

Руководство пользователя VI.Qube METACONTROL

Компонент разработан ООО «БИАЙ КУБ»

Приказ № _____ от «__» _____ 20__ г.

Акт сдачи № _____ от «__» _____ 20__ г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛОССАРИЙ.....	3
1 ЦЕЛИ И НАЗНАЧЕНИЕ METACONTROL	4
2 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ METACONTROL	4
2.1 Структура MetaControl	4
2.2 Преимущества использования компонента MetaControl.....	5
3 ТАБЛИЦЫ METACONTROL	5
3.1 Настраиваемые таблицы MetaControl.....	7
3.2 Таблица логов MetaControl	8
3.3 Таблицы дескрипторы MetaControl	9
3.4 Процедуры MetaControl	10
3.4.1 Назначение процедур и функций MetaControl	11
3.4.2 Процесс работы процедур	11
4 СЦЕНАРИЙ РАБОТЫ С METACONTROL	12

ВВЕДЕНИЕ

MetaControl – компонент, предназначенный для информирования пользователей путем рассылки уведомлений о произведенных контролях. Компонент имеет возможность настройки различных видов правил информация о выполнении которых в различных представления направляется пользователям.

В документе приведено описание компонента и принципы работы с ним. Рассмотрены примеры работы с правилами и отправление сообщений.

Изучение данного документа позволит понять принцип работы компонента.

ГЛОССАРИЙ

1.	MetaControl BI.Qube	-	Инструмент, предназначенный для информирования пользователей путем рассылки уведомлений о произведенных контролях.
2.	MetaStaging BI.Qube	-	Инструмент, предназначенный для транспортировки данных.
3.	MetaOrchestrator BI.Qube	-	Инструмент управления процессами.
4.	Датасет		Обработанный и структурированный массив данных
5.	Процедура		Объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере
6.	Представление		Виртуальная таблица, содержимое которой определяется запросом
7.	Функция		Подпрограмма, которая принимает параметры, выполняет действие, например сложное вычисление, и возвращает результат этого действия в виде значения. Возвращаемое значение может быть либо единичным скалярным значением, либо результирующим набором.
8.	Постконтроль		Набор функций, процедур или представлений, которые вызываются после выполнения ETL процесса.

9.	smtp-сервер	Сетевой протокол, который отвечает за отправку писем
----	-------------	--

1 ЦЕЛИ И НАЗНАЧЕНИЕ METACONTROL

Цель компонента MetaControl заключается в своевременном уведомлении пользователей о статусе событий, происходящих с данными на основе правил, создаваемых пользователями.

Уведомления MetaControl отправляются путем рассылки о произведенных контролях с использованием почтового сервера и telegram-канала.

Компонент MetaControl входит в состав системы VI.Qube и может эксплуатироваться как отдельный компонент, так и в составе системы, так и под управлением компонента MetaOrchestrator, в такой конфигурации использование компонента является наиболее эффективной.

2 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ METACONTROL

2.1 Структура MetaControl

Для работы MetaControl необходимо создать правило контроля данных в одном из следующих форматов:

- Процедура (P);
- Функция (F);
- Представление (V).

На основе полученных данных будет произведено формирование датасета – тела сообщения о произведенном контроле. После чего с настроенного smtp-сервера произойдет рассылка писем на указанные адреса, в необходимом формате.

Ряд форматов, поддерживаемых компонентом:

- HTML – вся информация будет содержаться в самом письме;
- PDF – набор данных будет загружен в pdf формате и прикреплен к сообщению в telegram-канале или почтовому сообщению.

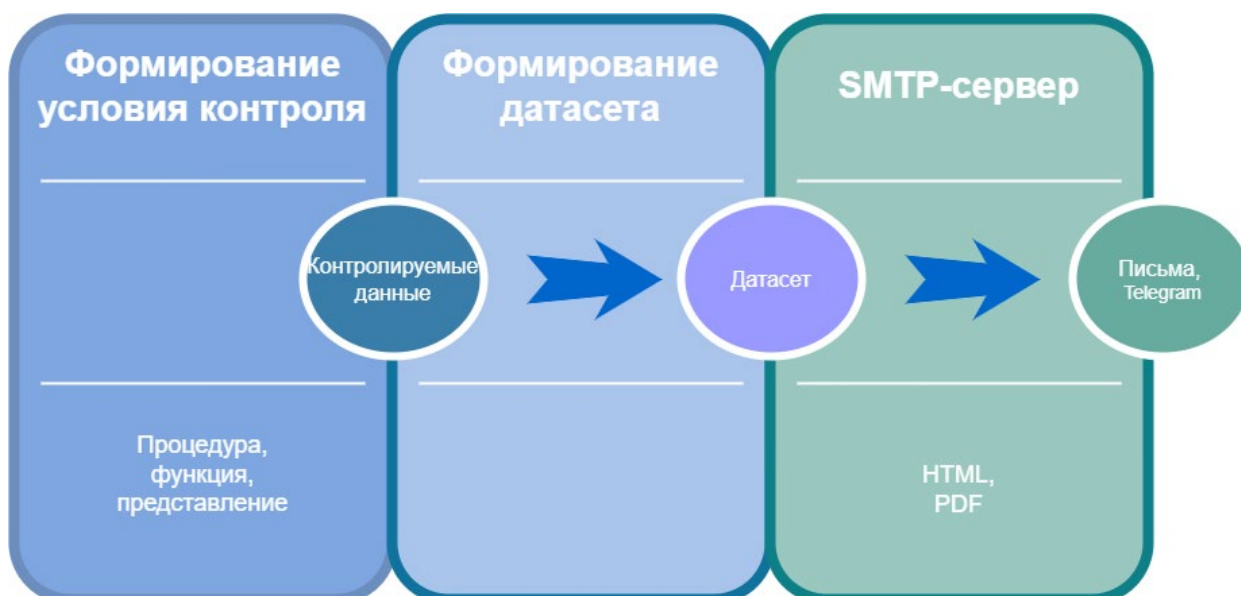


Рисунок 1. Структура MetaControl

2.2 Преимущества использования компонента MetaControl

1. MetaControl имеет умное формирование писем, поэтому несколько контролей он объединит в одно письмо и отправит на указанные адреса.

Например, есть контроли 1р, 2р, 3р с соответствующими группами адресов 1m, 2m, 3m. В группах находятся пересекающиеся адреса:

1р рассылка на 1m = (mail_1@sample.com)

2р рассылка на 2m = (mail_1@sample.com, mail_2@sample.com)

3р рассылка на 3m = (mail_1@sample.com, mail_2@sample.com, mail_3@sample.com).

2. MetaControl определит какие контроли необходимо разослать на адреса, после чего сгруппирует контроли по конечным адресам, тем самым каждый пользователь получит лишь ОДНО письмо содержащее все контроли, в которые попадает его адрес:

mail_1@sample.com – Письмо с контролями: 1р, 2р, 3р

mail_2@sample.com – Письмо с контролями: 2р, 3р

mail_3@sample.com – Письмо с контролями: 3р

3. MetaControl имеет возможность задавать параметры в источниках данных, что добавляет динамики в выполнение постконтролей.

3 ТАБЛИЦЫ METACONTROL

Таблицы MetaControl делятся на три категории:

Настроечные таблицы – данные вносит пользователь в соответствии с правилами заполнения.

Таблицы логов (заполняются автоматически) – данные вносит система по итогам выполнения очередной сессии.

Таблицы дескрипторы (заполнены разработчиком) – данные внесены предварительно в соответствии с реализованным функционалом.

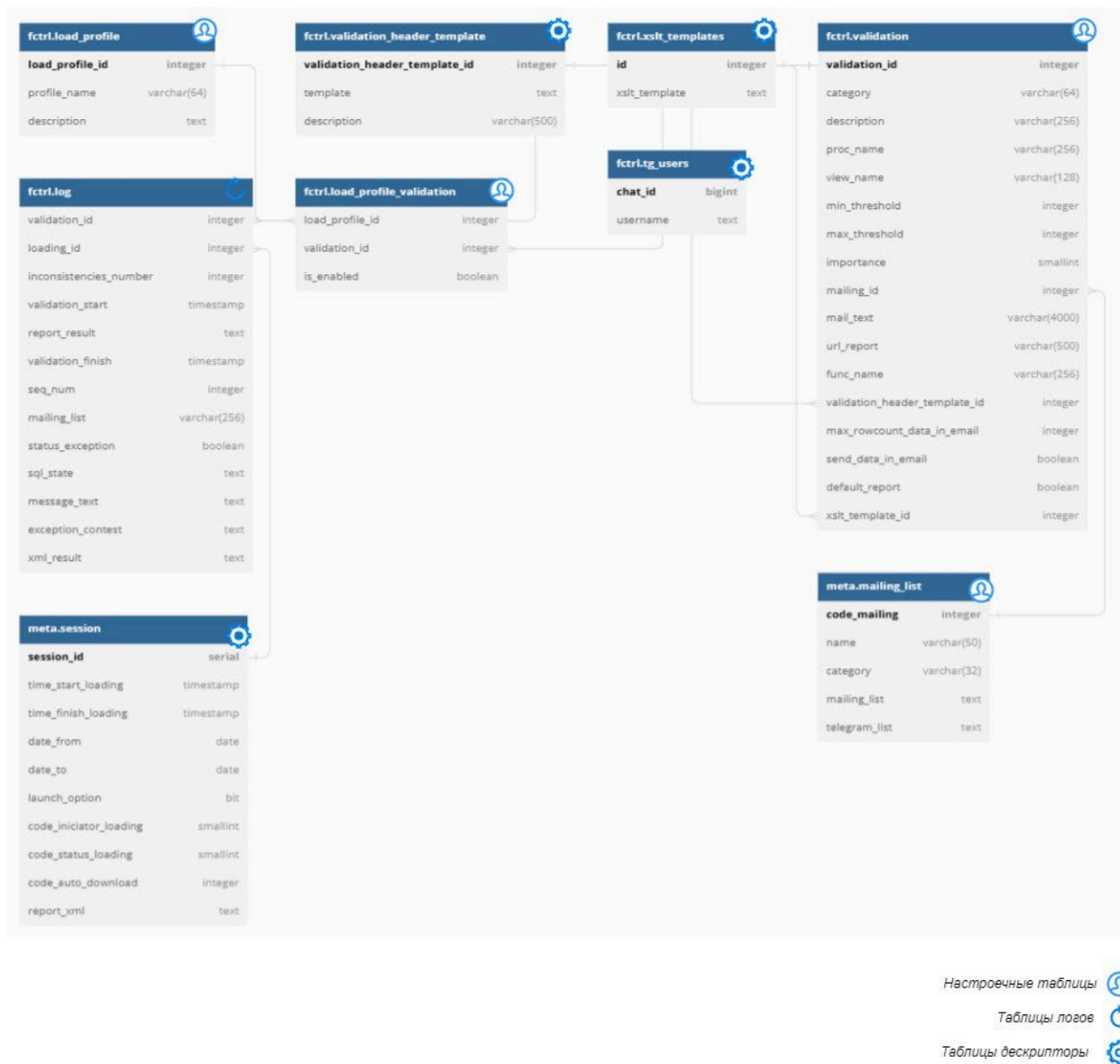


Рисунок 2. Структура таблиц MetaControl

3.1 Настроечные таблицы MetaControl

1. «fctrl.validation» – таблица, содержащая список контролей, их название, описание.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
validation_id	int	Автоинкремент	Уникальный ID проверки
category	text	Вручную	Категория проверки. Контроли группируются в письмах в рамках одной категории.
description	text	Вручную	Описание проверки
proc_name	text	Вручную	Наименование процедуры для производства контроля
view_name	text	Вручную	Наименование представления для производства контроля
func_name	text	Вручную	Наименование функции для производства контроля
min_threshold, max_threshold	int	Вручную	Границы допустимого кол-ва строк (если число записей в таблице или процедуре не входит в диапазон [min; max] то происходит рассылка
importance	boolean	Вручную	Степень важности контроля (пометка «важно» в письме
mailing_id	int	Вручную	ID из meta.mailing_list(email/telegram тех, кому нужна рассылка)
mail_text	text	Вручную	Дополнительный текст письма
url_report	text	Вручную	URL на отчет (если есть) по проверке
send_data_in_email	boolean	Вручную	Вкл/откл отправки всех данных из проверки
max_rowcount_data_in_email	int	Вручную	Ограничение максимального кол-ва строк данных в письме
validation_header_template_id	int	Вручную	ID шаблона из fctrl.validation_header_template
default_report	boolean	Вручную	Пометка стандартного отчета.
xslt_template_id	int	Вручную	Позволяет задавать формат xslt шаблона, что обеспечивает определенный шаблон отображения данных у пользователя. ID из таблицы fctrl.xslt template

2. «meta.mailing_list» – справочник адресов для рассылки.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
code mailing	int	Автоинкремент	Уникальный ID рассылки
name	text	Вручную	Наименование набора адресов
category	text	Вручную	Категория рассылки
mailing_list	text	Вручную	Список адресов для рассылки. В один список может входить множество адресов, задаваемых через запятую. Например: sample@it.ru, exmpl@it.ru
telegram_list	text	Вручную	Список telegram-аккаунтов для рассылки. В один список может входить множество адресов, задаваемых через запятую. Например: exmpl, testexmpl

3. «fctrl.load_profile_validation» – связующая таблица, в которой настраиваются отношения профиль-контроль, также имеет возможность отключения контроля.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
load_profile_id	int	Вручную	Уникальный ID профиля
validation_id	int	Вручную	Код контроля
is_enabled	boolean	Вручную	Вкл/откл проверки из постконтроля

4. «fctrl.load_profile» – таблица с описанием профилей.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
load_profile_id	int	Автоинкремент	Уникальный ID профиля
profile_name	text	Вручную	Наименование профиля
description	text	Вручную	Описание профиля

3.2 Таблица логов MetaControl

«fctrl.log» – таблица, определяющая функциональные особенности работы компонента.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
validation_id	int	Автоматически	Уникальный ID проверки
session_id	int	Автоматически	Код загрузки

inconsistencies_number	int	Автоматически	Количество расхождений данных на хранилище с эталоном
validation_start	timestamp	Автоматически	Время начала контроля
report_result	text	Автоматически	Ссылка на отчет
validation_finish	timestamp	Автоматически	Время окончания контроля
seq_num	int	Автоматически	Номер последовательности контроля (актуален при наличии нескольких контролей)
mailing_list	int	Автоматически	ID в таблице fctrl.mailing_list, которая содержит либо список email-адресов или telegram-аккаунты.
status_exception	boolean	Автоматически	Отчет об ошибке.
sql_state	text	Автоматически	Номер ошибки. Заполняется в случае значения поля «status_exception» «true».
message_text	text	Автоматически	Описание ошибки. Заполняется в случае значения поля «status_exception» «true».
exception_context	text	Автоматически	Контекст кода. Заполняется в случае значения поля «status_exception» «true».
xml_result	text	Автоматически	XML-код, в который оборачивается набор данных

3.3 Таблицы дескрипторы MetaControl

1. «meta.session» – таблица с описанием сессии загрузки.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
session_id	int	Автоинкремент	Уникальный ID загрузки
time_start_loading	timestamp	Автоматически	Время начала загрузки
time_finish_loading	timestamp	Автоматически	Время окончания загрузки
date_from	timestamp	Автоматически	Период, за который ETL процесс загружает данные («дата с»)
date_to	timestamp	Автоматически	Период, за который ETL процесс загружает данные («дата по»)
launch_option	bit	Автоматически	Статус действия загрузки
code_initiator_loading	smallint	Автоматически	Показатель кем была инициирована загрузка(Вручную/автоматически)

code_status_loading	smallint	Автоматически	Статус загрузки
code_auto_download	int	Автоматически	Если загрузка была инициирована автоматически, показывает её статус(ETL/по расписанию)
report_xml	text	Автоматически	XML-Отчет

2. «fctrl.validation_header_template» – таблица, содержащая шаблон для настройки вывода полей при выполнении контроля.

Поля таблицы:

Имя столбца	Тип данных	Источник	Назначение
validation_header_template_id	int	Автоинкремент	Уникальный ID шаблона
template	text	Вручную	Шаблон (html формат)
description	text	Вручную	Описание шаблона

3. «fctrl.xslt_templates» - таблица, содержащая XSLT-шаблоны для формирования html-результата.

Имя столбца	Тип данных	Источник	Назначение
xslt_templates_id	int	Автоинкремент	Уникальный ID XSLT-шаблона
xslt_templates	text	Вручную	XSLT-шаблон

4. «fctrl.tg_users» - таблица, содержащая id и username пользователей, которые инициировали диалог с telegram-ботом

Имя столбца	Тип данных	Источник	Назначение
chat_id	int	Автоматически	ID чата
username	text	Автоматически	Логин пользователя в Telegram

3.4 Процедуры MetaControl

Основными программными продуктами являются:

1. процедуры
 - a. fctrl.init_validation,
2. функции
 - a. fctrl.recive_mailing_list,
 - b. fctrl.recive_mailing_content
 - c. fctrl.recive_telegram_list.
 - d. fctrl.get_data_as_xml

3.4.1 Назначение процедур и функций MetaControl

- Процедура «**fctrl.init_validation**» – получает информацию из настроечной таблицы, вызывает определённые представления функции и процедуры и записывает эти данные в таблицу логов.
- Функция «**fctrl.recive_mailing_content**» – позволяет получить содержание писем для определенных адресов отправки по коду загрузки
- Функция «**fctrl.recive_mailing_list**» – позволяет получить список email-адресов, принадлежащие определенному коду загрузки.
- Функция «**meta.current_session_code**» позволяющая получить текущий код загрузки для записи этой информации в таблицу логов
- Функция «**meta.etl_is_launched**» – проверяет статус etl-процесса.
- Функция «**meta.sat_parameters**» – функция, которая вставляет параметры в таблицу «**meta.params**».
- Функция «**meta.split**» – служит разделителем для данных поля «**mailing_list**» таблицы «**mailing_list**».
- Процедура «**fctrl.get_data_as_xml**» – преобразует в xml-формат набор данных, полученных при выполнении контроля.

3.4.2 Процесс работы процедур

Процедура «**fctrl.init_validation**» производит обработку записей «**fctrl.validation**» после чего добавляет их результат в таблицу «**fctrl.Log**».

Приложение определяет количество записей попавших в «**fctrl.Log**», в случае ненулевого результата запускается C# скрипт формирования писем путем извлечения данных из «**fctrl.Log**».

Готовые письма рассылаются на указанные для них адреса.

4 СЦЕНАРИЙ РАБОТЫ С METACONTROL

1. Выполнить вход на удаленный рабочий стол и реализовать запуск компонента на Linux, запустить инструмент администрирования базы данных (в данном случае предлагается использовать DBeaver).

2. Выбрать сервер «Metacontrol – Rosreestr»

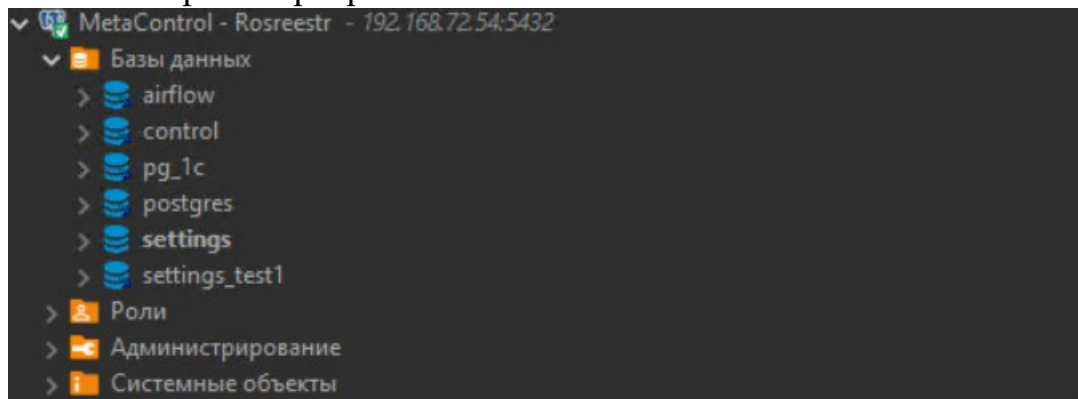


Рисунок 3. Схема «Metacontrol – Rosreestr»

3. Открыть базу данных «settings»

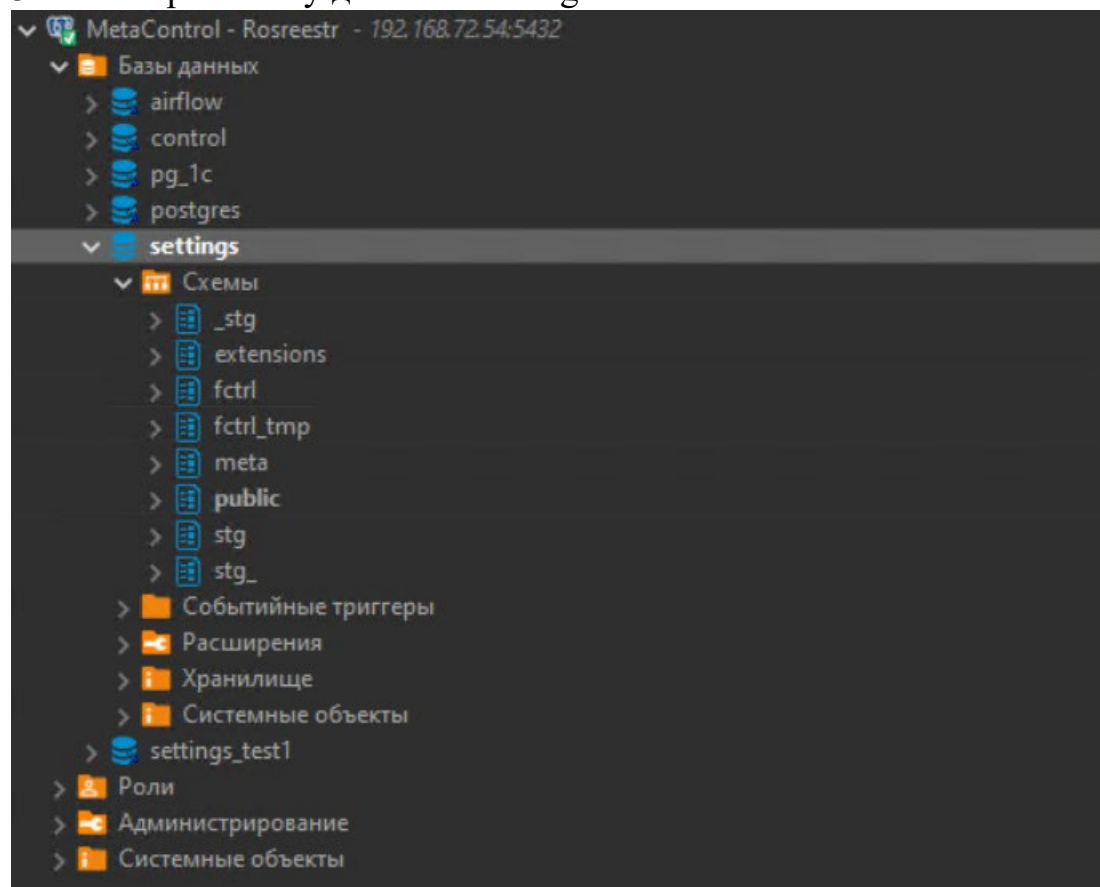


Рисунок 4. База данных «settings»

4. Компонент «Metacontrol» состоит из двух схем: «fctrl» и «meta». Наименования таблиц, относящихся к этим схемам содержат соответствующий префикс (например, таблица «meta.mailing_list» входит в схему «meta»).

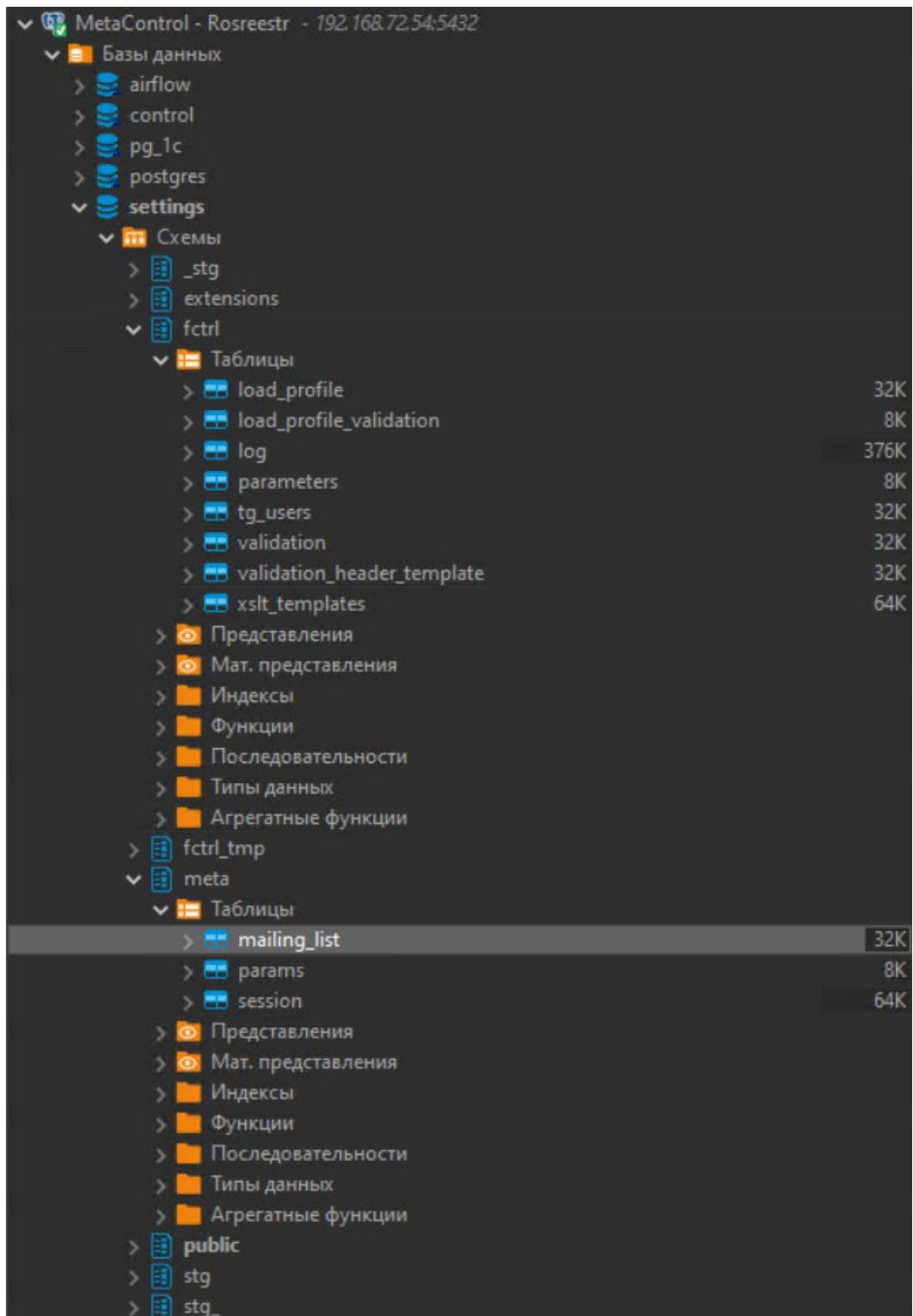


Рисунок 5. Структура схем метакомпонента

5. В схеме «fctrl», в таблице «fctrl.load_profile» необходимо занести данные о созданном профиле, а именно «profile_name» (имя профиля), «description» (описание профиля).

	load_profile_id	profile_name	description
1	1	ETL	Профиль основного ETL
2	2	ETL.НУ	Профиль ETL налогового учета
3	3	ETL.Логистика	Профиль ETL для отслеживания логистики

Рисунок 6. Таблица «fctrl.load_profile»

6. В схеме «meta», в таблице «meta.mailing_list», необходимо указать наименование «name», категорию «category», почту для рассылки «mailing_list», имя пользователя (login, учетная запись) в Telegram «telegram_list».

	code_mailing	name	category	mailing_list	telegram_list
1	1	Набор тестовых адресов разработчика#1	Developer's address	frantsev@biqube.ru, frantsev@biqube.ru	frantsev
2	2	Набор тестовых адресов разработчика#2	Developer's address	frantsev@biqube.ru, frantsev@biqube.ru	frantsev
3	3	Набор тестовых адресов разработчика#3	Developer's address	frantsev@biqube.ru, frantsev@biqube.ru	frantsev
4	4	uiedghrfiguerg	egrgere	FrantsevSD@biqube.ru	frantsev
5	5	test	egrgere	frantsev@biqube.ru	telegram#8

Рисунок 7. Таблица «meta.mailing_list»

7. Для осуществления рассылки через Telegram необходимо перейти по ссылке <https://t.me/MetaControlComponentTGBot> и с указанного в поле «telegram_list» аккаунта отправить сообщение с текстом «/start».

8. В схеме «fctrl», в таблице «fctrl.validation», в колонке «category» необходимо задать наименование контроля (по этому полю происходит группировка писем), в колонке «description» необходимо задать описание.

	validation_id	category	description	proc_name	view_name	min_threshold
1	1	Отчет о загрузке	Контроль о последних строках загрузки	[NULL]	fctrl.proverka	
2	2	Отчет о загрузке	Детальное логирование последней загрузки	[NULL]	stg_log_detail_last	
3	3	Отчет о загрузке	Обобщенное логирование последней загрузки	[NULL]	stg_log_summary_last	
4	4	RorsReestrTest	Тестирование функционала для росреestra	[NULL]	stg_log_detail where "Имя таблицы" like "%public.kinopoisk%"	

Рисунок 8. Таблица «fctrl.validation»

9. В той же таблице необходимо задать алгоритм выполнения контроля. Алгоритм выполнения контроля может быть задан одним из трех способов: представлением (view_name), процедурой (proc_name), функцией (func_name) – в зависимости от задачи, которую решает контроль.

Ниже приведен пример контроля, заданного с помощью представления. В поле «view_name» задается наименование представления для выполнения контроля (см пример ниже).

	description	proc_name	view_name	min_threshold	max_threshold	importance	mailing_id
1	контроль о последних строках загрузки	[NULL]	fctrl.proverka	0	0	1	3
2	детальное логирование последней загрузки	[NULL]	stg_log_detail_last	0	0	1	3
3	обобщенное логирование последней загрузки	[NULL]	stg_log_summary_last	0	0	1	3
4	тестирование функционала для росреestra	[NULL]	stg_log_detail where "Имя таблицы" like "%public.kinopoisk%"	0	0	1	5

Рисунок 9. Кейс с описанием примера контроля, заданного с помощью представления

В данном случае, для рассылки будет использоваться представление «stg_log_detail where "Имя таблицы" like "%public.kinopoisk%"» (где where "Имя таблицы" like "%public.kinopoisk%" - условие для фильтрации запроса). Это представление выводит данные соответствующей таблицы по написанному для него запросу:


```

select sc.session_id as "Код сессии",
sc.destination_object_with_parameters as "Имя таблицы",
sum(scs.processed_rows)
filter (where scs.message_source::text = 'Reader: Fill Buffer'::text) as
"Обработано строк",
sum(scs.processed_bytes)
filter (where scs.message_source::text = 'Reader: Fill Buffer'::text) as
"Обработано байт",
case
when count(*) filter (where scs.status::text = 'error'::text) = 0 then true
else false
end as "Статус выполнения",
string_agg(scs.error_message, '||| '::text)
filter (where scs.status::text = 'error'::text) as "Информация об ошибках",
string_agg(scs.command_with_parameters, '||| '::text)
filter (where scs.status::text = 'error'::text) as "Запрос к БД",
st.name as "Источник"
from stg_session_command_statistic scs
join stg_session_command sc on sc.session_command_id =
scs.session_command_id
join stg_command c on c.command_id = sc.command_id
join stg_source s on s.source_id = c.source_id
join stg_storage_type st on st.storage_type_id = s.storage_type_id
group by sc.session_id, sc.destination_object_with_parameters, st.name;

```

10. В поле «mailing_id» необходимо внести id поля таблицы «meta.mailing_list», в котором указаны нужные адреса для рассылки (для данной задачи «4»).

	123 min_threshold	123 max_threshold	123 importance	123 mailing_id	asc mail_text	asc url
1	0	0	1	3	В ходе выполнения ETL загрузились эти строки	[NULL]
2	0	0	1	3	Отчет об ошибках/успешного выполнения metastaging	[NULL]
3	0	0	1	3	LAZYTEXT	[NULL]
4	0	0	1	4	Тестирование расширенного описания MetaControl для	[NULL]

Рисунок 10. Данные поля «mailing_id» для таблицы «fctrl.validation»

11. Дополнительное описание, которое должно приходить в письме необходимо указать в поле «mail_text».

	importance	123 mailing_id	asc mail_text	asc url_report	asc func_name	123 validation_header_te
1	1	3	В ходе выполнения ETL загрузились эти строки	[NULL]	[NULL]	
2	1	3	Отчет об ошибках/успешного выполнения metastaging	[NULL]	[NULL]	
3	1	3	LAZYTEXT	[NULL]	[NULL]	
4	1	4	Тестирование расширенного описания MetaControl для	[NULL]	[NULL]	

Рисунок 11. Описание к тексту рассылки

12. Следующим шагом важно заполнить поле «validation_header_template_id» из таблицы

«fctrl.validation_header_template», в которой содержится шаблон общей сводки производительности контроля (для данной задачи «2»).

	url_report	func_name	validation_header_template_id	max_rowcount_data_in_email
1	TL загрузились эти строки	[NULL]	3	20
2	спешного выполнения metastaging	[NULL]	3	1 000
3	[NULL]	[NULL]	3	1 000
4	ренного описания MetaControl для	[NULL]	2	20

Рисунок 12. Таблица «fctrl.validation_header_template»

13. В схеме «fctrl», в таблице «fctrl.load_profile_validation» необходимо добавить связь между профилем (таблица «fctrl.load_profile») и контролем («fctrl.validation»), а именно заполнить «load_profile_id» (id профиля), «validation_id» (id контроля) и активировать эту связь в колонке «is_enabled» (задать значение «true»).

	load_profile_id	validation_id	is_enabled
1	2	1	[v]
2	1	2	[v]
3	1	3	[v]
4	2	4	[v]

14. Для запуска процесса интеграции данных нужно вызвать виртуальную машину linux (см. п. *Запуск компонента на linux*), в появившемся окне необходимо ввести **dotnet скрипты** для вызова компонента:

1. `cd MetaControl`
2. `Dotnet MetaControl.dll`

```

itpro_admin@biqube-etl-01: ~/MetaControl
itpro_admin@biqube-etl-01:~$ cd MetaControl
itpro_admin@biqube-etl-01:~/MetaControl$ dotnet MetaControl.dll
  
```

Рисунок 13. Скрипты для вызова компонента

15. В результате должна появиться статистика выполнения контролей.


```

itpro_admin@biqube-etl-01: ~/MetaControl
itpro_admin@biqube-etl-01:~/MetaControl$ cd MetaControl
itpro_admin@biqube-etl-01:~/MetaControl$ dotnet MetaControl.dll
Письмо отправлено: lupookaa@biqube.ru
Письмо отправлено: FrantsevSD@biqube.ru
Письмо отправлено: FrantsevSD@biqube.ru
Письмо отправлено: subbotinvv@biqube.ru
Запущен бот MetaControl

Контроль Отчет о загрузке.pdf преобразован в pdf файл. Для пользователя flame_br
k
Контроль Отчет о загрузке.pdf (Отчет о загрузке) отправлен пользователю flame_br
k (1589001083)

Контроль Отчет о загрузке.pdf преобразован в pdf файл. Для пользователя VladSub3
2
Контроль Отчет о загрузке.pdf (Отчет о загрузке) отправлен пользователю VladSub3
2 (286242280)

Контроль Отчет о загрузке.pdf преобразован в pdf файл. Для пользователя flxnbz
Контроль Отчет о загрузке.pdf (Отчет о загрузке) отправлен пользователю flxnbz (
225714360)

Контроль RorsReestrTest.pdf преобразован в pdf файл. Для пользователя flxnbz
Контроль RorsReestrTest.pdf (RorsReestrTest) отправлен пользователю flxnbz (2257

```

Рисунок 14.Статистика выполнения контролей

При этом, на указанные почту и Telegram придет рассылка заданного формата, как представлено на рисунке ниже.

RorsReestrTest

Кому: [аватар]

Перевести сообщение на русский | Никогда не переводить с этого языка: Английский | Параметры перевода

Категория: RorsReestrTest (Тестирование функционала для росреестра) 1

Просмотр отчета:

stg_log_detail 2

	Код сессии	Имя таблицы	Обработано строк	Обработано байт	Информация об ошибках	Запрос к БД	Источник
✘	9	public.kinopoisk_rus_review	26		Cannot write a value of CLR type 'System.Byte' as database type 'text'.	SELECT * FROM dwh.t_TaskStatuses	postgresql
✔	7	public.kinopoisk_rus_review	9	0			postgresql
✔	8	public.kinopoisk_rus_review	12	0			postgresql
✔	11	public.kinopoisk_rus_review	2				postgresql
✔	12	public.kinopoisk_rus_review	2				postgresql
✔	13	public.kinopoisk_rus_review	35				postgresql
✔	6	snapshot/public.kinopoisk_rus_review.parquet					postgresql

Рисунок 15.Формат рассылки

В данном случае на рисунке под номером 1 приведена сводка письма по настроенному шаблону. Под номером 2 приведен набор данных, которые выводит представление stg_log_detail, преобразованное в html – код.