Маршрутизация по бизнес-параметрам ответа из внешней системы](#).
4. Пути @libs/LIB\_MessageBus\_vX.Y/... заменены на @libs/MessageBus/....

#### Версия документа 3:

1. Запрос 224835. В раздел [Версионность компонентов в MIB-деревьях](#) добавлено описание параметра atlas.removeCreatedBackup.
2. Запрос 224835. В разделы [9.1](#), [10.1](#) и [12.1](#) добавлено описание параметра atlas.removeCreatedBackup.
3. Запрос 224282. Добавлен раздел [Лицензирование шины RTSIB](#).
4. В раздел [Кеширование ответов на внешние вызовы](#) добавлены примечания о приоритете MIB-параметров для операции, portType и в корневой группы.

5. Подсистема корреляции переименована в систему корреляции.

#### Версия документа 4:

1. Система корреляции переименована в систему Correlation System.
2. Correlation Manager DB переименована в БД Correlation Manager.

#### Версия документа 5:

1. Подсистема Scheduler переименована в систему Scheduler Subsystem.
2. Удалены разделы по установке и настройке Correlation System, Service Profile Management и Scheduler Subsystem. По этим системам есть отдельные документы.
3. Запрос 225708. В раздел [Настройка JMS-подключений СУБД Oracle](#) добавлено описание MIB-параметра *KeepAliveTimeout*. Группа CMP\_LWSAContainer\_vX.Y/JMS/ConnectionsList/<имя источника данных>/ProtocolLayers/JMS.
4. Запросы 214807, 227563, 227899, 227596. Добавлен раздел [Настройка автоматического обновления интерфейсов](#).
5. Обновлены MIB-параметры группы CMP\_LWSAContainer\_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/ProtocolLayers/Jdbc. Добавлены параметры:
  - *AutoCommit*
  - *DumpSQLOnValidationFail*
  - *DumpStackOnGetConnection*
  - *EnableDebugLog*
  - *InitSql*

В группу CMP\_LWSAContainer\_vX.Y/DataSources/ConnectionsList/<jdbc-имя источника данных>/Statistics/StateInfo добавлены параметры:

- *RejectCounter*
  - *RejectRateAvg*
  - *RejectRateMax*
6. Запрос 221843. В раздел [Версионность компонентов в MIB-деревьях](#) добавлено примечание о флаге «Показывать системный MIB».
  7. Запрос 227547. Добавлен раздел [Совместимость версий веб-сервиса](#).
  8. Запрос 228028. В раздел [Настройка условий маршрутизации](#) добавлен пример настройки условия.
  9. Добавлены разделы [Статистические параметры JMS-соединения](#) и [Статистические параметры обработки транзакций](#).

#### Версия документа 6:

1. В разделе [Изменение адреса сервиса deployer](#) удалено примечание о необходимости перезапуска SLES.
2. В раздел [Статистические параметры RTSIB](#) добавлено описание статистики группы @libs/MessageBus/ExternalConnections/Statistics/StateInfo.
3. В раздел [Настройка параметров аутентификации и авторизации в MIB](#) добавлено описание MIB-переменной *ForwardControlHeaders* в группах:
  - CMP\_SG\_vX.Y/AA/Configuration/Systems/<имя системы>;
  - CMP\_SG\_vX.Y/AA/Configuration/Users/<имя пользователя>.
4. В раздел [Настройка адаптивных очередей](#) добавлено примечание, что адаптивные очереди использует только LW SA Container.
5. В раздел [Настройка параметров аутентификации и авторизации в MIB](#) добавлено примечание к описанию переменной *MessagesPerSecondType* в пункте 2.
6. В раздел [8.5.2](#) добавлено описание переменной *ResponseCode500onFault*.
7. Справочник параметров MIB вынесен в отдельный документ.

8. В раздел [Подключение SA](#) добавлено примечание о невозможности удаления SA и использование одного экземпляра для LW SA Container.
9. Добавлено приложение [Недопустимые символы в строковых данных](#).
10. Удалены разделы по установке и настройке системы RTSIB Audit. По системе есть отдельные документы.

Версия документа 7:

В раздел [Настройка JMS-подключений СУБД Oracle](#) добавлены описания параметров:

- *MessageSelector*
- *McAllowNewSubscribers*.

## Изменения с версии 3.4 до 3.5

Версия документа 1:

Запросы 221843, 212467 и 224282.

1. В раздел [Настройка SLES](#) добавлены описания групп:
  - *CMP\_SLES\_vX.Y/TransactionCoordinator/Configuration*
  - *CMP\_SLES\_vX.Y/SlesManagement/Configuration*
2. Добавлены разделы:
  - «Координатор транзакций»;
  - [Настройка тайм-аутов](#);
  - [HTTP Binding](#);
  - [Публикация и автоматическое формирование HTTP Binding](#);
  - [Менеджер транзакций](#).
3. В раздел [Настройка взаимодействия по HTTP-протоколу](#) добавлены описания MIB-параметров:
  - *OnOff*
  - *BaseUri*
  - *AllowedIPs*
  - *BannedIPs*
  - *InactivityTimeout*
  - *TransportLogLevel*.
4. В раздел [История изменений системы](#) добавлены изменения для версии 3.5.
5. В раздел [Настройка портов приложений или взаимодействия между компонентами](#) добавлено описание автоматического отображения номеров портов, которые использует компонент.
6. В разделе [Программное обеспечение](#) изменена версия Oracle — 13.0.2.
7. В раздел [Настройка источников данных](#) добавлено описание возможности использования двух DataSource к разным БД в рамках одной транзакции.
8. В раздел [Установка и запуск](#) добавлено примечание об упрощенной настройке java-опций.

Версия документа 2:

1. В раздел [Примеры WSDL-описаний](#):
  - добавлены примеры WSDL-описаний для *UrlReplacement*-сообщений и *HTTP GET UrlReplacement*-сообщений;
  - дополнены описания *UrlEncoded*-, *RPC*-сообщений и *UrlReplacement*-сообщений.
2. В раздел [Примеры сообщений](#) добавлены примеры *HTTP GET UrlReplacement*.
3. Раздел «Публикация и автоматическое формирование HTTP Binding» переименован в «REST-представление с использованием HTTP Binding» и переработан.

## Изменения с версии 3.5 до 3.6

### Версия документа 1:

Запросы 233965, 233964, 233973 и 247137.

#### 1. Добавлены разделы:

- Система *Operations&Management*.
- Миграция системы *ATLAS Agent 2.0*.
- Настройка трассировки сообщений.
- Управляющие системные SOAP-заголовки.
- Логирование.
- Система *Operations&Management*.
- Логирование.

2. В раздел *Шина RTSIB* добавлена информация об использовании параметра *SessionId*.

3. В группу *CMP\_SLES\_vX.Y/Scheduler/Configuration/Schedule* добавлены параметры:

- *BusinessApplication*
- *Sampling*
- *Tracing*

4. В раздел *История изменений системы* добавлены изменения для версии 3.6.

5. Удален раздел «Аудит» и упоминания об аудите в других разделах.

### Версия документа 2:

1. Раздел *Установка и настройка компонента traceDecoder* дополнен описанием настроек конфигурационного файла.

2. Добавлен раздел *Постобработка цепочки трейсов*.

3. Дополнен раздел *Общие сведения*.

4. Дополнено описание MIB-параметра *BusinessProcessInactivityTimeout (9.3)*.

5. Добавлено описание MIB-параметра *Maintenance (9.3)*.

6. Добавлен раздел *Сервисный режим*.

7. В раздел *Развертывание и переустановка BP и SE* добавлено примечание для команды *undeploy\_all* при работе в сервисном режиме.

### Версия документа 3:

1. Добавлен раздел *Сборщик мусора Garbage Collector G1*.

2. Добавлен раздел *Платформа Tarantool*.

### Версия документа 4:

1. Раздел *Автоматическое создание HTTP Binding для BP и SE* дополнен описанием особенностей работы настроек автоматического формирования HTTP Binding.

2. Раздел *Прием и отправка REST-сообщений* дополнен описанием опубликованных REST-операций.

3. Дополнен раздел *Публикация серверного интерфейса*.

4. Описание особенностей выполнения команды *undeploy\_all* перенесено из раздела *Развертывание и переустановка BP и SE* в раздел *Особенности выполнения команды undeploy\_all*.

5. В разделе *Установка и настройка компонента traceDecoder* актуализирован список параметров конфигурационного файла.

6. В раздел *8.5.2* добавлено примечание о псевдонимах для URI в HTTP-сервере.

7. Раздел «Примеры WSDL-описаний» переименован в *Структуры данных*. В этот раздел добавлены дочерние разделы.

8. В раздел *REST-представление с использованием HTTP Binding* добавлены способы передачи параметров.

9. В раздел [Примеры сообщений](#) добавлены примеры:

- [UrlEncoded+UrlReplacement](#);
- [JSON с числовыми именами элементов](#);
- [JSON+UrlReplacement](#);
- [HTTP EMPTY RESPONSE](#).

10. В раздел [Публикация серверного интерфейса](#) добавлен пример XSD-описания.

Версия документа 5:

1. В раздел [Архитектура](#) добавлена информация о реализации [OpenAPI](#) для описания REST API в RTSIB.
2. Добавлен раздел [REST-представление с использованием OpenAPI](#) и подразделы:
  - 2.1. [Интеграция RTSIB C++ SDK с OpenAPI Repository](#).
  - 2.2. [Настройки OpenAPI Repository](#).
  - 2.3. [Описание протокола для взаимодействия с RTSIB](#).

Версия документа 6:

1. В раздел [Настройка маршрутов](#) добавлено описание MIB-параметра [ConnectAllReservedIfAllMainLeft](#).
2. В разделе [Настройка условий маршрутизации](#) в примечании к пункту 4 количество условий изменено на '48'.
3. В разделе [Установка и настройка компонента traceDecoder](#) в конфигурационный файл добавлен параметр [Topics/format](#).
4. Добавлен раздел [Настройка UserPrincipal и SystemPrincipal](#).

Версия документа 7:

1. В раздел [Настройка JMS-подключений СУБД Oracle](#) добавлено примечание для параметра [DataSourceName](#).
2. Добавлены следующие разделы:
  - [Алгоритм работы SA с очередью нотификаций](#);
  - [Нотификационная схема БД](#).
3. В раздел [Tarantool RTSIB Driver](#) добавлен подраздел **Доступ к HTTP Headers**.

Версия документа 8:

1. В разделе [Установка и запуск](#) обновлена инструкция для задания имени SLES.
2. В разделе [Развертывание и переустановка BP и SE](#) добавлена опция запуска `-f`.
3. В разделе [Настройка SLES](#) дополнено описание MIB-параметра [ProcessPerThread](#).
4. Из раздела [Настройка TransportEngine](#) удалены параметры [QueryRateAvg](#), [QueryRateMax](#) и [QueryCounter](#).
5. В раздел [Запись логов SA](#) добавлено описание MIB-параметра [Type](#).
6. В разделе [Настройка взаимодействия по проприетарному протоколу](#) добавлено описание параметров [AddSourceAddressToContainer](#), [AllowedIPs](#), [BannedIPs](#) и [TransportLogLevel](#).
7. В разделе [Настройка взаимодействия по HTTP-протоколу](#) добавлено описание параметров [AddSourceAddressToContainer](#), [KeyStorePassword](#) и [KeyStorePath](#).

Версия документа 9:

1. Добавлен раздел [Настройка Java-опций](#).
2. Добавлен раздел [Особенности работы ActivationManager](#).
3. Изменен текст раздела [Настройка SLES](#): в разделе оставлены общие настройки. Настройки по подгруппам перенесены в соответствующие разделы:
  - [Настройка ActivationManager](#);



- [Настройка BPELEngine](#);
- [Настройка Scheduler](#);
- [Настройка SeEngine](#);
- [Настройка SlesManagement](#).

В общих настройках из раздела CMP\_SLES\_vX.Y/Core/Configuration удален параметр *UseTargetNamespaceInURI*.

4. Добавлен раздел [Синтаксис, используемый при создании условий](#).
5. Раздел «Координатор транзакций» удален, текст перенесен в раздел [Настройка TransactionCoordinator](#).
6. Раздел «Транспортная статистика по бизнес-процессу» удален, текст перенесен в раздел [Настройка TransportEngine](#).
7. В раздел [Настройка маршрутов](#) добавлено примечание об операции с именем 'default'.
8. Добавлен раздел [Порядок выбора маршрута для разных типов вызова](#).
9. В раздел [Настройка взаимодействия по проприетарному протоколу](#) добавлено описание MIB-параметра *OnOff*.

#### Версия документа 10:

1. В раздел [Настройка трассировки сообщений](#) добавлено описание MIB-параметра *Propagate*.
2. Добавлен раздел [Настройка авторизации в Kafka](#).
3. В раздел [Tarantool RTSIB Driver](#) добавлен пример XSD-описания для доступа к HTTP Headers.