

ПРОГРАММНЫЙ КОМПЛЕКС ASTRAREGUL



РГДП.58.29.14.000-001-04 РП

# Базы данных

---

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Редакция 2

Соответствует версии Astra.Historian 1.1.3.1

Соответствует версии Astra.RMAP 1.2.2.15

# СПИСОК ИЗМЕНЕНИЙ

Редакция	Список изменений
Редакция 2	- Добавлен раздел <a href="#">Структура базы данных</a> .

# ОГЛАВЛЕНИЕ

СПИСОК ИЗМЕНЕНИЙ .....	2
1. Базы данных .....	5
1.1. Astra.Historian .....	6
1.1.1. Настройка .....	11
1.1.1.1. Вычисление значения атрибута MaxPoolMemorySize .....	13
1.1.1.2. Атрибуты баз данных .....	16
1.1.1.2.1. Атрибуты TCP-сервера .....	17
1.1.1.2.2. Атрибуты хранения .....	18
1.1.1.2.3. Атрибуты для ограничений на хранение данных.....	19
1.1.1.2.4. Атрибуты для сжатия архивированных данных .....	21
1.1.2. Структура базы данных.....	22
1.1.2.1. Общий формат записи и оценка её размера .....	24
1.1.2.1.1. Размер тела записи истории значений .....	25
1.1.2.1.2. Размер тела записи истории событий.....	27
1.1.2.2. Коэффициент сжатия .....	28
1.1.2.3. Общая оценка размера базы данных .....	29
1.1.2.4. Оценка суточного объема данных от сервера .....	32
1.1.2.4.1. Оценка суточного объема истории значений.....	33
1.1.2.4.2. Оценка суточного объема истории событий .....	34
1.1.2.4.3. Пример расчета .....	35
1.1.2.5. Оценка объема временного хранилища очереди данных.....	37
1.1.2.6. Оценка объема заполнения дискового пространства .....	39
1.1.3. Диагностика работы .....	41
1.1.4. Администрирование.....	46
1.1.4.1. Перенос баз данных .....	48
1.1.4.2. Сохранение части архива базы данных .....	49
1.1.4.3. Восстановление сохраненных архивных данных для просмотра .....	50
1.2. Astra.RMap.....	51
1.2.1. Настройка .....	52
1.2.1.1. Windows.....	53
1.2.1.2. AstraLinux.....	57
1.2.1.3. РЕД ОС .....	63
1.2.2. Виртуальные таблицы .....	65
1.2.2.1. Таблицы данных .....	66
1.2.2.2. Таблицы событий.....	70
1.2.3. SQL-запросы .....	73

1.2.3.1. Примеры.....	75
1.2.4. Подключение по TCP.....	79
1.2.5. Предоставление данных в MS Excel.....	81

# 1. Базы данных

Компонент	Версия	Описание
<a href="#">Astra.Historian</a>	1.1.1.1	Проприетарная нереляционная БД
<a href="#">Astra.RMAP</a>	1.2.1.1	Расширение СУБД PostgreSQL

## 1.1. Astra.Historian

**Astra.Historian** – программный компонент для хранения истории изменений значений сигналов и уведомлений о происшедших событиях.

Функции:

- › сбор и хранение оперативных значений параметров технологического процесса.
- › сбор и хранение истории событий и тревог технологического процесса.
- › предоставление исторических данных клиентам.

Функционирует в виде:

- › службы Astra.Historian.Server на ОС Windows;
- › сервиса astra.historian.server.service на ОС Linux.



На компьютере может быть установлен только один экземпляр сервера истории.

## Сохранение и получение данных

Источниками данных являются экземпляры сервера ввода/вывода Astra.Server.

Astra.Server обеспечивает сбор, фильтрацию и сохранение данных (событий и значений сигналов) в БД Astra.Historian через модуль истории.



Модуль истории в составе Astra.Server выполняет временное хранение данных на стороне сервера и передачу информации в хранилище Astra.Historian.

Буфер временного хранения данных на стороне Astra.Server располагается на жестком диске, что предотвращает потерю данных при аварийном отключении

компьютера сервера технологических данных. При следующем старте сервера переданные данные будут повторно отправлены в Astra.Historian. Буферизация данных позволяет также сгладить пиковые нагрузки при большой интенсивности получения данных.

Astra.Historian предназначен для управления базами данных и предоставления хранимой исторической информации клиентам. Сервер может обслуживать несколько БД одновременно.

Базы данных хранят данные, предоставляемые модулем истории. Используется БД, управляемая Astra.Historian. В одну БД могут сохраняться данные с нескольких независимых источников.

## Обработка данных

Данные, сохраняемые в сервер истории, попадают во внутренний журнал хранилища. Это способствует ускорению процесса перемещения данных из очереди источника.

Для каждого хранилища в рамках транзакций сервер истории выполняет последовательную обработку поступивших данных с целью размещения их в основной области. Обработка данных происходит в фоновом режиме и не препятствует операциям чтения и записи.

Сервер истории обрабатывает данные порциями в том порядке, в котором они были сохранены в очередь. Сервер истории фильтрует сохраняемые данные и отбрасывает устаревшие (метка времени лежит левее текущей нижней границы для всего хранилища) и архивные (метка времени лежит в текущих границах архивной области). Все остальные данные успешно сохраняются в основную область хранилища.

Сохраняемые данные записываются в соответствующие фрагменты оперативной области хранилища, если подходящего фрагмента нет, то он создается. В файл фрагмента попадают записи временных рядов по всем элементам, информация по которым хранится в истории.

Данные в БД хранятся в закрытом бинарном формате. Сервер допускает лишь добавление новых записей. При этом механизм обновления ранее сохраненной истории, реализованный в Astra.Imitator, не модифицирует существующие записи, а добавляет новые с пометками о перезаписи поверх

ранее сохраненных, формируя таким образом новый слой записей. В текущей реализации пользовательские инструменты позволяют просмотреть записи только из самого последнего сохраненного слоя. Для получения записей из предыдущих слоев в настоящее время необходимо обратиться к разработчику. В дальнейшем средства доступа к историческим данным будут доработаны таким образом, чтобы было возможным чтение в том числе и перезаписанных данных.

Содержимое источника представляет собой совокупность пространств, каждое из которых является совокупностью элементов. С течением времени происходят связанные с элементом события. Соответственно событиям строится временной ряд, который упорядочен по метке времени. Каждая запись временного ряда соответствует событию элемента.

По мере поступления сохраняемых данных в хранилище накапливается информация об источниках, пространствах и элементах, история значений которых сохраняется в данное хранилище, если таковой информации нет, то сервер истории создает ее.

## Хранение данных

В сервере истории данные хранятся в фрагментах - файлах, содержащих данные за сутки. У каждого фрагмента определяется нижняя граница времени для фрагмента - начало суток по времени UTC.

После поступления в сервер истории, данные последовательно проходят несколько стадий хранения:

- в активной области;
- в архивной области.

**Активная область** предназначена для формирования поступающих данных во временные ряды: это обеспечивает высокую скорость обработки запроса данных. Каждому источнику данных соответствует своя активная область. Длительность хранения фрагментов в активной области задается в настройках БД.

Фрагменты, нижняя граница которых оказывается левее предела времени хранения в активной области, переносятся в **архивную область**. Фрагменты из всех активных областей переносятся в единую архивную область, при этом



фрагменты из разных активных областей, имеющие одинаковую нижнюю границу времени сливаются в один файл фрагмента. Для уменьшения занимаемого места на диске, фрагменты в архивной области сжимаются с параметрами сжатия, указанными в настройках БД.

Хранение данных ведется в суточных файлах данных для увеличения скорости доступа к данным. Сервер реализует механизмы сохранения и поиска необходимых данных, направленные на обеспечение максимальной производительности работы с дисковой подсистемой компьютера.

Глубина хранения данных ограничена размерами дискового пространства. Скорость записи и чтения данных не зависит от глубины хранения. Запись в сервер – транзакционная. Сервер обеспечивает высокую плотность записи хранимых данных на диск, уменьшая таким образом объемы читаемых с диска данных.

## Резервирование

Astra.Historian позволяет формирование резервируемых хранилищ данных.

Резервирование представляет собой параллельное сохранение исторических данных источником данных в несколько баз данных, принадлежащих разным серверам истории.

При работе с резервируемыми хранилищами, данные из источника не удаляются, пока не пройдет запись во все хранилища.



Сохранение данных в несколько баз данных, принадлежащих одному серверу истории не решает задачу резервирования и не рекомендуется.

## Предоставление данных клиентам

Предоставление данных, хранящихся в базах данных сервера истории, выполняет модуль истории в составе Astra.Server или Astra.AccessPoint.

Предоставление данных клиентам осуществляется по проприетарному протоколу на базе TCP.



## Характеристики

- Производительность записи: до 750 000 изменений значений в секунду.
- Производительность чтения: до 1 000 000 изменений значений в секунду.

## Файловая структура базы данных

В файловой системе база данных содержит следующие папки и файл:

- active
- archive
- imported
- stx
- db.index

Historian > Databases > DB\_EVENTS > Поиск в: DB\_EVENTS

Имя	Дата изменения	Тип	Размер
active	21.09.2022 15:14	Папка с файлами	
archive	29.08.2022 8:55	Папка с файлами	
imported	31.08.2022 15:10	Папка с файлами	
stx	22.09.2022 14:26	Папка с файлами	
db.index	30.08.2022 11:41	Файл "INDEX"	8 КБ

Папки и файл создаются автоматически, когда сервер истории создает базу данных. Их расположение в файловой системе зависит от описания базы данных в файле конфигурации (атрибут **DefaultPrimaryDir** сервера истории).

## 1.1.1. Настройка

Вы можете настроить Astra.Historian в среде разработки Astra.IDE, используя плагин AstraRegul, либо вручную, отредактировав файл конфигурации **Astra.Historian.Server.xml** в папке установки.


### Конфигурационный файл



```
<?xml version="1.0" encoding="utf-8"?>
<Astra.Historian.Server StatPort="3388" DefaultPrimaryDir="c:
\Historian\Databases" MaxPoolMemorySize="512" EnableDCOM="0">
  <tcp-server default-port=4949 idle-sessions-count=1 idle-sessions-
timeout=15>
    <server-endpoint host="0.0.0.0" />
  </tcp-server>
  <Bases>
    <Base Alias="DB_EVENTS" PreferredCommonCacheLimit="128"
ActiveStorageDepth="3" StorageDepth="365" VolumeLimit="10000"/>
    <Base Alias="DB_HISTORY" PreferredCommonCacheLimit="128"
ActiveStorageDepth="3" StorageDepth="365" VolumeLimit="10000"/>
  </Bases>
</Astra.Historian.Server>
```

### Атрибуты

Атрибут	Описание	Значение по умолчанию
StatPort	Номер порта, по которому будет открыт доступ к статистики сервера. Обязательный атрибут.	3388
DefaultPrimaryDir	Путь до основного каталога файлов БД. Используется в качестве значения по умолчанию для БД, в которых не переопределен	"C:\Historian\Databases"

MaxPoolMemorySize	<p>Максимальный объем оперативной памяти, используемой в качестве кеша для работы с базами данных (Мб).</p> <p>0 - без ограничения.</p> <p>В 64-битном варианте верхняя граница - 128 ГБ.</p> <p>Значение данного атрибута вычисляется по <a href="#">формуле</a>.</p>	512
EnableDCOM	<p>Поддержка DCOM.</p> <p>Значения:</p> <p>0 – отключен</p> <p>1 – включен</p> <div>  <p>В новых проектах необходимо отключать поддержку DCOM.</p> </div>	1



Чтобы изменения вступили в силу, перезапустите службу (сервис) Astra.Historian.

# 1.1.1.1. Вычисление значения атрибута MaxPoolMemorySize

Для вычисления значения атрибута используется следующая формула:



$$MaxPoolMemory \geq \sum_i (ServerTagsCount_i * R_i) * 1.2 * 4KB + \sum_j DBCache_j * 1MB$$

где  $i$  – отдельный экземпляр Astra.Server, для которого настроено сохранение истории значений сигналов в базы данных данного сервера истории;

ServerTagsCount – количество сигналов, сохраняемых  $i$ -м экземпляром Astra.

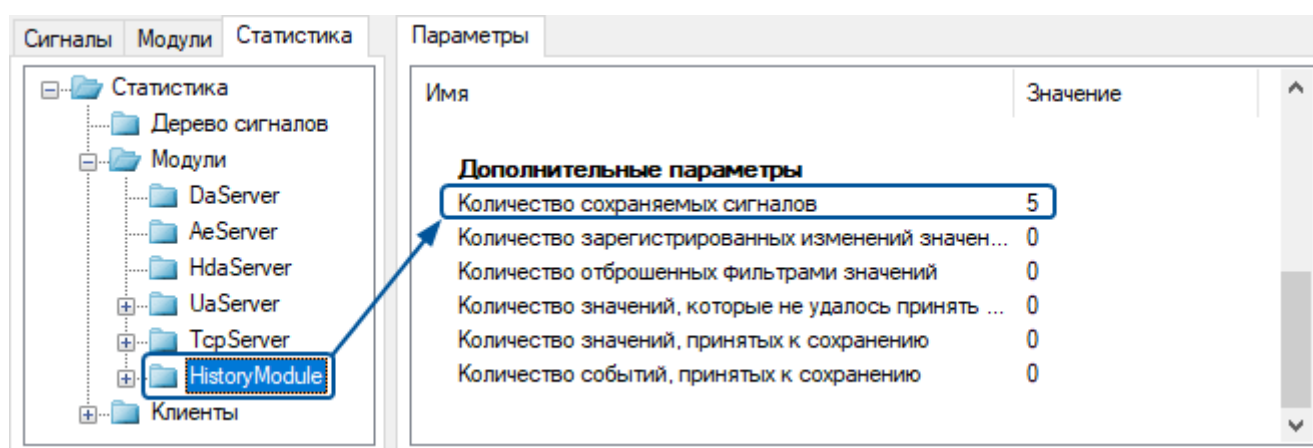
$R$  – если  $i$ -й Astra.Server работает в составе резервной пары, то значение равно 2, иначе - 1;

$j$  – отдельная БД данного сервера истории;

DBCACHE – размер оперативной памяти, выделенной для  $j$ -й базы данных в кэше (параметр PreferredCommonCacheLimit в файле конфигурации).



Количество сохраняемых сигналов можно посмотреть в приложении Статистика.



## Пример

Сервер истории управляет двумя базами данных и хранит историю значений для трех экземпляров Astra.Server:

- ProcessVals (значение PreferredCommonCacheLimit – 256);

›ControlVals (значение PreferredCommonCacheLimit не указано, берем значение по умолчанию - 128).



```
<?xml version="1.0" encoding="utf-8"?>
<Astra.Historian.Server StatPort="3388" DefaultPrimaryDir="c:
\Astra.Historian\Databases">
  <tcp-server default-port=4949 idle-sessions-count=1 idle-sessions-
timeout=15>
    <server-endpoint host="0.0.0.0" />
  </tcp-server>
  <Bases>
    <Base Alias="ProcessVals" PreferredCommonCacheLimit="256" />
    <Base Alias="ControlVals" />
  </Bases>
</Astra.Historian.Server>
```

Данный сервер истории хранит историю значений для трех экземпляров Astra.Server:

- ›Первый Astra.Server сохраняет значения 15000 сигналов.
- ›Второй Astra.Server сохраняет значения 10000 сигналов и работает в составе резервной пары.
- ›Третий Astra.Server сохраняет значения 7500 сигналов.



Для вычисления значения MaxPoolMemory неважно, в какую базу данных сервера истории записывает значения сигналов каждый из экземпляров Astra.Server.

Подставив значения в формулу, получим значение атрибута:


$$\text{MaxPoolMemory} = (15000 + 10000 * 2 + 7500) * 1.2 * 4\text{КБ} + (256 + 128) * 1\text{МБ} = 588\text{ МБ}$$

Поскольку значение атрибута должно быть не меньше вычисленного значения, то в файле конфигурации сервера истории можно указать вычисленное значение с округлением вверх.



```
<?xml version="1.0" encoding="utf-8"?>
<Astra.Historian.Server StatPort="3388" DefaultPrimaryDir="c:
\Astra.Historian\Databases" MaxPoolMemorySize="600">
  <tcp-server default-port=4949 idle-sessions-count=1 idle-sessions-
timeout=15>
    <server-endpoint host="0.0.0.0" />
  </tcp-server>
  <Bases>
    <Base Alias="ProcessVals" PreferredCommonCacheLimit="256" />
    <Base Alias="ControlVals" />
  </Bases>
</Astra.Historian.Server>
```

## 1.1.1.2. Атрибуты баз данных

Базы данных перечислены в элементе Bases.

После первой установки список баз данных пуст. Чтобы добавить базу данных, в элемент Bases добавьте элемент Base и укажите его атрибуты – параметры базы данных. Сервер истории создаст базу данных при перезапуске.

[Атрибуты TCP-сервера](#)

[Атрибуты хранения](#)

[Атрибуты для ограничений на хранение данных](#)

[Атрибуты для сжатия архивированных данных](#)



## 1.1.1.2.1. Атрибуты TCP-сервера

Атрибут	Описание	Значение по умолчанию
default-port	Порт по умолчанию, используется для дочерних элементов server-endpoint, у которых не указан порт.	4949
idle-sessions-count	Количество одновременно удерживаемых неактивных сессий. Значение по умолчанию – 0 (не ограничено).	1
idle-sessions-timeout	Время ожидания активации неактивных сессий (в секундах). Значение по умолчанию – 0 (не ограничено).	15
server-endpoint	Адрес, по которому доступен сервер.	0.0.0.0

### Атрибуты server-endpoint

В элемент tcp-server можно добавить любое количество дочерних элементов server-endpoint.

Атрибут	Описание	Значение по умолчанию
port	Порт точки доступа к серверу. Если не указан, используется порт default-port родительского элемента.	default-port
host	Хост точки доступа к серверу.	0.0.0.0 (любой адрес)

## 1.1.1.2.2. Атрибуты хранения

Название	Описание	Значение по умолчанию
Alias	Короткое имя БД (псевдоним), должно быть уникальным для сервера истории Обязательный атрибут.	
PrimaryDir	Путь до основного каталога БД. Если не указан, то совпадает с DefaultPrimaryDir из настроек сервера	
ArchiveDir	Путь до архивного каталога БД. Если не указан, то совпадает с PrimaryDir	

### 1.1.1.2.3. Атрибуты для ограничений на хранение данных

Название	Описание	Значение по умолчанию
PreferredCommonCacheLimit	Предпочтительный объем оперативной памяти в мегабайтах, используемой в качестве кеша для запроса данных из базы данных. 0 - без ограничений. Чем больше значение данного параметра, тем больше данных будет находиться в кэше, обращение к которому выполняется значительно быстрее, чем запрос данных с диска.	128
ActiveStorageDepth	Длительность хранения данных в активной области в сутках. Минимальное значение – 1.	3
StorageDepth	Максимальная глубина хранения данных в БД до удаления в сутках. 0 - без ограничений; если не 0, то должно быть больше или равно значения ActiveStorageDepth.	0
VolumeLimit	Предельный объем БД в мегабайтах; при превышении предела выполняется очистка архивных данных до снижения объема ниже предела, если это возможно. 0 – без ограничений.	0



Атрибуты ActiveStorageDepth и StorageDepth отвечают за бракование данных, которые не попали в базу данных из-за устаревшей или опережающей метки времени.

Для каждого активного раздела учитываются следующие временные границы (слева-направо в направлении оси времени):

$\text{OutdatedBound} < \text{ActiveLowerBound} < \text{FrontBound} < \text{CurrentHistorianServerTime}$ ,

где **OutdatedBound** – это общая нижняя граница по общей глубине хранения, все, что ниже нее, признается устаревшим и сразу удаляется (застойные данные, например, когда база долго была в простое и ничего не передавалось в архив).

**ActiveLowerBound** – это нижняя граница активности для данной конкретной активной области:



$$\text{ActiveLowerBound} = \text{FrontBound} - \text{ActiveStorageDepth}$$

Все, что поступает на запись левее нее, также признается устаревшим и отбрасывается. А то, что уже сохранено (имеющиеся фрагменты левее нижней границы) запечатываются и передаются в архив.

**FrontBound** – фронтальная граница, соответствующая начальной границе самого "молодого" фрагмента в этом разделе. Смещается при появлении очередного самого "молодого" фрагмента, который, в свою очередь, появляется, когда по одному из тегов накапливаются данные, примерно, на 4 КБ. При этом если «хвост» данных приходится на следующие сутки (по UTC), то граница **FrontBound** смещается на следующие сутки. При этом если это будут не прям следующие сутки, а +N суток, то создастся такой самый "молодой" фрагмент, что граница **FrontBound** сместится правее на +N суток, что в итоге приведет к соответствующему смещению **ActiveLowerBound**.

**CurrentHistorianServerTime** – текущее время на сервере (Astra.Historian). Используется для фильтрации записей "из будущего": если метка времени сохраняемой записи  $> \text{CurrentHistorianServerTime} + 1 \text{ hour}$ , то она отбрасывается как опережающая.

## 1.1.1.2.4. Атрибуты для сжатия архивированных данных

Название	Описание	Значение по умолчанию
ArchiveCompression	Алгоритм сжатия архивных файлов: <ul style="list-style-type: none"> <li>› none – без сжатия;</li> <li>› lzma – сжатие алгоритмом LZMA.</li> </ul>	none
BlocksPerCompressedPack	Архивные файлы сжимаются группами по несколько файлов в один сжатый файл. Значение данного параметра задает максимальное количество архивных файлов в группе. Чем больше значение данного параметра, тем выше коэффициент сжатия и тем медленнее чтение данных из сжатых файлов.	10
Recompress	Стратегия выбора файлов для сжатия: none – сжатие не применяется; uncompressed – сжатие применяется только к несжатым файлам; any – сжатие применяется ко всем архивированным файлам: сжатые файлы сжимаются с новыми параметрами сжатия. Файлы, параметры сжатия ArchiveCompression и BlocksPerCompressedPack которых совпадают с новыми параметрами сжатия, пересжиматься не будут.	none



Размер сжатых данных точно определить нельзя, так как коэффициент сжатия зависит от самих данных. На практике он варьируется от 2.5 до 3.5.

## 1.1.2. Структура базы данных

База данных состоит из следующих разделов:

- **active/<uuid>** – множество активных разделов, каждый из которых создается для отдельного источника, выполняющего сохранение данных.
- **archive** – единый архивный раздел, в который данные передаются из активных разделов по мере накопления.

Также база данных содержит служебный файл:

- **db.index** – основной индекс базы данных.

### Активный раздел

Активный раздел содержит:

- Служебные файлы:
  - **dbpart.index** – индекс элементов раздела.
  - **dbpart.workset** – рабочее множество раздела.
  - **dbpart.tx** – файл журнала состояния раздела (данный файл имеет небольшой размер (< 1 МиБ), и не учитывается в оценках.
- **ahdb-<Date>.store** – множество фрагментов данных.

### Архивный раздел

Архивный раздел содержит:

- **ahdb-<Date>.store** – множество фрагментов данных.

Размер основного индекса базы данных от общего числа элементов в базе данных и среднего размера идентификатора элементов (зависят от источника). Размеры индекса и рабочего множества активных разделов зависят от количества элементов от соответствующего источника.

Размер фрагментов данных зависит от числа содержащихся в них записей и их размеров.



Фрагменты в активных и архивном разделах имеют одинаковый внутренний формат.

Число фрагментов в активных разделах зависит от параметра `ActiveStorageDepth`, определяющего глубину активного хранения (с

возможностью записи) для базы данных. Число фрагментов в архивном разделе – от параметра StorageDepth, определяющего общую глубину хранения для базы данных.

В активных разделах фрагменты всегда хранятся в несжатом виде, в архивном – в зависимости от настроек сжатия.

Сжатые файлы (в архивной папке archive) визуально не отличаются от несжатых. Отличие состоит в их внутреннем содержимом (формат которого определяется по служебной информации, сохраняемой в файл).

Astra.Historian выполняет не сжатие архиватором всего файла, а сжатие отдельных блоков содержимого при поддержании одинаковых системных и индексных структур.

При перемещении фрагмента данных из активной области в архивную выполняется его дефрагментация для улучшения характеристики кластеризации данных по тегам (для повышения эффективности чтения). Это приводит к уплотнению данных и изменению итогового размера файла.



Размеры файлов индексов, рабочего множества и фрагментов данных в активных разделах выравниваются на 16 МиБ (связано с необходимостью упреждающего выделения внешней памяти для сокращения издержек).



Если источник резервируется, то активный раздел создается для каждого его экземпляра.

## 1.1.2.1. Общий формат записи и оценка её размера

В базах данных записи хранятся в виде пар < Timestamp, Body >:

- Timestamp – 8-байтная метка времени записи;
- Body – последовательность байтов, содержащая тело записи в том виде, в каком она была сохранена источником.

Метка времени и необходимая для хранения записи служебная информация образуют ее "константную" часть, совокупный размер которой составляет 12 байт.

Тело записи, размер которого зависит от источника, составляет ее "переменную" часть. Ее размер хранится в служебной информации (учтен в 12-байтной константной части).

Astra.Server сохраняет исторические данные двух видов:

- история значений – события изменения значений тегов (сигналов);
- история алармов – история событий, связанных с алармами: активация, деактивация, квитирование и др.

**Размер отдельной записи** в несжатой части базы данных вычисляется по следующей общей формуле:

*f*

$\text{RecordSize}(x) = 12 + \text{BodySize}(x)$ ,

где  $x$  – некоторая запись;

$\text{BodySize}(x)$  – размер тела записи, зависящий от источника и самой записи (тело может иметь фиксированную или переменную длину в зависимости от типа записи).



## 1.1.2.1.1. Размер тела записи истории значений

Тело записи содержит значение, качество, метку времени сервера (в б.х серверах сохраняется опционально) и необходимую служебную информацию.

Размер тела записи истории значений зависит от типа значения:

- для булевого, числовых типов и метки времени – фиксированный размер;
- для строк – в зависимости от размера строки.

В таблице ниже приведен размер элементов тела записи для различных типов данных:

Тип значения	Метка времени и необходимая служебная информация		Тело записи		
	Метка времени, байт	Служебная информация внутри БД, байт	Качество, байт	Дескриптор типа значения, байт	Значение, байт
bool	8	4	4	1	0
int1					1
uint1					1
int2					2
uint2					2
int4					4
uint4					4
int8					8
uint8					8
float					4
double					8
string*					4 + N

\* – В случае строкового типа N – размер тела строки строки.

Соответствие размеров тела записи типам с учетом и без учета метки времени приведено в следующей таблице:

Тип значения	Размер тела записи без учета метки времени, байт	Размер тела записи с учетом метки времени, байт
bool	5	17
int1	6	18
uint1	6	18
int2	7	19
uint2	7	19
int4	9	21
uint4	9	21
int8	13	25
uint8	13	25
float	9	21
double	13	25
string*	9 + N	21 + N

\* – В случае строкового типа N – размер тела строки строки.



Строки сохраняются в кодировке UTF-8, соответственно, при его вычислении необходимо учитывать следующее:

- латинские символы занимают 1 байт;
- кириллические символы - 2 байта.

## 1.1.2.1.2. Размер тела записи истории событий

История алармов от одного источника Astra.Server сохраняется в один общий временной ряд.

Записи истории алармов имеют переменный размер, так как:

- содержат строковые данные переменной длины: сообщение, условие, подусловие, сообщение квитирования;
- могут содержать переменное количество атрибутов.

В историю алармов сохраняются не только события активации (срабатывания алармов), но и последующие деактивации, события квитирования (которые могут быть множественными), события установки и снятия блокировок и подавлений. Поэтому оценка суточного объема истории алармов зависит от конкретной конфигурации Astra.Server и особенностей ее работы.

Количество байт, которые приходятся на одно сообщение, зависит от содержимого сообщения:

Содержимое	Средний размер на одно сообщение, байт
Цифры	174
Кириллица и цифры	200
Латиница и цифры	179



На практике среднее значение объема одного исходного события составляет 200-300 байт.

## 1.1.2.2. Коэффициент сжатия

Для сжатия содержимого фрагментов архивного раздела используется алгоритм LZMA. Коэффициент сжатия в значительной степени зависит от исходных данных. На практике он варьируется в пределах 2-5 крат (выводится в журнал в сообщении о завершении архивации фрагмента).

## 1.1.2.3. Общая оценка размера базы данных

Обозначения принятые при расчете размера базы данных:

- › **Source[i]** – множество источников, сохраняющих данные.
- › **Align16MiB(x)** - операция выравнивания размера до 16 МиБ:

*f*

$\text{Align16MiB}(x) = \text{ceil}(x / 16 \text{ MiB}) * 16 \text{ MiB}$ ,  
где **ceil** – оператор округления вверх до ближайшего целого;

- › **SUM(expr)** – сумма подвыражений expr по множеству.

Общий размер базы данных вычисляется по следующей формуле:

*f*

[Общий размер базы данных] = [Размер хранимых данных] + [Размер служебных данных]

Размер хранимых данных (РХД):

*f*

$\text{РХД} = \text{РАР} + \text{РХДАР}$ ,  
где **РАР** – размер архивного раздела;  
**РХДАР** – размер хранимых данных в активных разделах.

Размер служебных данных (РСД):

*f*

$\text{РСД} = \text{РФОИ} + \text{РСДАО}$ ,  
где **РФОИ** – размер файла основного индекса;  
**РСДАО** – размер служебных данных в активных областях.

Размер архивного раздела РАР:

*f*

$\text{РАР} = \text{SUM}(\text{Source}[i].\text{PerDayVolumeSizeEstimation}) * (\text{StorageDepth} - \text{ActiveStorageDepth}) / \text{CompressRatio}$ ,

**Source[i].PerDayVolumeSizeEstimation** – суточный объем данных по источникам;

**StorageDepth** – общая глубина хранения, задаваемая в настройках БД;

**ActiveStorageDepth** – глубина активного хранения, задаваемая в настройках БД;

**CompressRatio** – оценка коэффициента сжатия данных (1.0, если сжатие отключено).



В оценке размера архивного раздела из общей глубины вычтена глубина активного хранения, поскольку данные по мере переноса в архивный раздел удаляются из исходных активных разделов.

Размер хранимых данных активного раздела РХДАР:



$$РХДАР = SUM(Align16MiB( Source[i].PerDayVolumeSizeEstimation) ) * ActiveStorageDepth,$$

**Source[i].PerDayVolumeSizeEstimation** – суточный объем данных по источникам;

**ActiveStorageDepth** – глубина активного хранения, задаваемая в настройках БД.



Оценка совокупного размера данных в активных областях основана на предположении, что резервированные источники сохраняют данные с взаимоисключением: в каждый момент времени сохранение выполняет активный источник.

Размер файла основного индекса РФОИ:



$$РФОИ = Align16MiB( SUM( Source[i].NumItems ) * ( 4 + AvgItemIdSize ) ),$$

**Source[i].NumItems** – число элементов, по которым ведется сохранение;

**AvgItemIdSize** – средний размер идентификатора элемента.

Размер хранимых данных активного раздела РСДАО:



$$РСДАО = SUM( Source[i].RedundancyScaleFactor * ( Align16MiB( Source[i].NumItems * 16 ) + Align16MiB( Source[i].NumItems * 4096 ) ) ),$$

**Source[i].RedundancyScaleFactor** – множитель резервирования источника: 1 - не резервируется, 2 - 2-кратное резервирование, 3 - 3-кратное и т.д;

**Source[i].NumItems** – элементы, по которым ведется сохранение.



Оценка включает размер служебных данных.

В общем виде общий размер базы данных (ОРБД) определяется:



$$\begin{aligned} \text{ОРБД} = & \frac{\text{SUM}(\text{Source}[i].\text{PerDayVolumeSizeEstimation}) * (\text{StorageDepth} - \text{ActiveStorageDepth})}{\text{CompressRatio}} + \\ & \text{SUM}(\text{Align16MiB}(\text{Source}[i].\text{PerDayVolumeSizeEstimation})) * \\ & \text{ActiveStorageDepth} + \text{Align16MiB}(\text{SUM}(\text{Source}[i].\text{NumItems}) * (4 \\ & + \text{AvgItemIdSize})) + \text{SUM}(\text{Source}[i].\text{RedundancyScaleFactor} \\ & * (\text{Align16MiB}(\text{Source}[i].\text{NumItems} * 16)) + \\ & \text{Align16MiB}(\text{Source}[i].\text{NumItems} * 4096)) \end{aligned}$$

**Source[i].PerDayVolumeSizeEstimation** – суточный объем данных по источникам;

**StorageDepth** – общая глубина хранения, задаваемая в настройках БД;

**ActiveStorageDepth** – глубина активного хранения, задаваемая в настройках БД;

**CompressRatio** – оценка коэффициента сжатия данных (1.0, если сжатие отключено).

## 1.1.2.4. Оценка суточного объема данных от сервера

Оценка суточного объема данных от Astra.Server выполняется для вычисления суточного объема данных от источников `Source[i].PerDayVolumeSizeEstimation`.

Для получения оценки совокупного суточного размера исторических данных от Astra.Server необходимо сложить оценки суточного объема истории значений и истории событий. При этом необходимо учитывать, как ведется сохранение истории значений и алармов: совместно или отдельно (в зависимости от настроек Astra.Server/HistoryModule).

### Размер идентификатора элемента `AvgItemIdSize`

В случае с Astra.Server полный идентификатор каждого отдельного элемента истории (для истории значений – отдельно на каждый сохраняемый тег, для истории алармов – один) имеет фиксированный размер 40 байт.



## 1.1.2.4.1. Оценка суточного объема истории значений

История значений сохраняется отдельно по каждому тегу (каждому тегу, сохраняемому в историю, в базе данных соответствует отдельный элемент и отдельный временной ряд, соответственно). Сохранение выполняется событийно с учетом настроек чувствительности по значению, метки времени и принудительной повторной записи.



Для оценки совокупного суточного объема истории значений необходимо сложить оценки по всем сохраняемым тегам.

Оценка суточного объема данных по отдельному тегу имеет следующий вид:


$$\text{PerDayTagValueDataVolumeSize} = 86400 * \text{AvgTagFrequencyEstimation} * (12 + \text{TagValueRecordBodySize}),$$

где 86400 – число секунд в сутках,

**AvgTagFrequencyEstimation** – оценка средней частоты сохранения значений по тегу в историю (после фильтрации по чувствительности и с учетом принудительного повторного сохранения);

**TagValueRecordBodySize** – размер тела записи.

## 1.1.2.4.2. Оценка суточного объема истории событий

Оценка совокупного суточного объема истории алармов, исходя из оценки среднего удельного размера на один аларм, может быть вычислена следующим образом:

*f*

$PerDayTagValueDataVolumeSize = 86400 * AvgTagFrequencyEstimation * AvgAlarmDataSizeEstimation,$

где 86400 – число секунд в сутках,

**AvgTagFrequencyEstimation** – оценка средней частоты сохранения значений по тегу в историю (после фильтрации по чувствительности и с учетом принудительного повторного сохранения);

**AvgAlarmDataSizeEstimation** – оценка среднего удельного размера данных на один аларм.

### 1.1.2.4.3. Пример расчета



Пусть источник Source - нерезервированный (RedundancyScaleFactor = 1) экземпляр Astra.Server, который сохраняет историю значений по 1000 тегам (Source.NumItems = 1000) типа double. В сутки в совокупности (по всем тегам) в среднем сохраняется 100 000 000 записей (в среднем ~1 157 записей/с). История алармов не сохраняется. Срок активного хранения (DbSettings.ActiveStorageDepth) – 3 суток, общий срок хранения (DbSettings.StorageDepth) – 1000 суток. Сжатие отключено (CompressRatio = 1.0). Размер полного идентификатора элемента (AvgItemIdSize) от Astra.Server – 40 байт.

Запись истории значений для типа double занимает в БД 25 байт. Соответственно, **оценка суточного объема данных в байтах от источника:**



$Source.PerDayVolumeSizeEstimation = 100\,000\,000 * 25 = 2\,500\,000\,000$   
 $\approx 2\,384.2 \text{ МиБ} \approx 2.3 \text{ ГиБ}.$

**Общий размер базы данных ОРБД:**



$ОРБД = РХД + РСД = 2\,384\,247,4 + 48 = 2\,384\,295,4 \text{ МиБ} \approx 2\,328 \text{ ГиБ} \approx 2,273 \text{ ТиБ}.$

**Размер хранимых данных РХД:**



$РХД = РАР + РХДАР = 2\,377\,047,4 + 7200 = 2\,384\,247,4 \text{ МиБ}.$

**Размер архивного раздела РАР:**



$РАР = Source.PerDayVolumeSizeEstimation * (DbSettings.StorageDepth - DbSettings.ActiveStorageDepth) * CompressRatio = 2384,2 * (1000 - 3) / 1,0 = 2\,377\,047,4 \text{ МиБ}.$

## Размер хранимых данных активного раздела РХДАР:



$PXДAP = SUM(Align16MiB( Source.PerDayVolumeSizeEstimation ) * DbSettings.ActiveStorageDepth = Align16MiB( 2384,2 ) * 3 = 2400 * 3 = 7200 \text{ МиБ.}$

## Размер служебных данных РСД:



$PCД = PΦOI + PCДАО = 16 + 32 = 48 \text{ МиБ}$

## Размер файла основного индекса РΦОИ:



$PΦOI = Align16MiB( SUM( Source.NumItems ) * ( 4 + AvgItemIdSize ) ) = Align16MiB( 1000 * ( 4 + 40 ) ) = Align16MiB( 44000 ) = 16 \text{ МиБ.}$

## Размер хранимых данных активного раздела РСДАО:



$PCДАО = SUM( Source.RedundancyScaleFactor * ( Align16MiB( Source.NumItems * 16 ) + Align16MiB( Source.NumItems * 4096 ) ) = 1 * ( Align16MiB( 1000 * 16 ) + Align16MiB( 1000 * 4096 ) ) = 1 * ( Align16MiB( 16000 ) + Align16MiB( 4096000 ) ) = 1 * ( 16 + 16 ) = 32 \text{ МиБ.}$



При использовании сжатия при среднем коэффициенте сжатия 3.5 крат, размер архивного раздела при тех же параметрах сократится:



$PAP = Source.PerDayVolumeSizeEstimation * ( DbSettings.StorageDepth - DbSettings.ActiveStorageDepth ) * CompressRatio = 2384,2 * ( 1000 - 3 ) / 3,5 = 679 \ 156,4 \text{ МиБ.}$

Тогда оценка совокупного размера БД будет следующей:

$ОРБД = 679 \ 156,4 + 7200 + 16 + 32 = 686 \ 404,4 \text{ МиБ} \approx 671 \text{ ГиБ} \approx 0,655 \text{ ТиБ.}$

## 1.1.2.5. Оценка объема временного хранилища очереди данных

В файловые очереди попадают записи, соответствующие параметру «Тип данных хранилища»: «история значений», «история событий», «история значений и событий», «имитационные данные». Если выбран вариант «история значений и событий», то в файловые очереди по этому хранилищу (для каждой БД она своя) пишутся и значения, и события.

Для **оценки размера файловой очереди**, в которой накоплены данные **за промежуток времени** (EQFS - Estimated Queue File Size), используется следующее выражение:

*f*

$$EQFS = Duration * ( AVHRI * AVHRS + AEHRI * AEHRS ) + SegmentSize,$$

где **Duration** – длительность временного интервала, за который данные накапливаются в файловой очереди;

**AVHRS** (Average Value History Record Size) – оценка среднего размера записи истории значений в файловой очереди;

**AVHRI** (Average Value History Record Intensity) – оценка средней интенсивности потока записей истории значений, сохраняемых в файловую очередь;

**AEHRS** (Average Event History Record Size) – оценка среднего размера записи истории событий в файловой очереди;

**AEHRI** (Average Event History Record Intensity) – оценка средней интенсивности потока записей истории событий, сохраняемых в файловую очередь;

**SegmentSize** – размер сегмента файловой очереди, на данный момент 32 МиБ;



Следует учитывать, что если для хранилища Astra.Historian настроено N баз данных, то приведенную оценку следует умножить на N, т.к. для каждой БД в текущей реализации создается отдельная файловая очередь.

**Оценка среднего размера записи файловой очереди (AVHRS и AEHRS):**



$AVHRS = QRHS + AvgValueHistoryBodySize;$   
 $AEHRS = QRHS + AvgEventHistoryBodySize,$

где **QRHS** – заголовок записи файловой очереди константного размера 54 байта (содержит полный идентификатор элемента Astra.Server, по которому сохраняется запись + служебные данные);

**AvgValueHistoryBodySize** - средний размер сериализованного тела записи истории значений;

**AvgEventHistoryBodySize** - средний размер сериализованного тела записи истории событий.



На одно значение в файловой очереди при отключенной метки времени приходится примерно 60 байт, на одно событие – 350 байт.

## 1.1.2.6. Оценка объема заполнения дискового пространства

Вычисление объема заполнения дискового пространства за период времени хранения без сжатия:



$$D = \sum (r * T * v),$$

где D – суммарный объем данных без сжатия, байт.

r – размер одной записи тега, байт.

T – время хранения, с.

v – частота изменения значения тега, Гц.

### Пример



10 тегов типа double (r = 25 байт) получают значения с частотой 1 Гц.  
50 тегов типа int4 (r = 21 байт) получают значения с частотой 2 Гц.

За 1 год диск заполнится на:

$$D = \sum (r * T * v) = 10 * (25 * 31\,536\,000 * 1) + 50 * (21 * 31\,536\,000 * 2) = 7884000000 + 66225600000 = 74109600000 \text{ байт} \approx 75 \text{ ГБ} \approx 69 \text{ ГиБ},$$

где T = 86 400 (число секунд в 1 сутках) \* 365 (число дней в году) = 31 536 000 с.



Приведенные расчеты применимы к расчету размера архива (содержимое папки archive). К фрагментам в активных областях (active/{...}) эти расчеты также можно применить, однако следует иметь в виду, что их размер выравнивается вверх на 16 МиБ с целью оптимизации записи (для сокращения издержек на обновление метаданных файловой системы производится упреждающее увеличение размеров активных фрагментов блоками по 16 МиБ).



В приведенной методике расчета не учитывается размер дополнительных служебных данных, используемых для поддержки внутренней структуры фрагментов и индексации их содержимого.

Издержки на хранение этих структур составляют не более 2% от полезного объема содержимого.



Следует учитывать, что все фрагменты базы данных имеют заголовок фиксированного размера 12 КБ.

Расчет дискового пространства для параметров, у которых не указана зона нечувствительности по времени зависит от средней совокупности интенсивности поступления записей.

## Пример



Имеется 100 000 сигналов, по которым раз в секунду приходит 5000 значений и эти записи формата float4 ( $r = 21$  байт). Рассчитать итоговый вес несжатых данных на диске за сутки.

$D = \sum (r * T * v) = 5000 * (21 * 86400 * 1) = 9072000000$  байт = 9,1 ГБ  $\approx$  8,5 ГиБ,

где  $T = 86\,400$  с (число секунд в 1 сутках).

Расчет дискового пространства для событий.

Типичный размер одного события примерно 200-300 байт (объем сильно зависит от размера сообщения, размера полного пути тега, по которому событие сработало).

## Пример



Средний размер события составляет 300 байт, средняя генерации – 1 раз в секунду. Рассчитать итоговый вес несжатых данных на диске за сутки.

$D = \sum (r * T * v) = \sum r * (T * v) = 300 * (86400 * 1) = 25920000$  байт  $\approx$  26 МБ  $\approx$  24,72 МиБ,


где  $\sum r$  – средний размер события;

$T = 86\,400$  с (число секунд в 1 сутках).



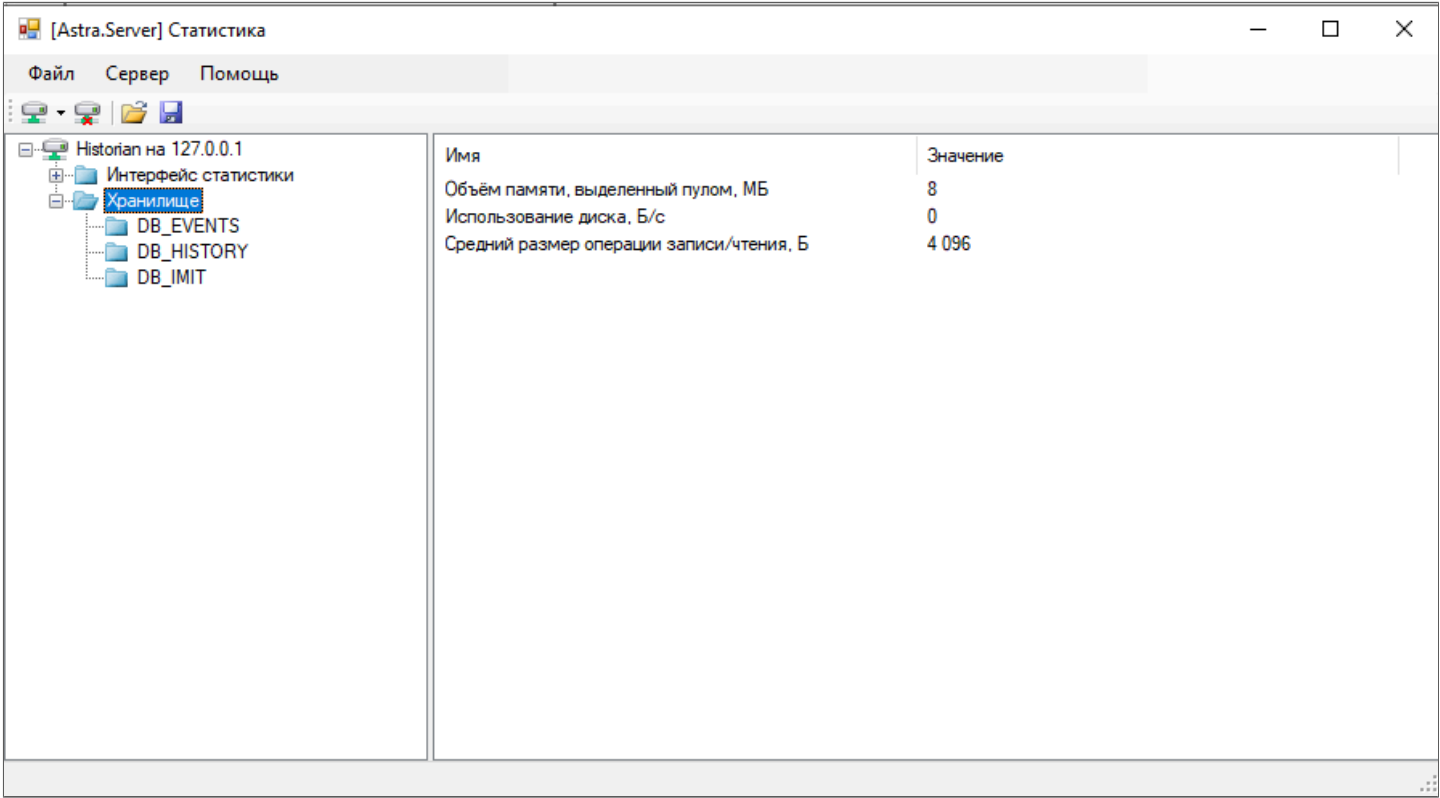
# 1.1.3. Диагностика работы

Для просмотра статистической информации Astra.Historian используется сервисное приложение [Статистика](#).



Для подключения к серверу истории используется порт StatPort, указанный в файле конфигурации.  
По умолчанию: 3388

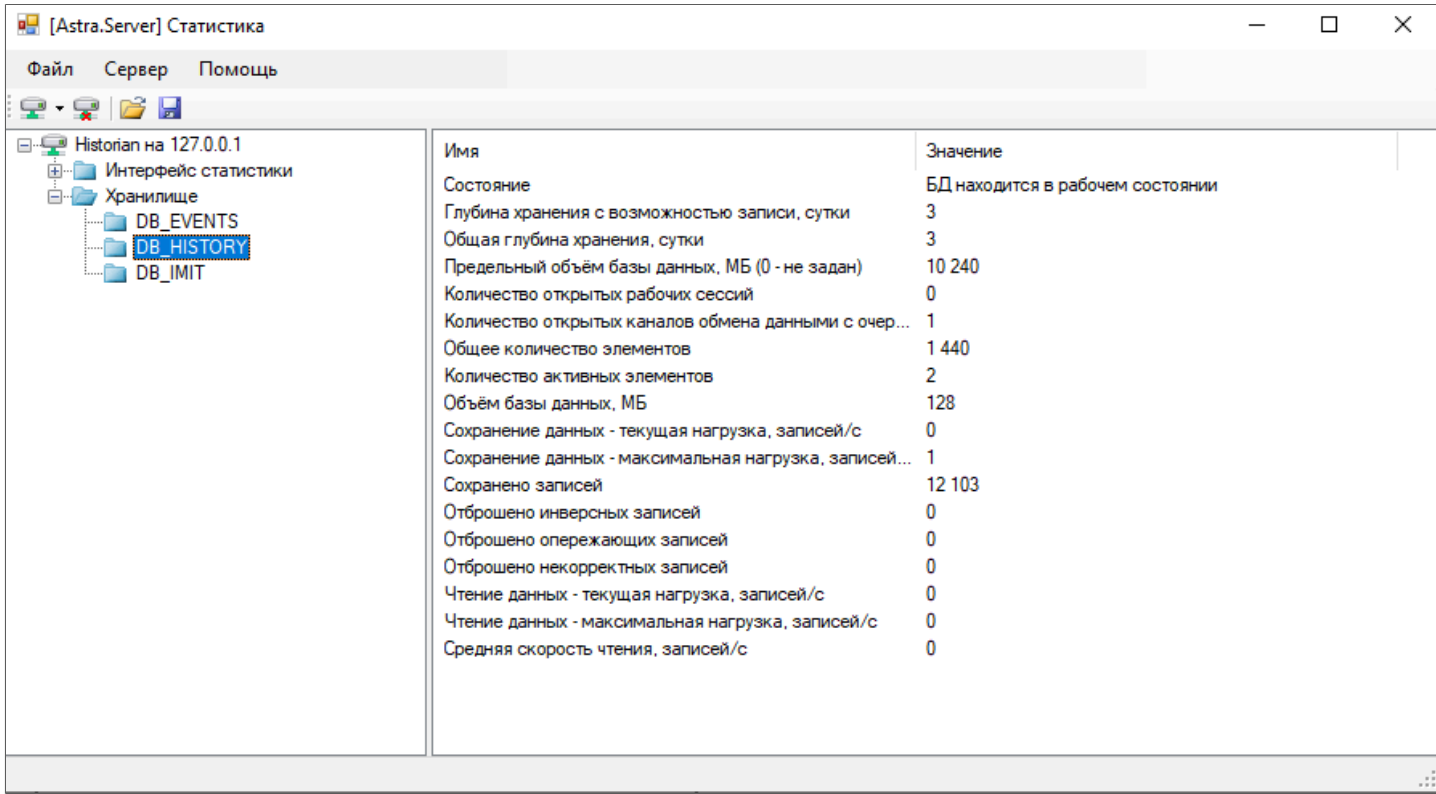
Узел статистики Хранилище содержит статистическую информацию о работе сервера истории.



Парметр	Описание
Объем памяти, выделенный пулом, МБ	Объем оперативной памяти, используемой в качестве кэша для работы с базами данных в текущий момент времени
Использование диска, Б/с	Нагрузка на диск. Рассчитывается средствами сервера истории.

Средний размер операции записи/чтения, Б	Средний объем данных, запрашиваемый/записываемый в рамках одной операции чтения/записи с диска.
--	---

Чтобы посмотреть статистические данные о подключенной БД, выберите узел статистики  
Хранилище → <Название БД>.



Параметр	Описание
Состояние	Состояние связи сервера истории с БД: БД находится в рабочем состоянии БД завершает работу БД не инициализирована
Глубина хранения с возможностью записи, сутки	Значение атрибута ActiveStorageDepth базы данных в файле конфигурации
Общая глубина хранения, сутки	Значение атрибута StorageDepth базы данных в файле конфигурации
Предельный объем базы данных, МБ	Значение атрибута VolumeLimit базы данных в файле конфигурации. 0 - не задан

Количество открытых рабочих сессий	Количество транзакций записи в БД, выполняемых в текущий момент времени
Количество открытых каналов обмена данными с очередью БД	Количество каналов, открытых клиентами на запись в БД. Канал открывается клиентом при первой транзакции записи в сервер истории (все дальнейшие транзакции записи выполняются клиентом в открытом канале), закрывается при потере соединения клиента с сервером истории (в этом случае для продолжения записи будет открыт новый канал)
Общее количество элементов	Количество элементов, по которым в БД хранится информация
Количество активных элементов	Количество элементов, участвующих в транзакциях записи в текущий момент времени
Объем базы данных, МБ	Текущий объем места на диске, занимаемого базой данных
Сохранение данных - текущая нагрузка, записей/с	Количество записей, сохраненных в БД за секунду в текущий момент времени
Сохранение данных - максимальная нагрузка, записей/с	Максимальное значение предыдущего параметра за время функционирования сервера истории
Сохранено записей	Количество записей, сохраненных в БД с момента запуска сервера истории
Отброшено инверсных записей	Количество записей, которые не были записаны в БД из-за устаревшей метки времени (метка времени записи меньше метки времени последней сохраненной записи этого элемента)
Отброшено опережающих записей	Количество записей, которые не были записаны в БД из-за метки времени, опережающей текущее время сервера истории
Чтение данных - текущая нагрузка, записей/с	Количество записей, переданных в рамках всех транзакций чтения за секунду в текущий момент времени

Чтение данных - максимальная нагрузка, записей/с	Максимальное значение предыдущего параметра за время функционирования сервера истории
Средняя скорость чтения, записей/с	Средняя продолжительность выполнения запросов на чтение из БД

# Журнал работы

Для просмотра системных событий по компоненту используется приложение [EventLogViewer](#).

Журнал приложений

Источник	Время	Сообщение
Astra.Historian.Server	05.04.2023 08:25:57	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:24:57	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:23:57	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:22:56	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:21:56	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:20:56	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:19:56	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:18:55	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:17:55	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:16:55	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:15:54	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:14:54	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:13:54	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:12:54	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:11:54	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:10:53	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:09:53	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:08:53	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:07:53	Лицензия не обнаружена или ограничена. Использование Сервера Astr...
Astra.Historian.Server	05.04.2023 08:06:52	Лицензия не обнаружена или ограничена. Использование Сервера Astr...

Показаны 6306 событий из 31829

Настройка фильтра

☒ Источники

☐ Кроме выбранных

Выбрать все

Очистить

☐ AdobeARMService

☐ Application Error

☐ Application Hang

☐ Astra.AStudio

☐ Astra.Diagnostics.Hashcheck

☐ astra.domain.agent

☒ Astra.Historian.Server

☐ Astra.Licensing.Agent

☐ astra.net.agent

☐ astra.security.agent

☐ Astra.Server

☐ Astra.Server.AeServer

☐ Astra.Server.DaServer

☐ Astra.Server.HistoryModule

☐ Astra.Server.OpcUaClient

☐ Astra.Server.SNMP Manager

☐ Astra.Server.TcpServer

☐ Astra.Server.UaServer

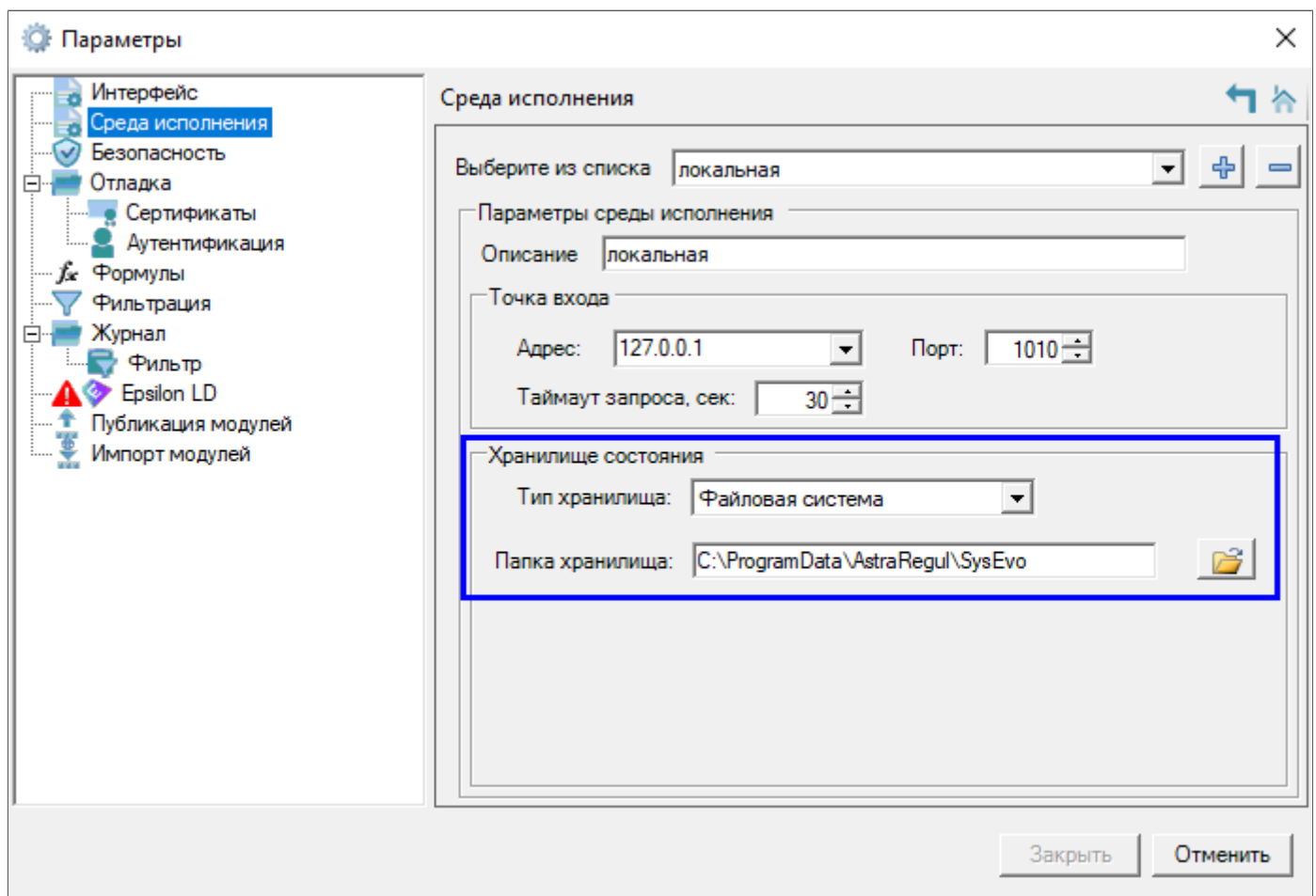
## 1.1.4. Администрирование

Данные сохраняются и запрашиваются в базу данных по внутренним идентификаторам сигналов.

Чтобы после переноса/восстановления базы данных из нее можно было получить данные, необходимо:

- Сохранить проект Astra.AStudio обязательно вместе с хранилищем состояния.

Расположение хранилища состояния указано в Astra.AStudio: Меню Файл → Параметры → Среда исполнения.



По умолчанию:



C:\ProgramData\AstraRegul\SysEvo.

- Сохранить конфигурацию экземпляра Astra.Server, который сохранял в нее данные.

После переноса/восстановления базы данных нужно применить сохраненную конфигурацию к тому экземпляру Astra.Server, который будет запрашивать сохраненные данные из базы данных.

## 1.1.4.1. Перенос баз данных

Чтобы перенести базу данных на другой компьютер:

1. У сервера истории, в который будет перенесена база данных, в файл конфигурации `Astra.Historian.Server.xml` добавьте описание новой базы данных.

Любые параметры новой базы данных могут отличаться от параметров исходной базы данных.



Если глубина хранения (атрибут `StorageDepth`) новой базы данных будет меньше, чем у исходной, то после переноса данные, которые окажутся старше новой глубины хранения, будут удалены.

2. Перезапустите службу (сервис) `Astra.Historian`.



При запуске сервер истории создаст структуру папок и файл новой базы данных.

3. Остановите службу (сервис) `Astra.Historian`, чтобы файл базы данных не был заблокирован для замены.

4. Скопируйте с заменой файл и папки исходной базы данных в новую базу данных в соответствии со структурой новой базы данных.

5. Если переносили базу данных в рамках одного сервера истории, из файла конфигурации удалите описание исходной базы данных.

6. Запустите службу (сервис) `Astra.Historian`.



## 1.1.4.2. Сохранение части архива базы данных

Архив или его части можно сохранить в отдельную папку или на внешний носитель информации. Сохраненные таким образом данные впоследствии можно восстановить.

Чтобы создать копию архивных данных:

1. Создайте папку.
2. Скопируйте в нее файл db.index из структуры базы данных.
3. Скопируйте в нее папку archive из структуры базы данных.

В ней данные сгруппированы в подпапки по датам и источникам, если источников более одного. В копии архива можно оставить папки только от интересующих вас источников и за нужные вам даты.

После копирования, из папки archive базы данных можно удалить скопированные подпапки, если для текущей работы они не нужны.



Файл db.index и папку archive базы данных удалять **нельзя**.

Чтобы дополнить созданную ранее копию архивных данных:

1. Скопируйте с заменой файл db.index из базы данных в копию архива.
2. В копию архива скопируйте подпапки с данными от интересующих вас источников и за нужные Вам даты.

## 1.1.4.3. Восстановление сохраненных архивных данных для просмотра

Чтобы восстановить сохраненные архивные данные для просмотра, выполните следующие действия:

1. В файл конфигурации сервера истории `Astra.Historian.Server.xml` добавьте новую базу данных.



Для нее не указывайте глубину хранения (по умолчанию – не ограничена) или укажите неограниченную глубину хранения: **`StorageDepth="0"`**. В противном случае сервер истории при запуске может удалить данные, имеющие большую глубину относительно текущей даты.

2. Перезапустите службу (сервис) `Astra.Historian`.

В результате в файловой системе будут созданы файл и папки для новой базы данных.

3. Остановите службу (сервис) `Astra.Historian`.

4. Замените в папке базы данных файл `db.index` одноименным файлом из сохраненного архива.

5. Из папки `archive` сохраненного архива скопируйте папки интересующих вас источников и за нужные для просмотра даты в одноименную папку базы данных.

6. Запустите службу (сервис) `Astra.Historian`.

После этого восстановленные архивные данные можно запрашивать из добавленной базы данных.

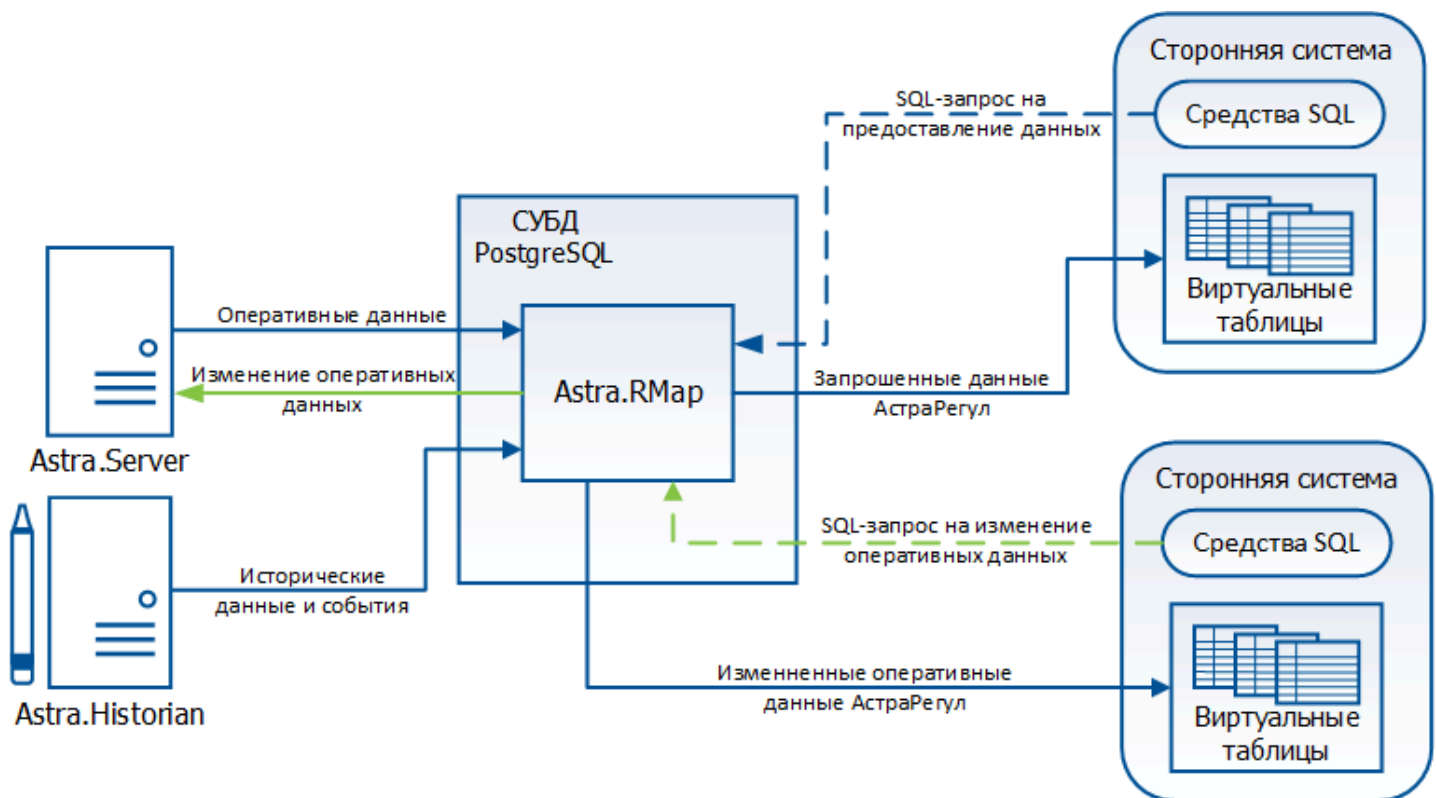
## 1.2. Astra.RMap

**Astra.RMap** – расширение СУБД PostgreSQL, с помощью которого возможно представление оперативных значений, истории значений и событий в реляционном виде.

Компонент предоставляет SQL-доступ к данным ПТК AstraRegul, с возможностью изменения оперативных значений.

Оперативные значения для Astra.RMap предоставляет Astra.Server, историю значений и событий - Astra.Historian.

Полученные данные Astra.RMap предоставляет СУБД PostgreSQL. СУБД PostgreSQL предоставляет полученные данные сторонним системам (например, Microsoft Excel, 1C, системам отчетности, SQL-клиентам и т.п.), в которых данные отображаются в виде виртуальных таблиц.



## 1.2.1. Настройка

[Windows](#)

[AstraLinux](#)

[РЕД ОС](#)

## 1.2.1.1. Windows

### Настройка драйвера ODBC

Настройку драйвера ODBC для PostgreSQL выполните в следующем порядке:

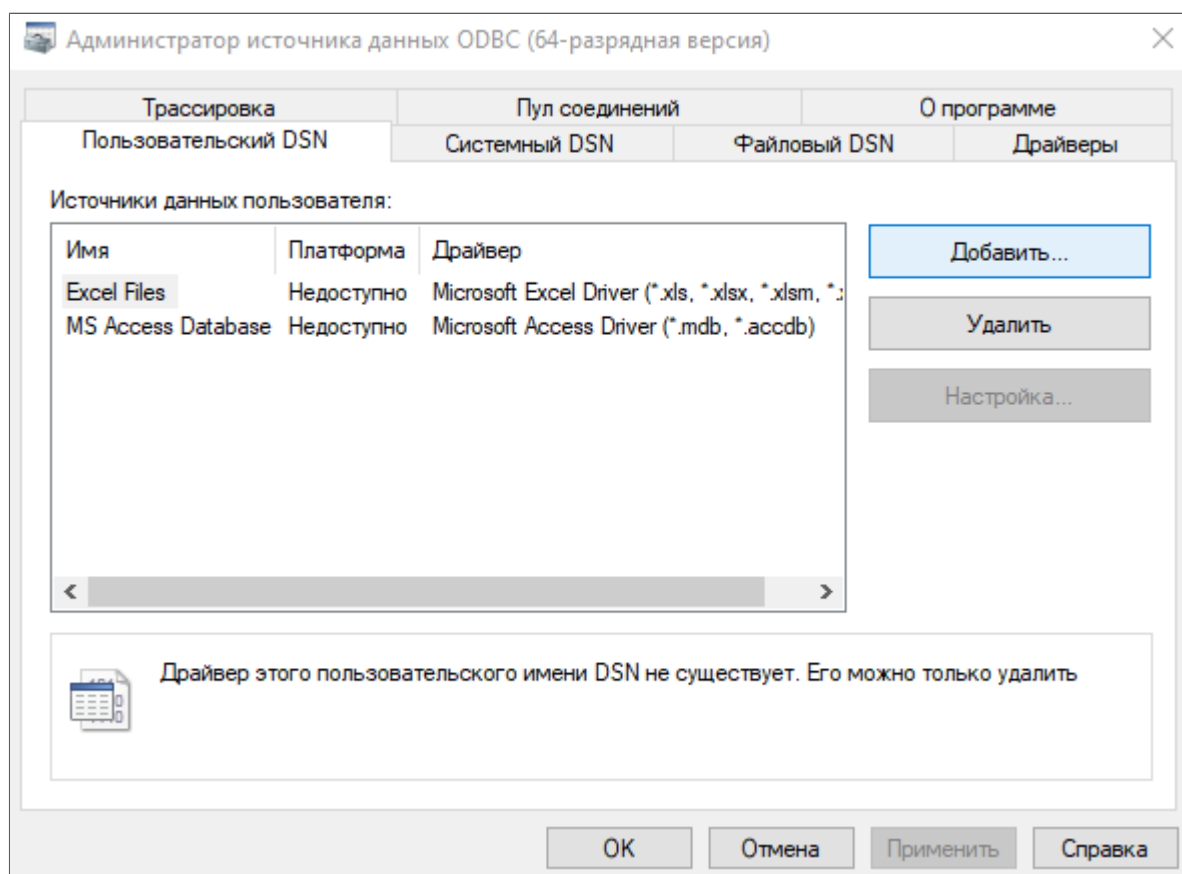
1. Выполните команду Пуск → Служебные → Панель управления → Система безопасности → Администрирование → Источники данных ODBC. Аналогичное действие можно выполнить, запустив файл `odbcad32.exe`, расположенный в папке `C:\Windows\System32`.

Приведенные способы актуальны в случае если разрядность ОС Windows совпадает с разрядностью программного продукта, в который будут предоставляться данные.

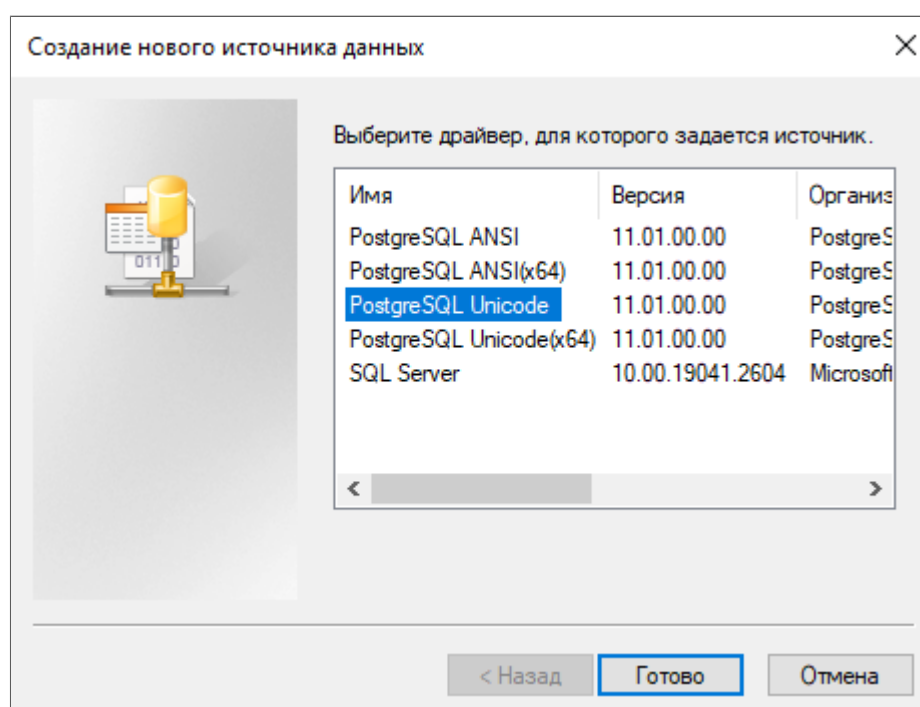


Если разрядность ОС Windows - 64 бита, а разрядность приложения - 32 бита, запустите файл `odbcad32.exe`, расположенный в папке `C:\Windows\SysWOW64`

2. В окне Администратор источников данных ODBC перейдите на вкладку Пользовательский DSN и нажмите кнопку Добавить...



В появившемся окне Создание нового источника данных выберите драйвер «PostgreSQL Unicode» и нажмите кнопку Готово.



3. В появившемся окне PostgreSQL Unicode ODBC Driver (psqlODBC) Setup укажите:

- › Database - имя базы данных PostgreSQL;

- › Server - адрес сервера;
- › User Name - имя пользователя для подключения к серверу PostgreSQL;
- › Password - пароль пользователя.

PostgreSQL Unicode ODBC Driver (psqlODBC) Setup

Data Source: PostgreSQL35W Description:

Database: postgres SSL Mode: disable

Server: localhost Port:

User Name: postgres Password:

Options: Datasource Global

Test Save Cancel

4. После произведенных настроек драйвер для Postgres будет готов к работе:

Администратор источника данных ODBC (64-разрядная версия)

Трассировка Пул соединений О программе

Пользовательский DSN Системный DSN Файловый DSN Драйверы

Установленные драйверы ODBC:

Имя	Версия	Организация	Файл
PostgreSQL ANSI	11.01.00.00	PostgreSQL Global Development Group	PSQLODBC30A.DLL
PostgreSQL ANSI(x64)	11.01.00.00	PostgreSQL Global Development Group	PSQLODBC30A.DLL
PostgreSQL Unicode	11.01.00.00	PostgreSQL Global Development Group	PSQLODBC35W.DLL
PostgreSQL Unicode(x64)	11.01.00.00	PostgreSQL Global Development Group	PSQLODBC35W.DLL
SQL Server	10.00.19041.2604	Microsoft Corporation	SQLSRV32.DLL

Драйвер ODBC позволяет программам ODBC получать сведения из источников данных ODBC. Он устанавливается с помощью программы установки.

OK Отмена Применить Справка

# Настройка Astra.RMAP



Настройки для работы Astra.RMap указываются в файле скрипта `init_rmap_history.sql`, который при выполнении формирует виртуальные таблицы для отображения данных.

1. Откройте файл скрипта `init_rmap_history.sql` любым текстовым редактором.

Каталог по умолчанию:



`C:\Program Files\AstraRegul\Astra.RMap\Initialize`

2. Если пользователей СУБД несколько, пропишите команды SQL `CREATE USER MAPPING` для каждого пользователя:

```
init_rmap_history — Блокнот
Файл  Правка  Формат  Вид  Справка

    ActorId          text NOT NULL,
    AckComment       text
)
SERVER history_server;

CREATE USER MAPPING FOR "postgres"
SERVER history_server;

CREATE USER MAPPING FOR "admin"
SERVER history_server;

CREATE USER MAPPING FOR "user"
SERVER history_server;
```



## 1.2.1.2. AstraLinux

### Настройка драйвера ODBC

1. Откройте файл конфигурации драйвера с помощью команды



```
sudo nano /etc/odbcinst.ini
```

```
user@user:~$ sudo nano /etc/odbcinst.ini
```

2. Файл должен содержать следующие записи о драйверах PostgreSQL ANSI (если требуется кодировка ANSI) и PostgreSQL Unicode (если требуется кодировка Unicode).



```
[PostgreSQL ANSI]
```

```
Description=PostgreSQL ODBC driver (ANSI version)
```

```
Driver=/usr/lib/x86_64-linux-gnu/odbc/psqlodbc.so
```

```
Setup=libodbcpsqlS.so
```

```
Debug=0
```

```
CommLog=1
```

```
UsageCount=1
```

```
[PostgreSQL Unicode]
```

```
Description=PostgreSQL ODBC driver (Unicode version)
```

```
Driver=/usr/lib/x86_64-linux-gnu/odbc/psqlodbcw.so
```

```
Setup=libodbcpsqlS.so
```

```
Debug=0
```

```
CommLog=1
```

```
UsageCount=1
```

Если их нет, то создайте их.

```
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
Driver=psqlodbc.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1

[PostgreSQL Unicode]
Description=PostgreSQL ODBC driver (Unicode version)
Driver=/usr/lib/x86_64-linux-gnu/odbc/psqlodbcw.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

3. Создайте подключение к базе данных. Откройте файл `odbc.ini` с помощью команды :



```
sudo nano /etc/odbc.ini
```

```
user@user:~$ sudo nano /etc/odbc.ini
```

Файл должен содержать следующие записи.



```
[PostgreSQL-connector]
Description    = PostgreSQL connection
Driver        = PostgreSQL Unicode
Database      = user_db //укажите имя БД
Servername    = localhost
Username      = user
Password      = MyPassword1 //укажите пароль
Port          = 5432
ReadOnly      = No
RowVersioning = No ShowSystemTables = No ShowOidColumn = No
FakeOidIndex  = No
ConnSettings  =
```

Если их нет, то создайте их.

```
[PostgreSQL-connector]
Description = PostgreSQL connection
Driver = PostgreSQL Unicode
Database = postgres
Servername = localhost
Username = test_user
Password = 123
Port = 5432
Protocol = 9.6
ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```

4. Откройте файл `pg_hba.conf` с помощью команды:



```
sudo nano /etc/postgresql/11/main/pg_hba.conf
```

5. Файл `pg_hba.conf` должен содержать следующую запись:



```
host    all         all         ::1/128 md5
```



Если данная запись отсутствует, ее необходимо добавить в конец файла и перезапустить PostgreSQL.

```
local  all         postgres                                peer
# TYPE  DATABASE  USER  ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local  all         all                                peer
# IPv4 local connections:
#host   all         all         127.0.0.1/32      md5
host   all         all         0.0.0.0/0         md5
# IPv6 local connections:
host   all         all         ::1/128          md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local  replication  all                                peer
host   replication  all         127.0.0.1/32      md5
host   replication  all         ::1/128          md5
host   mydb1       test_user  all              md5
```

6. После этого перезапустите PostgreSQL с помощью команды:



```
sudo systemctl restart postgresql.service
```

```
user@user:~$ sudo systemctl restart postgresql.service
user@user:~$
```

# Настройка Astra.RMAP



Настройки для работы Astra.RMap указываются в файле скрипта `init_rmap_history.sql`, который при выполнении формирует виртуальные таблицы для отображения данных.

1. Произведите настройку Astra.RMap для этого откройте файл `init_rmap_history.sql` с помощью команды `sudo nano /usr/share/postgresql/11/extension/init_rmap_history.sql`.

```
user@user:~$ sudo nano /usr/share/postgresql/11/extension/init_rmap_history.sql
user@user:~$ █
```

Если пользователей СУБД несколько, пропишите команды SQL `CREATE USER MAPPING` для каждого пользователя:

```
CREATE USER MAPPING FOR "postgres"
SERVER history_server;

CREATE USER MAPPING FOR "admin"
SERVER history_server;

CREATE USER MAPPING FOR "user"
SERVER history_server;█
```

2. Настройте подключение к Astra.Server:

```
CREATE SERVER history_server TYPE 'astra.rmap.historian.rprovider'
FOREIGN DATA WRAPPER rmap_fdw
OPTIONS (
    SourceLocation '127.0.0.1',
    --SourceDB 'Astra.OPCDA Server',
    SourceTcpPort '4388',
    HistorianLocation '127.0.0.1',
    --HistorianName 'Astra.Historian.Server',
    HistorianTcpPort '4949',
    HistorianDB 'default',
    --SourceAE 'Astra.OPCAEServer',
    SourceAETcpPort '4388',
    AeHistorianLocation '127.0.0.1',
    --AeHistorianName 'Astra.Historian.Server',
    AeHistorianTcpPort '4949',
    AeHistorianDB 'aebase'
);
```

- `SourceLocation` – адрес Astra.Server;
- `SourceTcpPort` и `SourceAETcpPort` – порт модуля TCP Server, указанный в параметре конфигурации модуля Номер TCP порта. По умолчанию порт 4388.

3. Настройте подключение к Astra.Historian:

```
CREATE SERVER history_server TYPE 'astra.rmap.historian.rprovider'
FOREIGN DATA WRAPPER rmap_fdw
OPTIONS (
    SourceLocation '127.0.0.1',
    --SourceDA 'Astra.OPCDA_Server',
    SourceTcpPort '4388',
    HistorianLocation '127.0.0.1',
    --HistorianName 'Astra.Historian.Server',
    HistorianTcpPort '4949',
    HistorianDB 'default',
    --SourceAE 'Astra.OPCAEServer',
    SourceAETcpPort '4388',
    AeHistorianLocation '127.0.0.1',
    --AeHistorianName 'Astra.Historian.Server',
    AeHistorianTcpPort '4949',
    AeHistorianDB 'aebase'
);
```

- › HistorianLocation и HistorianTcpPort – адрес и порт Astra.Historian, в котором хранятся значения;
- › HistorianDB – имя базы данных, в которой хранятся значения;
- › AeHistorianLocation и AeHistorianTcpPort – адрес и порт Astra.Historian, в котором хранятся события;
- › AeHistorianDB – имя базы данных, в которой хранятся события.

Символы комментирования -- перед атрибутами HistorianTcpPort и AeHistorianTcpPort должны отсутствовать.

Атрибуты HistorianName и AeHistorianName должны быть закоментированы символами "--".

4. Для получения истории с помощью модуля TCP Server в значениях атрибутов HistorianTcpPort и AeHistorianTcpPort укажите порт доступа к истории, заданный в настройках модуля TCP Server в Astra.AStudio в свойстве Номер TCP порта группы Настройки доступа к истории.

Настройки доступа к истории	
Номер TCP порта	4949
Время удержания сессии	10
Ограничение ожидающих сессий	64

5. Если в настройках модуля порт не задан, то установите в соответствующем параметре модуля любой свободный порт.

```
CREATE SERVER history_server TYPE 'astra.rmap.historian.rprovider'
FOREIGN DATA WRAPPER rmap_fdw
OPTIONS (
    SourceLocation '127.0.0.1',
    --SourceDA 'Astra.OPCDA_Server',
    SourceTcpPort '4388',
    HistorianLocation '127.0.0.1',
    --HistorianName 'Astra.Historian.Server',
    HistorianTcpPort '4949',
    HistorianDB 'default',
    --SourceAE 'Astra.OPCAEServer',
    SourceAETcpPort '4388',
    AeHistorianLocation '127.0.0.1',
    --AeHistorianName 'Astra.Historian.Server',
    AeHistorianTcpPort '4949',
    AeHistorianDB 'aebase'
);
```

6. Запустите консольный клиент PostgreSQL. Для этого от суперпользователя root выполните команду:



```
sudo -u postgres psql
```

```
user@user:~$ sudo -u postgres psql
could not change directory to "/home/user": Отказано в гоступе
psql (11.17 (Debian 11.17-astra.sel+b1))
Type "help" for help.

postgres=#
```

7. Выполните команду инициализации скрипта для настройки Astra.RMAP:



```
\i /usr/share/postgresql/x.x/extension/init_rmap_history.sql
```

```
postgres=# \i /usr/share/postgresql/11/extension/init_rmap_history.sql
psql:/usr/share/postgresql/11/extension/init_rmap_history.sql:1: ЗАМЕЧАНИЕ: расширение "rmap_fdw" не существует, пропускается
DROP EXTENSION
CREATE EXTENSION
CREATE SERVER
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE FOREIGN TABLE
CREATE USER MAPPING
postgres=#
```

## 1.2.1.3. РЕД ОС

### Настройка драйвера ODBC

1. Откройте файл конфигурации драйвера с помощью команды:



```
sudo nano /etc/odbcinst.ini
```

2. Файл должен содержать следующие записи о драйверах PostgreSQL ANSI (если требуется кодировка ANSI) и PostgreSQL Unicode (если требуется кодировка Unicode).



```
[PostgreSQL]
Description=ODBC
for
PostgreSQL
Driver=/usr/pgsql-
x.x/lib/psqlodbcw.so
Setup=/usr/lib64/libodbcpsqlS.so
FileUsage=1
```



Если их нет, то создайте их.

3. Создайте подключение к базе данных. Откройте файл odbc.ini с помощью команды:



```
sudo nano /etc/odbc.ini
```

Файл должен содержать следующие записи.



```
[PostgreSQL-connector]
Description = PostgreSQL connection
Driver = PostgreSQL Unicode
Database = user_db //укажите имя БД
```

```
Servername = localhost
Username = user
Password = MyPassword1 //укажите пароль
Port = 5432 ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```



Если их нет, то создайте их.

4. Откройте файл `pg_hba.conf` с помощью команды:



```
sudo nano /var/lib/pgsql/11/data/pg_hba.conf
```

5. Файл `pg_hba.conf` должен содержать следующую запись:



```
host    all    all    ::1/128    md5
```



Если данная запись отсутствует, ее необходимо добавить в конец файла и перезапустить PostgreSQL.

6. После этого перезапустите PostgreSQL с помощью команды:



```
sudo systemctl restart postgresql.service
```

7. Выполните команду инициализации скрипта для настройки Astra.RMAP:



```
\i /usr/pgsql-11/share/extension/init_rmap_history.sql
```



## 1.2.2. Виртуальные таблицы

При выполнении скрипта `init_rmap_history.sql` формируются виртуальные таблицы для отображения данных.

### Типы данных

Тип данных	Описание
<code>bigint</code>	Знаковое целое из 8 байт $[-9 \times 10^{-18}; 9 \times 10^{18}]$
<code>boolean</code>	Логическое значение <code>[true; false]</code>
<code>double precision</code>	Значение с плавающей точкой из 8 байт $[\pm 5.0 \times 10^{-324}; \pm 1.7 \times 10^{308}]$ , точность 15-17 цифр
<code>integer</code>	Знаковое целое из 4 байт <code>[-2 147 483 648; 2 147 483 647]</code>
<code>smallint</code>	Знаковое целое из 2 байт <code>[32 768; 32 767]</code>
<code>text</code>	Символьная строка переменной длины
<code>timestamp</code>	Дата и время

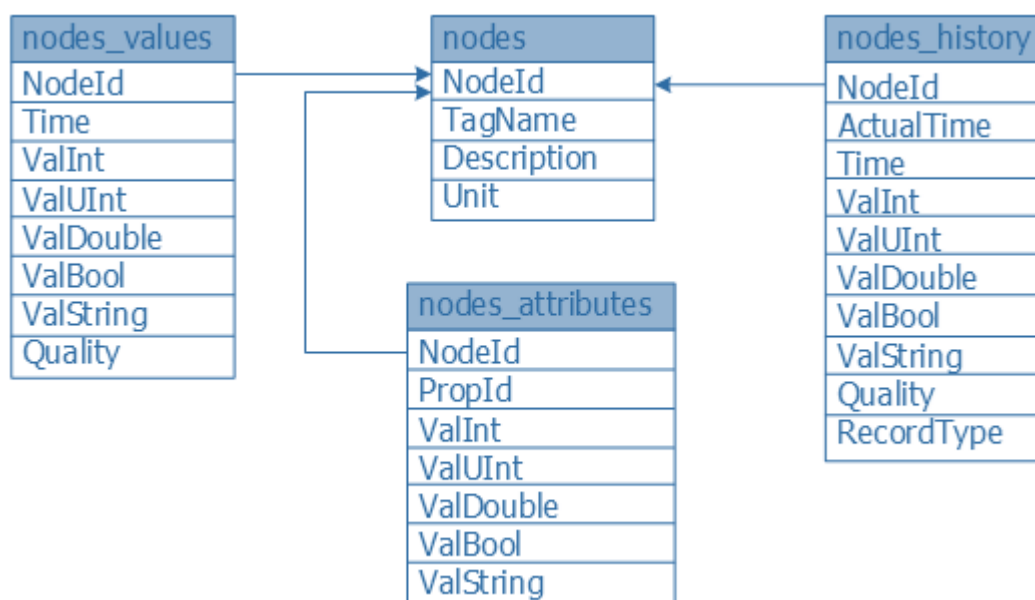
## 1.2.2.1. Таблицы данных

К виртуальным таблицам данных Astra.RMap относятся:

- › nodes – сигналы;
- › nodes\_values – оперативные значения;
- › nodes\_history – история значений;
- › nodes\_attributes – свойства сигналов.

### Связи таблиц

В таблицах nodes\_values, nodes\_history и nodes\_attributes поля NodeId являются внешними ключами и ссылаются на таблицу nodes.



### nodes

Отображает сигналы, по которым возможен просмотр данных.

Столбец	Тип данных	Описание
NodeId	bigint	Идентификатор сигнала
TagName	text	Имя сигнала
Description	text	Описание сигнала
Unit	text	Единицы измерения

## nodes\_values

Отображает оперативные значения сигналов.

Столбец	Тип данных	Описание
NodeId	bigint	Идентификатор сигнала
Time	timestamp	Дата и время
ValInt	bigint	Значение сигнала (знаковое)
ValUInt	bigint	Значение сигнала (беззнаковое)
ValDouble	double	Значение сигнала (с плавающей точкой)
ValBool	boolean	Значение сигнала (логическое)
ValString	text	Значение сигнала (текстовое)
Quality	integer	Качество



В зависимости от типа данных значение сигнала отображается в соответствующем столбце.

## nodes\_history

Отображает историю значений сигналов.

Столбец	Тип данных	Описание
NodeId	bigint	Идентификатор сигнала
ActualTime	timestamp	Действительная метка времени сигнала. Для всех значений внутри временного интервала совпадает с полем Time, для граничных точек содержит действительную метку времени границы
Time	timestamp	Дата и время
ValInt	bigint	Значение сигнала (знаковое)

ValUInt	bigint	Значение сигнала (беззнаковое)
ValDouble	double	Значение сигнала (с плавающей точкой)
ValBool	boolean	Значение сигнала (логическое)
ValString	text	Значение сигнала (текстовое)
Quality	integer	Качество
RecordType	text	Запрашиваемый тип значения. Принимаемые значения: > lbound - значение является левой граничной точкой > ubound - значение является правой граничной точкой > inner - значение является точкой внутри интервала



В зависимости от типа данных значение сигнала отображается в соответствующем столбце.

## nodes\_attributes

Отображает свойства сигналов.

Столбец	Тип данных	Описание
NodeId	bigint	Идентификатор сигнала
PropId	bigint	Идентификатор свойства сигнала
ValInt	bigint	Значение свойства (знаковое)
ValUInt	bigint	Значение свойства (беззнаковое)
ValDouble	double	Значение свойства (с плавающей точкой)
ValBool	boolean	Значение свойства (логическое)
ValString	text	Значение свойства (текстовое)



В зависимости от типа данных значение свойства сигнала отображается в соответствующем столбце.



Astra.RMap преобразует значения свойства CDT (Канонический тип данных, PropId = 1), получаемые от Astra.Server, к собственным кодам типов.

В таблице приведено соответствие значений свойства 1 (CDT) кодам типов в Astra.RMap:

Тип сигнала	Int1	UInt1	Int2	UInt2	Int4	UInt4	Int8	UInt8	Float	Double	Bool	String	Time
Значение 1 (CDT)	1	3	9	8	7	6	13	12	14	15	5	17	18
Код	16	17	2	18	3	19	20	21	4	5	11	8	7

## 1.2.2.2. Таблицы событий

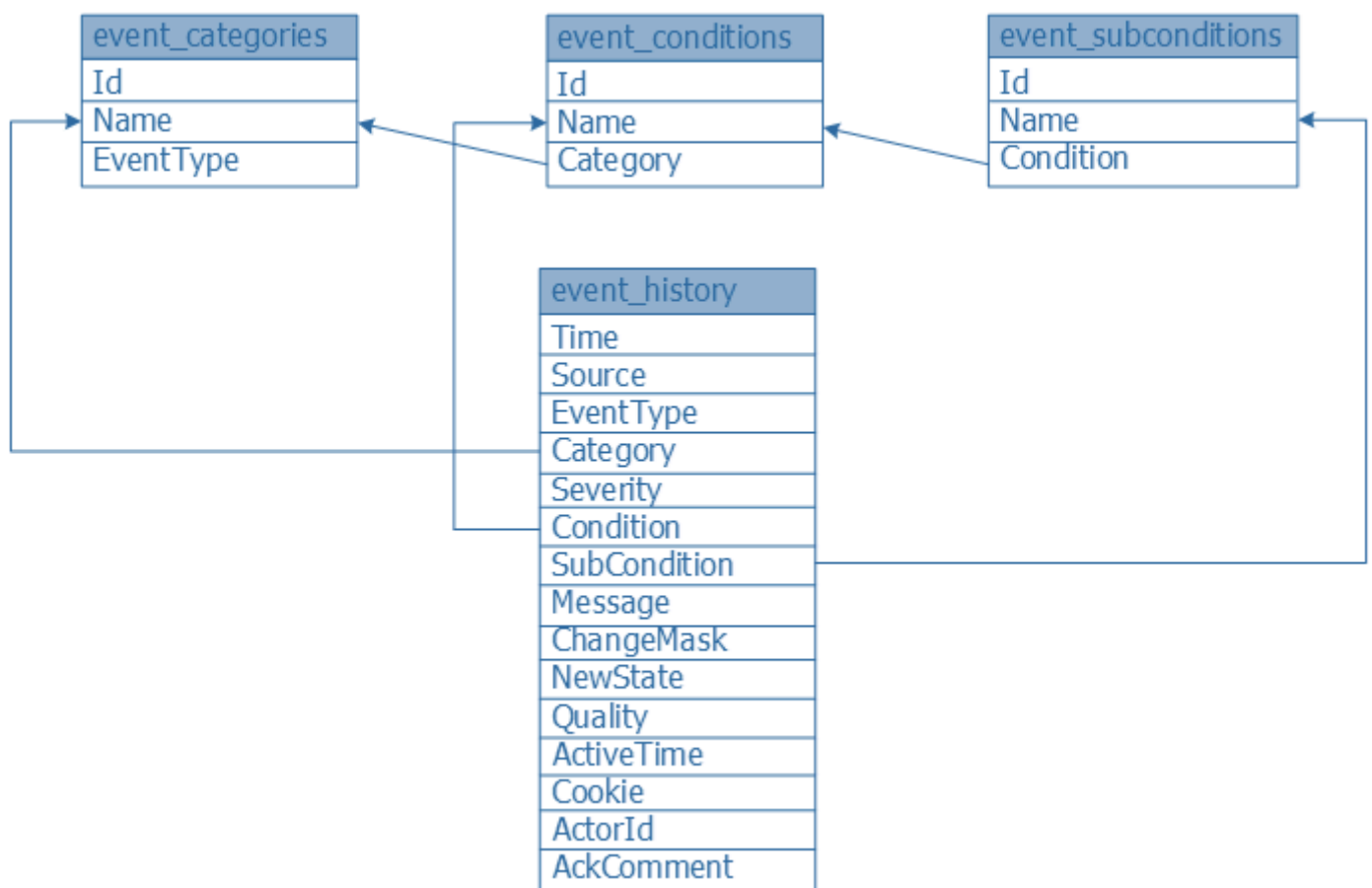
К виртуальным таблицам событий Astra.RMap относятся:

- › event\_categories – доступные категории событий;
- › event\_conditions – доступные условия событий;
- › event\_subconditions – доступные подусловия событий;
- › event\_history – история событий.

### Связи таблиц

В таблице event\_history поле Category является внешним ключом и ссылается на поле Name таблицы event\_categories. Аналогично, поля Condition и SubCondition ссылаются на поля Name таблиц event\_conditions и event\_subconditions соответственно.

В таблице event\_subconditions поле Condition ссылается на поле Name таблицы event\_conditions, в которой поле Category ссылается на поле Name таблицы event\_categories.



event\_categories

Отображает доступные категории событий.

Столбец	Тип данных	Описание
Id	bigint	Идентификатор категории событий
Name	text	Название категории событий
EventType	text	Тип события, к которому относится категория

## event\_conditions

Отображает доступные условия событий.

Столбец	Тип данных	Описание
Id	bigint	Идентификатор условия событий
Name	text	Название условия событий
Category	text	Категория, к которой относится условие

## event\_subconditions

Отображает доступные подусловия событий.

Столбец	Тип данных	Описание
Id	bigint	Идентификатор подусловия событий
Name	text	Название подусловия событий
Condition	text	Условие генерации события, к которому относится подусловие

## event\_history

Отображает историю событий.

Столбец	Тип данных	Описание
Time	timestamp	Время, когда сгенерировано уведомление о событии
Source	text	Полное имя сигнала, от которого произошло событие
EventType	text	Тип события
Category	text	Категория события
Severity	integer	Приоритет – важность данного события
Condition	text	Имя условия генерации события
SubCondition	text	Имя подусловия генерации события
Message	text	Сообщение – текст, который описывает событие
ChangeMask	smallint	Маска изменений
NewState	smallint	Индикатор нового состояния условия
Quality	integer	Качество
ActiveTime	timestamp	Время возникновения события
Cookie	integer	Текстовая информация о событии, хранящаяся в сервере
ActorId	text	Идентификатор OPC клиента, который квитирует событие
AckComment	text	Комментарий квитирования



## 1.2.3. SQL-запросы

### Общие рекомендации по построению SQL-запросов

1. При построении SQL-запросов с объединением таблиц рекомендуется использовать оператор JOIN без дополнительных операндов. Общий вид запроса:



```
SELECT {столбцы} FROM {таблица_1} JOIN {таблица_2} ON  
{условие_объединения} WHERE {условие_запроса}
```

2. Объединять таблицы рекомендуется в следующем порядке:

- в качестве {таблица\_1} использовать таблицы nodes\_values при запросе оперативных значений и nodes\_history при запросе истории значений;
- в качестве {таблица\_2} использовать таблицу nodes.

3. Для получения оперативных значений или истории значений по имени сигнала рекомендуется использовать запрос следующего вида:



```
SELECT {столбцы} FROM {таблица_1} JOIN nodes ON  
nodes.NodeId={таблица_1}.NodeId WHERE nodes.TagName =  
'{Имя_сигнала}'
```

### SQL-запросы к таблицам данных

Для изменения оперативных значений используется запрос UPDATE к таблице nodes\_values. Например, запрос на изменение значения сигнала NPS.BIK172\_6.BB97n.Cmd на значение 10 типа integer имеет вид:



```
UPDATE nodes_values SET valint=10 FROM nodes WHERE  
nodes.nodeid=nodes_values.nodeid AND nodes.TagName =  
'NPS.BIK172_6.BB97n.Cmd'
```



Если новое значение сигнала не совпадает с типом сигнала, происходит попытка преобразования типа. Если преобразование невозможно, запрос завершается ошибкой.

Для вывода данных таблиц используется запрос SELECT. Например, запрос на получение текущего значения сигнала NPS.BIK172\_6.BB97n.Cmd имеет вид:



```
SELECT * FROM nodes_values JOIN nodes ON  
nodes.NodeId = nodes_values.NodeId WHERE  
nodes.TagName='NPS.BIK172_6.BB97n.Cmd'
```

Запрос на получение истории значений сигнала NPS.BIK172\_6.BB97n.Cmd имеет вид:



```
SELECT * FROM nodes_history JOIN nodes ON nodes.NodeId  
= nodes_history.NodeId WHERE nodes.TagName =  
'NPS.BIK172_6.BB97n.Cmd'
```

## SQL-запросы к таблицам событий

Для вывода данных таблиц используется запрос SELECT. Например, запрос на получение истории событий, у которых условия генерации события DISCRETE, имеет вид:



```
SELECT * FROM event_history JOIN event_conditions ON  
event_history.Condition=event_conditions.Name WHERE  
Name='DISCRETE'
```

## 1.2.3.1. Примеры

1. Запрос на получение истории изменения значений сигнала NPS.MNS1.PT001\_1.Value.



```
SELECT * FROM nodes_history h JOIN nodes n ON n.NodeId=h.NodeId
WHERE n.TagName='NPS.MNS1.PT001_1.Value'
```

Результат запроса в MS Excel (значения сигнала отображаются в столбце valdouble):

nodeid	actualtime	time	valint	valuint	valdouble	valbool	valstring	quality	recordtype	nodeid2	tagname	description	unit
457	23.06.2021 4:47	23.06.2021 4:47			23			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:47	23.06.2021 4:47			25			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:48	23.06.2021 4:48			69			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:48	23.06.2021 4:48			24			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			44			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			55			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			23			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			24			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			63			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			0			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			24			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			21			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	

2. Запрос на получение истории изменения значений сигнала NPS.MNS1.PT001\_1.Value за промежуток времени с 04:48 до 04:49 23 июня 2021 года (включая граничные значения):



```
SELECT * FROM nodes_history h JOIN nodes n ON n.NodeId=h.NodeId
WHERE (n.TagName='NPS.MNS1.PT001_1.Value') AND (h.time BETWEEN
'2021-06-23 04:48' AND '2021-06-23 04:49')
```

Результат запроса в MS Excel (в столбце recordtype отображается тип значения - граничная точка или значение внутри интервала):

nodeid	actualtime	time	valint	valuint	valdouble	valbool	valstring	quality	recordtype	nodeid2	tagname	description	unit
457	23.06.2021 4:47	23.06.2021 4:48			25			216	lbound	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:48	23.06.2021 4:48			69			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:48	23.06.2021 4:48			24			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:49	23.06.2021 4:49			44			216	ubound	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	

3. Запрос на получение истории изменения значений сигнала NPS.MNS1.PT001\_1.Value за промежуток времени с 04:48 до 04:49 23 июня 2021 года (без граничных значений):



```
SELECT * FROM nodes_history h JOIN nodes n ON n.NodeId=h.NodeId
WHERE (n.TagName='NPS.MNS1.PT001_1.Value') AND (h.time BETWEEN
'2021-06-23 04:48' AND '2021-06-23 04:49') AND (recordType !=
'ubound') AND (recordType != 'lbound')
```

## Результат запроса в MS Excel:

nodeid	actualtime	time	valint	valuint	valdouble	valbool	valstring	quality	recordtype	nodeid	tagname	description	unit
457	23.06.2021 4:48	23.06.2021 4:48			69			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	
457	23.06.2021 4:48	23.06.2021 4:48			24			216	inner	457	NPS.MNS1.PT001_1.Value	Давление ДТ на приём МНС	

4. Для получения истории изменений состояний задвижки ZDV\_007\_8\_1 за промежуток времени с 04:30 до 04:50 29 января 2021 года создайте запрос с выборкой по столбцам Сообщение и Время возникновения события:



```
SELECT message, activetime FROM event_history eh WHERE (eh.source='NPS.UZA8.ZDV_007_8_1.Msgs') AND (eh.time BETWEEN '2021-01-29 04:30' AND '2021-01-29 04:50')
```

## Результат запроса в MS Excel:

message	activetime
УЗА №8. Задвижка №007.8.1. Открывается	29.01.2021 4:30
УЗА №8. Задвижка №007.8.1. Открытие невозможно. Нет напряжения в схеме управления	29.01.2021 4:30
УЗА №8. Задвижка №007.8.1. Открыта. Выполнение команды не требуется	29.01.2021 4:31
УЗА №8. Задвижка №007.8.1. Открывается в режиме имитации	29.01.2021 4:32
УЗА №8. Задвижка №007.8.1. Заккрытие невозможно. Авария задвижки	29.01.2021 4:32
УЗА №8. Задвижка №007.8.1. Открывается	29.01.2021 4:33

5. Запрос на получение текущего значения сигнала NPS.BIK172\_6.BB97n.Cmd:



```
SELECT * FROM nodes_values v JOIN nodes n ON n.NodeId=v.NodeId WHERE n.TagName='NPS.BIK172_6.BB97n.Cmd'
```

Результат запроса в MS Excel (текущее значение сигнала отображается в столбце valuint):

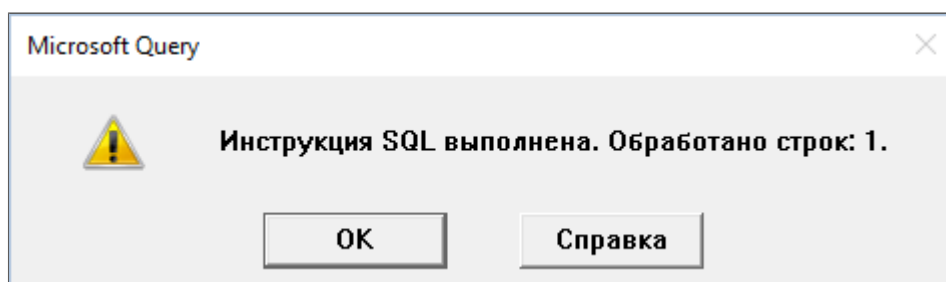
nodeid	time	valint	valuint	valdouble	valbool	valstring	quality	nodeid2	tagname
43071	01.02.2021 8:39		30				216		

6. Запрос на изменение текущего значения сигнала NPS.BIK172\_6.BB97n.Cmd на значение 100:



```
UPDATE nodes_values v SET valint=100 FROM nodes n WHERE n.nodeid=v.nodeid AND n.TagName='NPS.BIK172_6.BB97n.Cmd'
```

После выполнения данной команды программа, через которую осуществляется SQL-доступ, выдает сообщение о выполнении инструкции SQL. Пример сообщения программы Microsoft Query:



Результат запроса на получение текущего значения сигнала NPS.BIK172\_6.BB97n.Cmd (текущее (измененное) значение сигнала отображается в столбце valuint):

nodeid	time	valint	valuint	valdouble	valbool	valstring	quality	nodeid2	tagname
43071	01.02.2021 8:52		100					216	

7. Для получения истории изменений состояний всех задвижек узла UZA12 НПС создайте запрос с выборкой по столбцам Время возникновения события, Имя сигнала и Сообщение:



```
SELECT activetime, source, message FROM event_history WHERE source LIKE '%UZA12.%'
```

Результат запроса в MS Excel:

activetime	source	message
20.02.2021 8:12:51	NPS.UZA12.ZDV_007_12_4.Msgs	УЗА №12. Задвижка №007.12.4. Открывается
20.02.2021 8:12:56	NPS.UZA12.ZDV_007_12_6.Msgs	УЗА №12. Задвижка №007.12.6. В промежуточном положении
20.02.2021 8:13:33	NPS.UZA12.ZDV_007_12_4.Msgs	УЗА №12. Задвижка №007.12.4. Открытие невозможно. Авария задвижки
20.02.2021 8:13:37	NPS.UZA12.ZDV_007_12_6.Msgs	УЗА №12. Задвижка №007.12.6. Открытие невозможно. Нет напряжения в схеме управления
20.02.2021 8:14:30	NPS.UZA12.ZDV_007_12_4.Msgs	УЗА №12. Задвижка №007.12.4. Открыта. Выполнение команды не требуется
20.02.2021 8:14:34	NPS.UZA12.ZDV_007_12_6.Msgs	УЗА №12. Задвижка №007.12.6. Открыта. Выполнение команды не требуется
21.02.2021 7:52:00	NPS.UZA12.ZDV_007_12_3.Msgs	УЗА №12. Задвижка №007.12.3. Открыта. Выполнение команды не требуется
21.02.2021 7:52:05	NPS.UZA12.ZDV_007_12_5.Msgs	УЗА №12. Задвижка №007.12.5. Открыта. Выполнение команды не требуется
21.02.2021 7:52:13	NPS.UZA12.ZDV_007_12_3.Msgs	УЗА №12. Задвижка №007.12.3. Резерв
21.02.2021 7:52:18	NPS.UZA12.ZDV_007_12_5.Msgs	УЗА №12. Задвижка №007.12.5. Открывается
21.02.2021 7:52:25	NPS.UZA12.ZDV_007_12_4.Msgs	УЗА №12. Задвижка №007.12.4. Открыта в режиме имитации

8. Для получения истории событий по задвижкам, находящимся в состоянии Открыта, за промежуток времени с 00:00 20 февраля 2021 года до 23:00 21 февраля 2021 года создайте запрос с выборкой по столбцам Время возникновения события, Имя сигнала и Сообщение:



```
SELECT activetime, source, message FROM event_history eh WHERE (eh.time BETWEEN '2021-02-20 00:00' AND '2021-02-21 23:00') AND (message LIKE '%Открыта');
```

## Результат запроса в MS Excel:

activetime	source	message
20.02.2021 8:13:45	NPS.UZA8.ZDV_007_8_4.Msgs	УЗА №8. Задвижка №007.8.4. Открыта
21.02.2021 8:09:24	NPS.UZA8.ZDV_007_8_1.Msgs	УЗА №8. Задвижка №007.8.1. Открыта
21.02.2021 8:09:29	NPS.UZA12.ZDV_007_12_3.Msgs	УЗА №12. Задвижка №007.12.3. Открыта
21.02.2021 8:09:33	NPS.UZA12.ZDV_007_12_5.Msgs	УЗА №12. Задвижка №007.12.5. Открыта
21.02.2021 8:09:37	NPS.UZA8.ZDV_007_8_6.Msgs	УЗА №8. Задвижка №007.8.6. Открыта
21.02.2021 8:09:42	NPS.UZA12.ZDV_007_12_6.Msgs	УЗА №12. Задвижка №007.12.6. Открыта
21.02.2021 8:09:47	NPS.UZA13.ZDV_007_13_2.Msgs	УЗА №13. Задвижка №007.13.2. Открыта
21.02.2021 8:09:59	NPS.UZA12.ZDV_007_12_4.Msgs	УЗА №12. Задвижка №007.12.4. Открыта

## 1.2.4. Подключение по TCP

Чтобы подключиться к Astra.Server и Astra.Historian по TCP, укажите значения следующих атрибутов параметра OPTIONS команды SQL CREATE SERVER в файле скрипта init\_rmap\_history.sql:

- › SourceLocation – адрес Astra.Server.
- › SourceTcpPort – порт, на который настроен модуль TCP Server.
- › HistorianLocation – адрес Astra.Historian, в котором хранятся значения.
- › HistorianTcpPort – порт Astra.Historian, в котором хранятся значения, или порт доступа к истории модуля TCP Server;
- › HistorianDB – имя базы данных, в которой хранятся значения.
- › SourceAETcpPort – порт, на который настроен модуль TCP Server.
- › AeHistorianLocation – адрес Astra.Historian, в котором хранятся события.
- › AeHistorianTcpPort – порт Astra.Historian, в котором хранятся события, или порт доступа к истории модуля TCP Server.
- › AeHistorianDB – имя базы данных, в которой хранятся события.



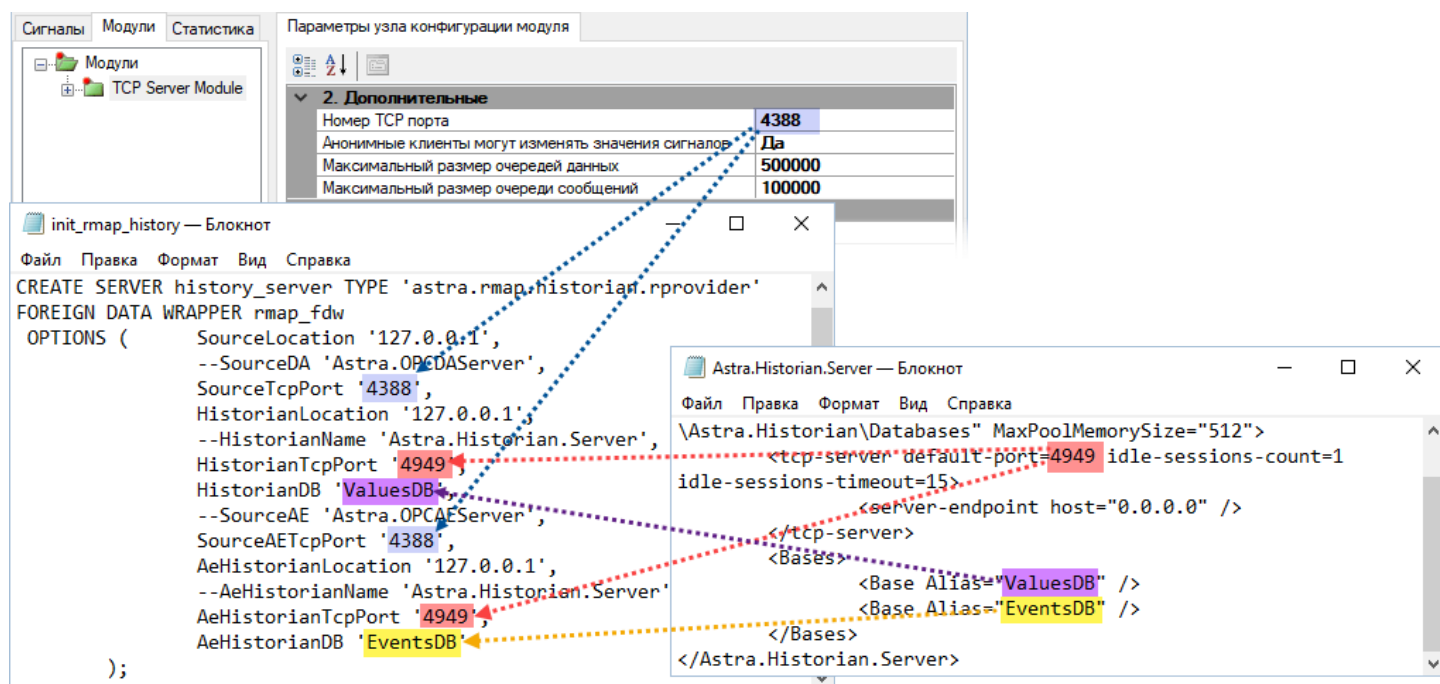
Символы комментирования "--" перед атрибутами SourceTcpPort, HistorianTcpPort, SourceAETcpPort, AeHistorianTcpPort должны отсутствовать.



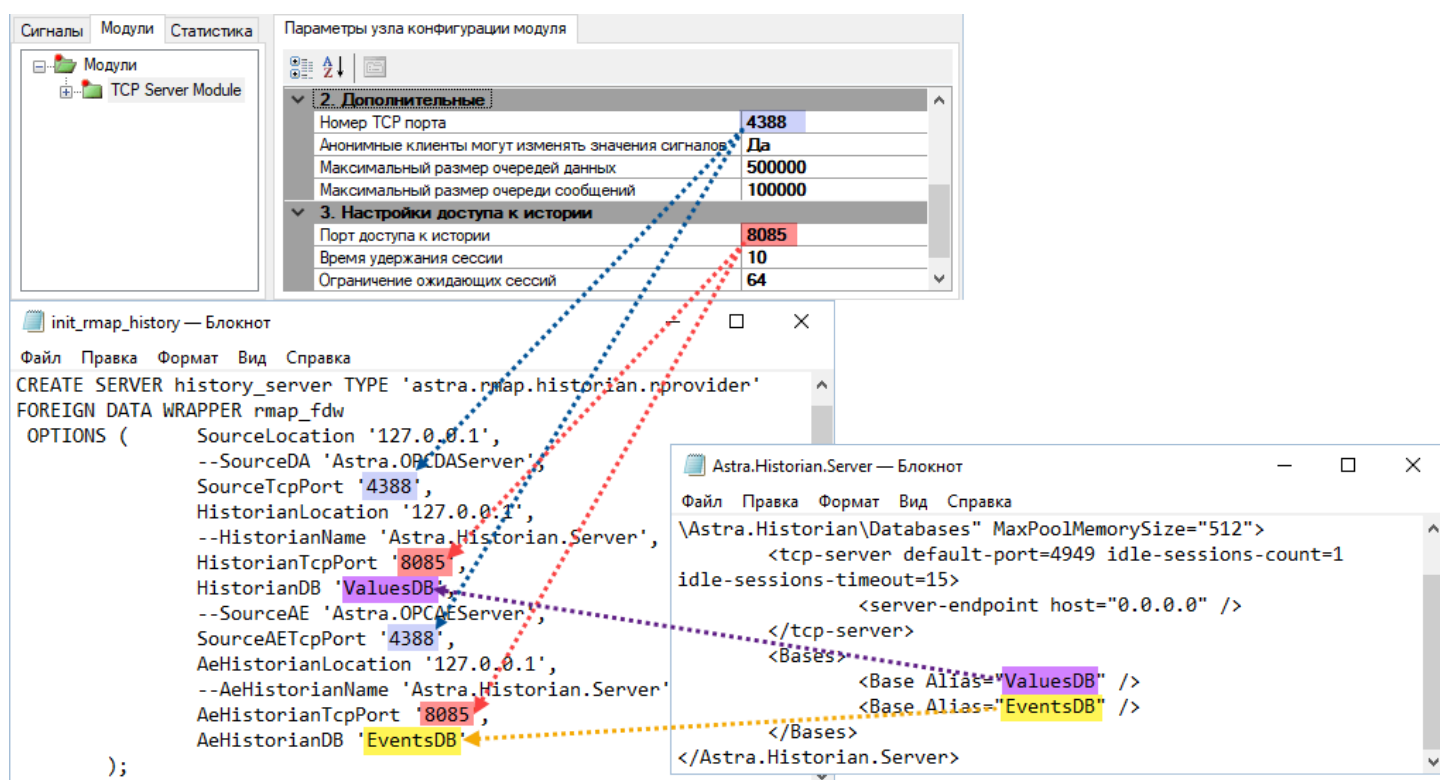
Атрибуты SourceDA, HistorianName, SourceAE, AeHistorianName должны быть закомментированы символами "--".

Параметры Astra.Historian, необходимые для настройки Astra.RMap, прописаны в конфигурационном файле Astra.Historian.Server.xml. Номера портов модуля TCP Server указаны в конфигурации Astra.Server.





Для получения истории с помощью модуля TCP Server в значениях атрибутов HistorianTcpPort и AeHistorianTcpPort укажите порт, заданный в параметре Порт доступа к истории модуля TCP Server. Если в настройках модуля порт не задан, установите в параметре Порт доступа к истории любой свободный порт.



После внесения изменений файл init\_rmap\_history.sql сохраните и исполните данный скрипт с помощью среды разработки и администрирования pgAdmin.



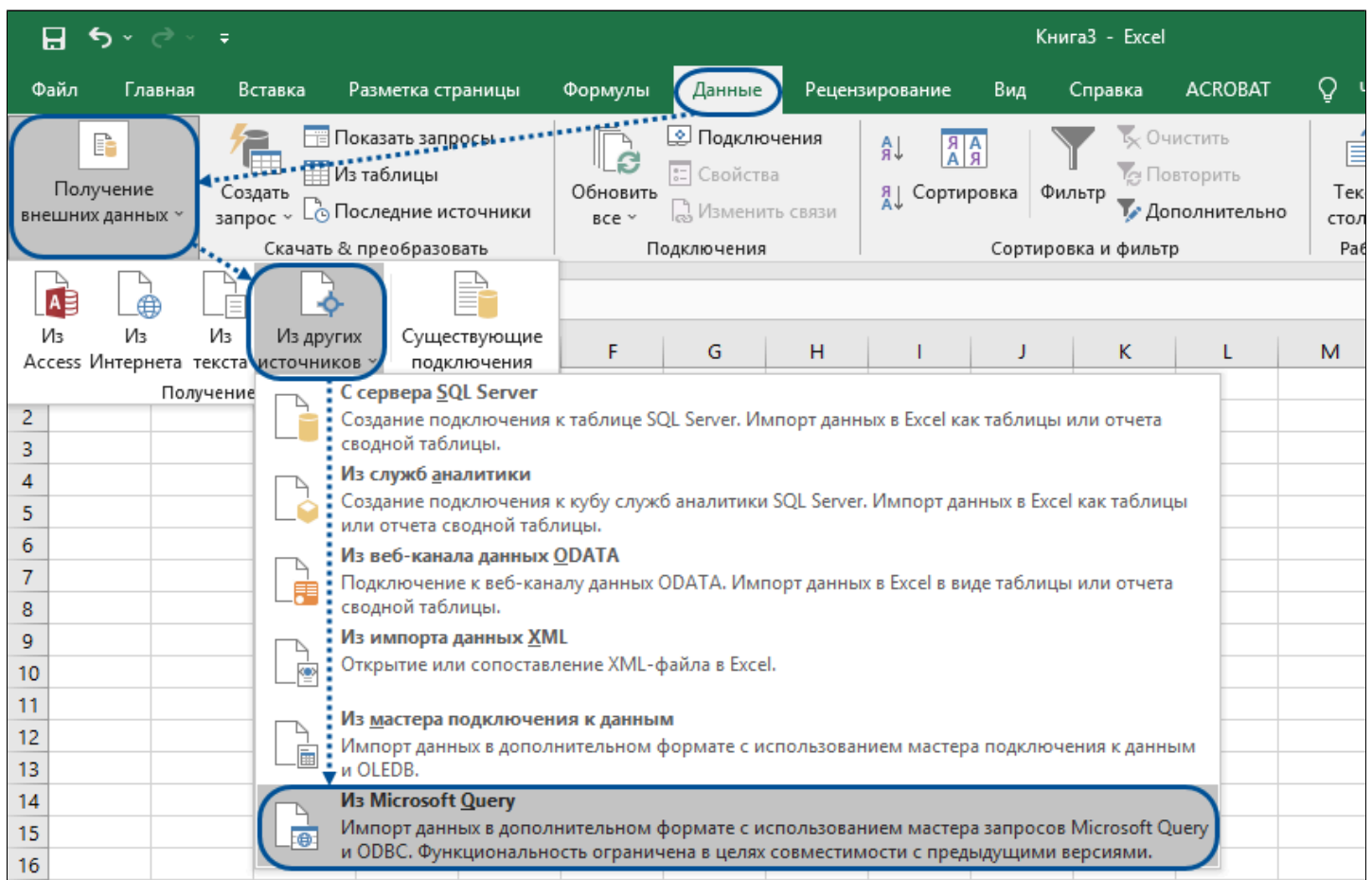
## 1.2.5. Предоставление данных в MS Excel



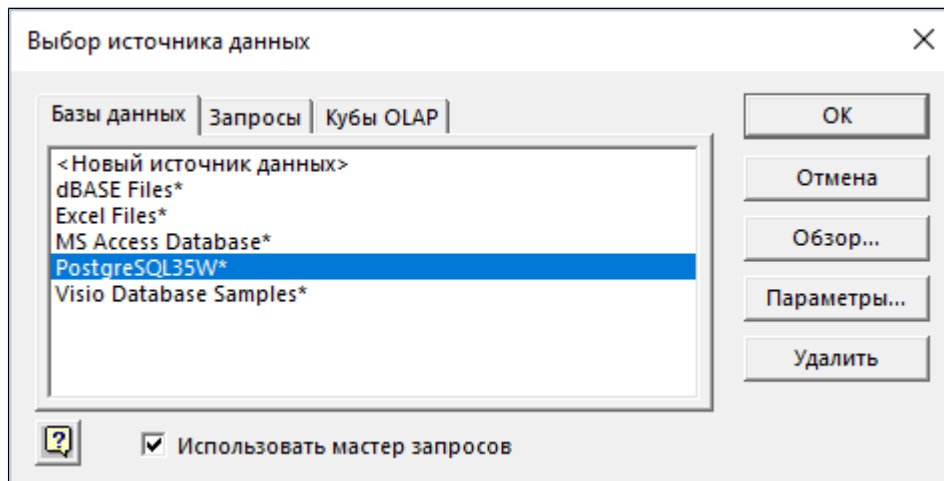
Для представления данных в MS Excel через Astra.RMap должен быть установлен драйвер ODBC для PostgreSQL.

### Порядок действий:

1. Запустите программу Microsoft Excel.
2. Создайте новую Книгу.
3. На вкладке Данные в области Получение внешних данных нажмите кнопку Из других источников и в выпадающем списке выберите Из Microsoft Query:



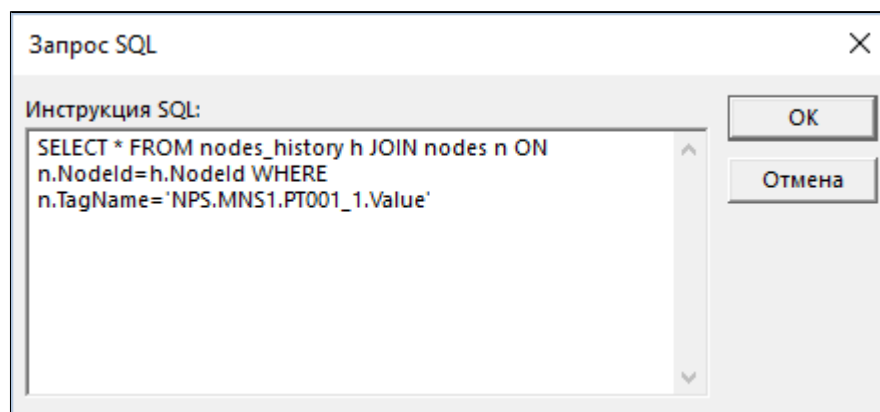
4. В появившемся окне Выбор источника данных на вкладке Базы данных выберите базу PostgreSQL и нажмите кнопку ОК:



В результате будет запущена программа Microsoft Query.

5. В окне программы Microsoft Query нажмите кнопку **Режим SQL**.

6. В появившемся окне "Запрос SQL" напишите SQL-запрос и нажмите кнопку ОК:



Результат запроса будет выведен в окне программы Microsoft Query:

Microsoft Query - [Запрос из PostgreSQL35W]

Файл Правка Вид Формат Таблица Условия Записи Окно Справка

SQL

	nodeid	actualtime	time	valint	valuint	valdouble	valbool	valstring	quality	
▶	4290016109	2021-06-23 04:47:56,14	2021-06-23 04:47:56,14			23			216	in
	4290016109	2021-06-23 04:47:59,49	2021-06-23 04:47:59,49			25			216	in
	4290016109	2021-06-23 04:48:01,73	2021-06-23 04:48:01,73			69			216	in
	4290016109	2021-06-23 04:48:04,37	2021-06-23 04:48:04,37			24			216	in
	4290016109	2021-06-23 04:49:29,61	2021-06-23 04:49:29,61			44			216	in
	4290016109	2021-06-23 04:49:33,16	2021-06-23 04:49:33,16			55			216	in
	4290016109	2021-06-23 04:49:35,66	2021-06-23 04:49:35,66			23			216	in
	4290016109	2021-06-23 04:49:37,65	2021-06-23 04:49:37,65			24			216	in
	4290016109	2021-06-23 04:49:40,05	2021-06-23 04:49:40,05			63			216	in
	4290016109	2021-06-23 04:49:41,90	2021-06-23 04:49:41,90			0			216	in
	4290016109	2021-06-23 04:49:44,32	2021-06-23 04:49:44,32			24			216	in
	4290016109	2021-06-23 04:49:46,39	2021-06-23 04:49:46,39			21			216	in

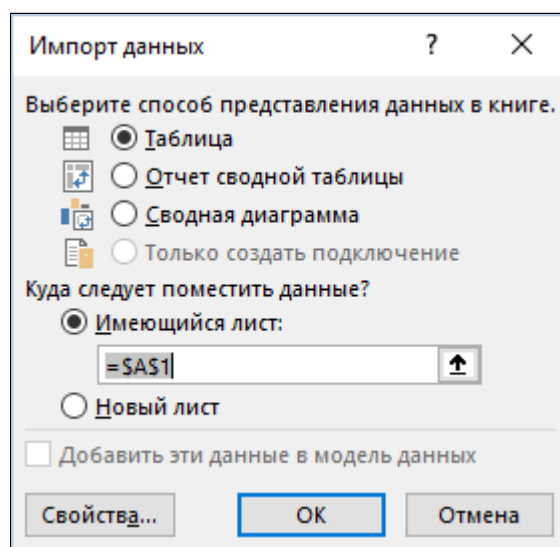
Запись: 1

Готово

NUM

7. Нажмите кнопку Вернуть данные.

8. В появившемся окне Импорт данных укажите параметры импорта данных и нажмите кнопку ОК:



Результаты SQL-запроса будут импортированы в Microsoft Excel:

nodeid	actualtime	time	valint	valuint	valdouble	valbool	valstring	quality	recordtype	nodeid2	tagname	description	unit
4290016109	23.06.2021 4:47	23.06.2021 4:47			23			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:48	23.06.2021 4:48			69			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:48	23.06.2021 4:48			24			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			44			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			55			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			23			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			24			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			63			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			0			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			24			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	
4290016109	23.06.2021 4:49	23.06.2021 4:49			21			216	inner	4290016109	NPS.MNS1.PT001_1.Value	Давление ДТ на приёме МНС	