

# Очереди сообщений

- [Очередь Artix-Queue](#)
  - [Ошибки работы с очередью Artix-Queue](#)
- [Очередь File-Queue](#)
- [Система сообщений QPID](#)
  - [Брокер сообщений](#)
  - [Очереди сообщений](#)
  - [Команды для работы с очередью сообщений](#)
  - [Маршрутизация сообщений](#)

Начиная с версии 4.6.132 реализована отправка сообщений об операциях только через очередь Artix-Queue.

Начиная с версии 4.6.144 пакеты artix-aqspy и artix-queue удалены из кассового ПО Artix. Сервис Artix-Queue входит в состав пакета artix45-core.

Начиная с версии 4.6.160 изменен процесс загрузки справочников на кассу, система сообщений Qpid заменена на очередь File-Queue.

## Очередь Artix-Queue

Начиная с версии 4.6.128 реализована возможность отправки транзакций по уникальным купонам, бонусам, сертификатам через разные очереди Artix-Queue.

Обмен сообщениями между сервисами, приложениями и процессами может происходить посредством БД кассовой программы.

Управление сервисом осуществляется командами:

- service queue start
- service queue stop
- service queue restart

Отправка сообщений через очередь Artix-Queue используется при работе с:

- [Артикс Бонус](#)
- [Артикс Сертификат](#)
- [Уникальными купонами](#)
- [Внешними системами лояльностей](#)
- [Резервированием статуса акцизных марок](#)
- [Резервированием статуса кодов маркировки \(табак и обувь\)](#)
- [Акциями со счетчиками](#)
- [Универсальным передаточным документом](#)
- [Гифтоманом \(Giftoman\)](#)
- [Сервисом электронных рецептов \(Республика Беларусь\)](#)
- [ПКС "Цифровые кассы"](#) при отправке чека на адрес покупателя

Настройки сервиса задаются в файле /linuxcash/cash/queue/config/queue.ini:

Наименование	Тип данных	Описание	Примечания
timeout	целочисленный	Таймаут перезапуска обработчиков	По умолчанию 20
port	целочисленный	Порт HTTP-сервера для обработки REST-запросов	По умолчанию 28085

### Пример настройки

[queue]

; Таймаут перезапуска обработчиков  
; по умолчанию 20 секунд  
;timeout=20

; Порт HTTP-сервера для обработки REST-запросов  
; по умолчанию 28085  
;port=28085

Все сообщения, отправляемые через очередь Artix-Queue, записываются в [БД Queues в таблице Message](#). После добавления в таблицу сообщения обрабатываются сервисом и отправляются адресату, после подтверждения доставки сообщение удаляется из очереди.

Если обработчик (artixbonus и т.п.) не запущен, но в очереди присутствуют сообщения соответствующего типа, то обработчик перезапускается.



Типы сообщений определяются как строковые константы, совпадающие с именами обработчиков, и хранятся в [БД Queues в таблице Message](#) в поле type.

Настройки обработчиков хранятся в директории /linuxcash/cash/queue/config/queue.d/.

Логи работы сервиса Artix-Queue записываются в файл /linuxcash/logs/current/queue.log. Настройка логирования осуществляется в конфигурационном файле /linuxcash/cash/conf/Artix/queue.conf, уровень логирования по умолчанию – INFO.

В настройках можно изменить уровень логирования на любой из следующих:

- INFO,
- DEBUG,
- ERROR.

Самым детализированным уровнем логирования является DEBUG (вес лог-файла выше, чем у остальных).

По умолчанию размер основного файла, в который осуществляется запись логов, не может превышать 100 Mb. Если размер данных, записываемых в основной файл, превысит 100 Mb, то данные из него будут скопированы в дополнительный лог-файл с индексом "n", где "n" – номер лог-файла, а основной лог-файл будет перезаписан. Чем меньше индекс в названии дополнительного файла, тем актуальнее находящаяся в нем информация.

Количество таких файлов и максимальный размер можно задать в настройках, где по умолчанию "100MB" – размер файла в мегабайтах, "10" – количество файлов:

#### Пример настройки

```
[Log4Qt]
Properties\log4j.appender.root=org.apache.log4j.RollingFileAppender
Properties\log4j.appender.root.file=/linuxcash/logs/current/queue.log
Properties\log4j.appender.root.appendFile=true
Properties\log4j.appender.root.maxFileSize=100MB
Properties\log4j.appender.root.maxBackupIndex=10
Properties\log4j.appender.root.layout=org.apache.log4j.TTCCLayout
Properties\log4j.appender.root.layout.DateFormat=ISO8601
Properties\log4j.rootLogger="INFO, root"
```

При остановке или перезапуске сервиса, а также при завершении работы системы сервис завершает обработку текущего сообщения и после чего останавливается.

При наличии в очереди некорректных сообщений и последующем добавлении сообщений без ошибок сервис последовательно просмотрит все записи и отправит корректные сообщения адресату.

## Ошибки работы с очередью Artix-Queue

В случае ошибки добавления сообщений в очередь данные для отправки помещаются в /linuxcash/cash/data/queue/<типсообщения\_дата\_время\_номер>.queue. В верхней части экрана будет выведено уведомление "Очередь сообщений недоступна".

При попытке помещения нового сообщения в очередь осуществляется проверка файлов с ошибками и их помещение в очередь сообщений для отправки.

Помещение ошибочных сообщений в очередь осуществляется для всех сообщений:

- обращения к системам лояльности,
- работы с универсальными передаточными документами,
- резервирования на КС акционных марок алкогольной продукции и кодов маркировки для маркированной продукции.

В случае ошибки добавления сообщений в очередь при закрытии чека работа кассы блокируется, на экран выводится диалог "Ошибка добавления сообщения в Artix-очередь. Проверьте службу MySQL. Повторить попытку подключения?":


- при отказе от операции осуществляется переход в 1-ю консоль для аварийного завершения работы,
- при попытке подключения проверяется доступность БД:
  - если БД не была восстановлена, то на экран повторно будет выведен диалог "Ошибка добавления сообщения в Artix-очередь. Проверьте службу MySQL. Повторить попытку подключения?",
  - если БД была восстановлена, то диалог закрывается, работа кассы может быть продолжена в обычном режиме. Обработка файлов /linuxcash/cash/data/queue/<типсообщения\_дата\_время\_номер>.queue будет осуществлена при попытке помещения нового сообщения в очередь.

В кассовом ПО Artix предусмотрена возможность проверки состояния БД после закрытия чека, задается параметром `checkDatabaseAfterDocumentClose` в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[ArtixQueue]`:

- при значении `true` осуществляется проверка состояния БД. Если БД недоступна, то в верхней части экрана будет выведено уведомление "Ошибка подключения к БД",
- при значении `false` проверка состояния БД не осуществляется.

Оповещение о количестве неотправленных сообщений регулируется параметром `messagesCountWarning` в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[ArtixQueue]`. В верхней части экрана будет выведено уведомление "Большое количество сообщений в очереди (<количество\_сообщений>)".

Оповещение о наличии в очереди старых сообщений регулируется параметром `firstMessageDaysWarning` в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[ArtixQueue]`. В верхней части экрана будет выведено уведомление "В очереди есть старые сообщения (<дата>)".

 Настройки `firstMessageDaysWarning` и `messagesCountWarning` работают только при включенной настройке `checkDatabaseAfterDocumentClose`.

Наименование	Тип данных	Возможные значения	Описание	Примечания
<code>checkDatabaseAfterDocumentClose</code>	логический	<ul style="list-style-type: none"><li>• <code>true</code></li><li>• <code>false</code></li></ul>	Проверка состояния БД после закрытия чека	По умолчанию <code>false</code>
<code>firstMessageDaysWarning</code>	строковый		Количество дней с момента добавления первого неотправленного сообщения в очереди для вывода предупреждения	По умолчанию <code>-1</code> (отключено)
<code>messagesCountWarning</code>	строковый		Количество сообщений в очереди при достижении которого выводится предупреждение	По умолчанию <code>-1</code> (отключено)

Пример настройки

```
[ArtixQueue]
;
; Настройки для очереди Artix
...
; Проверка состояния БД после закрытия чека
; По умолчанию false
;checkDatabaseAfterDocumentClose = false

; Количество дней с момента добавления первого неотправленного сообщения в очереди для вывода предупреждения
; По умолчанию -1 (отключено)
;firstMessageDaysWarning = -1

; Количество сообщений в очереди при достижении которого выводится предупреждение
; По умолчанию -1 (отключено)
;messagesCountWarning = -1
```

## Очередь File-Queue

Начиная с версии 4.6.160 реализована новая очередь `File-Queue` для загрузки справочников на кассу.

Загрузка справочников на кассу осуществляется посредством очереди `File-Queue`, которая представляет собой файловую загрузку. Настройка очереди сообщений `File-Queue` осуществляется в конфигурационном файле `/linuxcash/cash/exchangesystems/exchangers/config/cash/exchangers.ini` в секции `[FILEQUEUE]`.

Наименование	Тип данных	Описание	Примечания
<code>fileQueueRoot</code>	строковый	Путь до корневой директории файл-очереди	По умолчанию <code>/linuxcash/cash/data/queues</code>
<code>queueChunkSize</code>	целочисленный	Количество <code>aif</code> -объектов в одном сообщении файл-очереди	По умолчанию <code>25000</code>

#### Пример настройки

[FILEQUEUE]

; Путь до корневой директории файл-очереди  
fileQueueRoot = /linuxcash/cash/data/queues

; Количество aif-объектов в одном сообщении файл-очереди  
queueChunkSize = 25000

Справочник с данными будет расформирован на json-файлы с расширением \*.objs, которые будут помещены в директорию artix.loaddict.cash, расположение которой задается настройкой fileQueueRoot. Команды справочника будут распределены по разным файлам. Если на кассу поступает справочник с большим количеством объектов, то он будет разбит на несколько файлов. Количество объектов в одном файле задается настройкой queueChunkSize. После того, как справочник будет разбит на части, формируется ещё один json-файл с данными об окончании транзакции. После завершения формирования всех файлов объекты будут отправлены на загрузку в базу данных. После завершения загрузки все json-файлы будут удалены из директории.

Если во время расформирования справочника в json-файлы произошла ошибка, то будет осуществлена повторная попытка разбиения справочника.

Более подробно о механизме работы очереди File-Queue можно прочитать в разделе "[Очередь](#)".

## Система сообщений QPID

Обмен сообщениями – это способ обмена определенными данными между процессами, приложениями и серверами, как виртуальными, так и физическими. Прием, создание очереди сообщений и их передача осуществляется брокером сообщений. Очереди сообщений представляют собой связующее звено между различными процессами в приложении, обеспечивая возможность взаимодействия с другими подключенными системами и устройствами. Использование очередей сообщений позволяет гарантировать доставку и обработку сообщений. Обработка очереди сообщений осуществляется автоматически или по запросу при помощи программы-посредника – брокера сообщений.

### Брокер сообщений

В кассовом ПО Artix используется брокер Qpid – система сообщений с открытым кодом, реализующая AMQP протокол. Qpid предоставляет набор инструментов для создания очередей сообщений, защиты и управления транзакциями.

Очередь сообщений позволяет выполнять передачу информации между компонентами системы и используется кассовым ПО для осуществления обменных процессов.

Демон qpidd является демоном брокера сообщений. Его основная задача – чтение файлов конфигурации очереди, ее создание и запуск, мониторинг работы, а так же непосредственный обмен сообщениями с другими модулями системы. Протокол AMQP основан на трех понятиях:

1. Сообщение (message) – единица передаваемых данных, основная его часть (содержание) никак не интерпретируется сервером.
2. Точка обмена (exchange) – в нее отправляются сообщения. Точка обмена распределяет сообщения в одну или несколько очередей. При этом в точке обмена сообщения не хранятся.
3. Очередь (queue) – здесь хранятся сообщения до тех пор, пока не будут забраны клиентом. Клиент всегда забирает сообщения из одной или нескольких очередей.

Очередь сохраняется в директории /var/spool/qpid/.

Управление состоянием процесса производится с помощью утилиты service. Подробнее о Qpid можно прочитать на [официальном сайте](#).

### Очереди сообщений

Для хранения и передачи различных данных используются отдельные очереди, каждая из которых имеет свою структуру и зависит от потребностей системы. Более подробно о механизме работы очереди сообщений можно прочитать в разделе "[Очередь](#)".

Очереди, используемые для обмена сообщениями:

- reply-ubuntu, qmfc-v2-ubuntu, qmfc-v2-hb-ubuntu, topic-ubuntu, qmfc-v2-ui-ubuntu – очереди системы,
- artix.load, artix.backward.cs – очереди, используемые при обмене.

### Команды для работы с очередью сообщений

Просмотр очереди сообщений выполняется по команде:

```
qpid-stat -q
```

Просмотр подключений выполняется по команде:

```
qpid-stat -c
```

Очистка очереди:

```
#Запустить qpid-tool  
qpid-tool  
#Просмотреть ID очередей  
list queue  
#Очистить данные в очереди  
call <ID очереди> purge 0 {}
```

Удаление очереди сообщений выполняется по команде:

```
qpid-config del queue <имя очереди> --force  
#Пример: qpid-config del queue artix.loaddict.222_shop --force
```



После удаления очереди сообщений необходимо перезапустить сервисы обмена.

## Маршрутизация сообщений

Начиная с версии 4.6.121 добавлен контроль роутинга сообщений выгрузки продаж с кассы на кассовый сервер.

Выгрузка продаж осуществляется через очередь artix.backward.cs, куда сервис sync-core2 помещает файлы продаж. Контроль доставки сообщений из очереди qpid на кассовый сервер осуществляется по настройке bridge, отражающей состояние роутинга сообщений.

Значение статуса роутинга можно посмотреть, выполнив команду:

```
qpid-config list bridge --show-property sync
```

При отправке сообщения из qpid кассы на КС выполняется контроль доставки, после успешной доставки сообщение удаляется из очереди. В случае корректной работы сервисов обмена настройка принимает значение 1. При отсутствии связи сообщения накапливаются в очереди, их отправка возобновляется после того, как связь с кассовым сервером будет восстановлена.

Если при отправке сообщения возникла ошибка, то статус настройки принимает значение 0. Для восстановления контроля необходимо устранить причину и перезапустить сервис sync-core2.