

УДК: 004.624

АВТОМАТИЗАЦИЯ СИСТЕМЫ ОБМЕНА ДАННЫМИ ЧЕРЕЗ БРОКЕР СООБЩЕНИЙ RABBITMQ В ПРОГРАММЕ УЧЕТА «МАТРИЦА. КРОЛИКОВОДСТВО»

Неговора Д. О.¹

¹НИУ БелГУ – Белгородский государственный национальный исследовательский университет, Россия, Белгород. e-mail: negovoradiana@gmail.com

Статья посвящена разработке внешней обработки для обмена сообщениями через брокер RabbitMQ на базе отраслевого решения «Матрица. Кролиководство»; рассмотрены основные понятия, такие как протокол обмена данными AMQP, брокер сообщений, файл формата xml, особенности их связи между собой с учетом программной реализации. Освещены требования к разрабатываемой обработке со стороны действующей на платформе 1С конфигурации «Матрица. Кролиководство». На основании требований спроектирован и описан интерфейс решения, обоснован принятый к реализации функционал, его связь с пользовательским интерфейсом, а также приведены скриншоты результатов работы. Листинг основных модулей с комментариями также приведен в основной части статьи и отражает отличительные особенности программного решения для данной задачи. Помимо этого приведены результаты тестирования функционирования модуля обмена (обработки) в пользовательском режиме, положительный результат отправки сообщения в очередь для обмена подтвержден на стороне брокера сообщений RabbitMQ при помощи менеджера очередей RabbitMQ Management. Сделаны выводы о дальнейших путях совершенствования работы внешней компоненты, выявлены актуальные пути развития реализованной технологии для существующей конфигурации в целом. Ключевые слова: обмен данными, протокол AMQP, брокер сообщений, внешняя обработка, 1С, RabbitMQ

AUTOMATION OF THE DATA EXCHANGE SYSTEM THROUGH THE RABBITMQ MESSAGE BROKER IN THE "MATRIX. RABBIT BREEDING "

Negovora D.O.¹

¹NRU BelSU – Belgorod State National Research University, Russia, Belgorod, e-mail: negovoradiana@gmail.com

The article is devoted to the development of external processing for messaging through the RabbitMQ broker based on the industry solution "Matrix. Rabbit breeding "; the basic concepts, such as the AMQP data exchange protocol, message broker, xml format file, the peculiarities of their connection with each other, taking into account the software implementation, are considered. The requirements for the developed processing from the side of the "Matrix. Rabbit breeding ". Based on the requirements, the solution interface was designed and described, the functionality adopted for implementation was substantiated, its connection with the user interface, as well as screenshots of the work results. Listing of the main modules with comments is also given in the main part of the article and reflects the distinctive features of the software solution for this problem. In addition, the results of testing the functioning of the exchange (processing) module in the user mode are given; the positive result of sending a message to the exchange queue was confirmed on the side of the RabbitMQ message broker using the RabbitMQ Management queue manager. Conclusions are made

about further ways to improve the work of the external component, the actual ways of developing the implemented technology for the existing configuration as a whole are revealed.
Keywords: data exchange, AMQP protocol, message broker, external processing, 1С, RabbitMQ

На сегодняшний день направления развития агропромышленного бизнеса как в стране в целом, так и в Белгородской области расширились. Все больше людей проявляют интерес к такой форме предпринимательской деятельности, как разведение животных. Внедрение средств автоматизации в процесс разведения позволяет повысить доходность бизнеса, сэкономить ресурсы, сократить издержки и разумнее использовать резервы и активы бизнеса. Основным видом деятельности компании ООО «Кроль и К» является производство и переработка мяса кролика в охлажденном виде. На данный момент предприятие использует решение на платформе 1С «Матрица. Кролиководство», но ряд операций, таких как, например, обмен данными для ведения аналитического учета в конфигурации 1С:Бухгалтерия не автоматизирован. Это снижает скорость получения актуальной информации о профильных документах, связанных с отраслью животноводства, что в результате сказывается на загруженности работников предприятия, продуктивности бизнес-процессов комплекса. Опираясь на вышеизложенную информацию, выявлена необходимость автоматизировать процесс обмена данными [5]. Для реализации поставленной цели необходимо: изучить понятия «брокер сообщений», принцип работы шины обмена данными RabbitMQ и протокола передачи данных AMQP, рассмотреть способ преобразования файлов из 1С в формат xml [4].

Цель работы заключается в создании внешней обработки с понятным интерфейсом, интеграция с текущим отраслевым решением «Матрица. Кролиководство» должна происходить без снятия конфигурации с поддержки. Требуется добавить функцию экспорта документов о поступлении материалов (документ, содержащий список закупленных кормов, например) для загрузки информации в виде структуры xml-файла в рабочую область обработки. Так же интерфейс формы обработки должен содержать поля для регистрации пользователя шины обмена данными (ввод адреса, порта, логина, пароля, виртуального хоста, точки обмена и имени очереди).

Таким образом, разрабатываемый внешний модуль обмена должен выполнять следующие функции:

- экспорт необходимого для передачи документа в файл формата xml;
- возможность выбора, где использовать компоненту: на стороне клиента или сервера;
- настройка параметров подключения: адрес, порт, логин, пароль, виртуальный хост;

- вывод сообщения об успешном соединении с сервером и подключении к шине обмена данными;
- создание дополнительных точек обмена и очередей (для возможности распределения каналов передачи сообщений между получателями и отправителями);
- отдельная вкладка для вставки текста отправляемого сообщения (для проверки, та ли информация, которую действительно хотят отправить в обмен, экспортировалась в файл);
- вкладка для проверки, получено ли сообщение (вывод информации пользователю о том, что сообщение получено и очередь освобождена от сообщений).

Для соответствия готового решения поставленным требованиям, следует спроектировать форму будущей обработки. Ее внешний вид показан ниже на (рис. 1).

Рисунок 1 – Форма обработки для обмена

Таким образом, на форме реализованы 4 вкладки: «Настройки подключения», «Точка обмена/очередь», «Отправить сообщение», «Получить сообщение». Интерфейс с вкладками создается при помощи создания иерархии групп элементов на форме. Для успешного функционирования и отработки всех ключевых операций были созданы пять команд: «Отправить сообщение», «Проверить компоненту», «Проверить подключение», «Прочитать сообщение», «Создание точки и очереди». Пример листинга для реализации функции «Проверить подключение» представлен ниже (рис. 2) [2, с. 326].

```

<НаКлиентеНаСервереВезКонтекста
] Процедура ПроверитьПодключениеКлиентСервер (КлиентКомпоненты, форма)

    Попытка
        КлиентКомпоненты.Connect (
            форма.Адрес,
            форма.Порт,
            форма.Логин,
            форма.Пароль,
            форма.ВиртуальныйХост);
    Исключение
        СистемнаяОшибка = ОписаниеОшибки();
        ТекстСообщения = "Ошибка подключения!%СистемнаяОшибка%";
        ТекстСообщения = СтрЗаменить (ТекстСообщения, "%СистемнаяОшибка%", СистемнаяОшибка);
        ВызватьИсключение ТекстСообщения;
    КонецПопытки;

    Сообщить (НСтр ("ru = 'Подключение успешно выполнено!'"));

- КонецПроцедуры

```

Рисунок 2 – Листинг процедуры «ПроверитьПодключениеКлиентСервер»

В пользовательском режиме через поиск (рис. 3) выбирается любой из документов «Поступление материалов», потому что именно для них был реализован функционал по экспорту в xml.

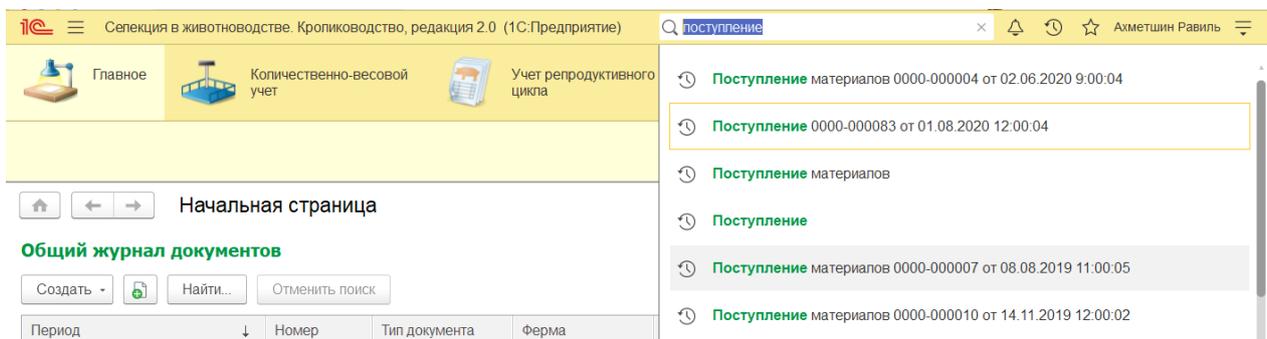


Рисунок 3 – Выбор документа из списка

Далее необходимо нажать на кнопку «Экспорт в XML», в результате этого действия описанная в коде команды кнопки процедура сформирует xml файл с учетом структуры документа и автоматически откроет вкладку с компонентой для обмена данными. Листинг обработчика события для экспорта документа в формат xml приведен ниже (рис. 4) [1, с. 91-94].

```

&НаКлиенте
Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)

    Результат = ПолучитьМассивДанных(ПараметрКоманды);

    ЗаписьXML = Новый ЗаписьXML();
    ЗаписьXML.УстановитьСтроку();
    ЗаписьXML.ЗаписатьОбъявлениеXML();

    ЗаписьXML.ЗаписатьНачалоЭлемента("ПоступлениеМатериалов"); // Начало элемента

    Для каждого ЭлементМассива Из Результат Цикл

        ЗаписьXML.ЗаписатьНачалоЭлемента("ПоступлениеМатериала");

        ЗаписьXML.ЗаписатьАтрибут("Номер", Строка(ЭлементМассива.Номер));
        ЗаписьXML.ЗаписатьАтрибут("Корм", Строка(ЭлементМассива.Корм));
        ЗаписьXML.ЗаписатьАтрибут("ЕдиницаИзмерения", Строка(ЭлементМассива.ЕдиницаИзмерения));
        ЗаписьXML.ЗаписатьАтрибут("Количество", Строка(ЭлементМассива.Количество));
        ЗаписьXML.ЗаписатьАтрибут("СтавкаНДС", Строка(ЭлементМассива.СтавкаНДС));
        ЗаписьXML.ЗаписатьАтрибут("Склад", Строка(ЭлементМассива.Склад));

    ЗаписьXML.ЗаписатьКонецЭлемента()

КонецЦикла;

ЗаписьXML.ЗаписатьКонецЭлемента(); // совсем конец записи взять и в строку передать
Стр = ЗаписьXML.Закрыть(); //чтобы не в файл, а сразу в строку вывел
//Сообщить(Стр);
Параметры = Новый Структура;
Параметры.Вставить("ТекстСообщения", Стр);
ОткрытьФорму("Обработка.ПростойПримерРаботыRabbitMQ.форма", Параметры);
//ВызватьОбработку(стр);

КонецПроцедуры
    
```

Рисунок 4 – Листинг процедуры «ЭкспортВXML»

Это сделано для уменьшения количества переходов между частями конфигурации и для ускорения процесса обработки документов (рис. 5).

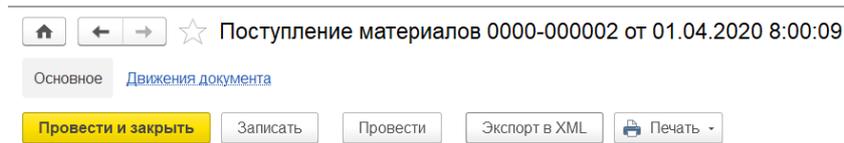


Рисунок 5 – Расположение кнопки экспорта

Автоматически сформируется xml-файл и будет помещен в окно формы обработки «Отправить сообщение» (рис. 6).

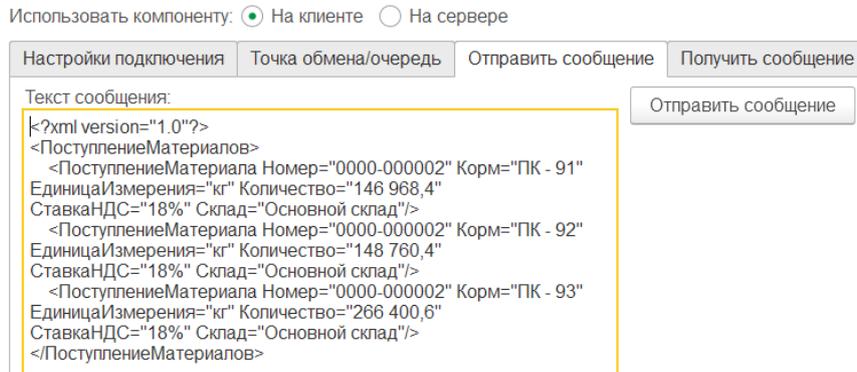


Рисунок 6 – Помещение файла в окно отправки сообщения

По нажатию на кнопку «Отправить сообщение», системное сообщение выдает информацию о том, что сообщение успешно отправлено (рис. 7).

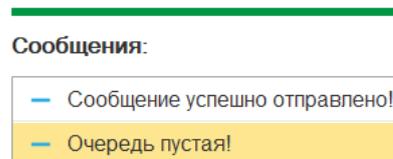


Рисунок 7 – Системное сообщение пользователю

Используя имя пользователя и пароль для регистрации в системе отслеживания передачи пакетов от RabbitMQ, которые вносятся на первой вкладке компоненты, можно отследить динамику обработки сообщений [3, с. 105-107]. На графике виден скачок – это отправленное в очередь для обмена сообщение (рис. 8).

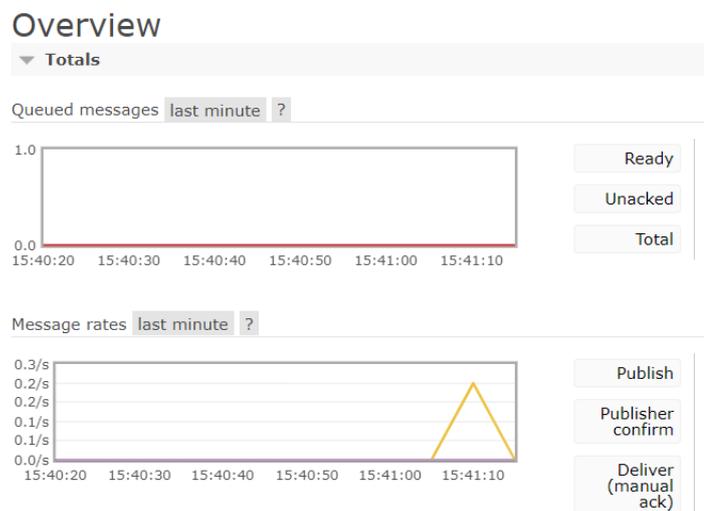


Рисунок 8 – Проверка отправки на стороне брокера

Разработанная внешняя компонента позволяет автоматически подключаться к серверу и отправлять необходимые для обмена со сторонней конфигурацией документы, предварительно экспортируя их в формат xml-файл. Результаты об успешном подключении компоненты, установлении соединения с сервером и отправке файла сразу выводятся на экран в виде системных сообщений. Таким образом, цель работы была достигнута, а поставленные задачи решены.

Бородин, В. Ю. Использование технологий xml при разработке автоматизированных систем [Текст]/ В.Ю. Бородин //Прогрессивные технологии в мировом научном пространстве. – 2020. – С. 91-94.

Габец А.П. Профессиональная разработка в системе 1С: Предприятие 8.0 / А.П. Габец и др. - М.: 1С-Публишинг, 2016. - 808 с.

Поляков, В. А. Обмен данными между информационными базами «1С» различной структуры [Текст]/ В.А. Поляков //Инновационное развитие техники и технологий в промышленности (ИНТЕКС-2020). – 2020. – С. 105-107.

Учебник по 1С конвертации данных [Электронный ресурс] – URL:

<https://helpme1c.ru/uchebnik-po-1s-konvertacii-dannyx-> (дата обращения 16.12.2020).

Функционал решения Матрица. Кролиководство [Электронный ресурс] – URL:
<https://www.matrix24.ru/rabbit/> (дата обращения 17.12.2020).