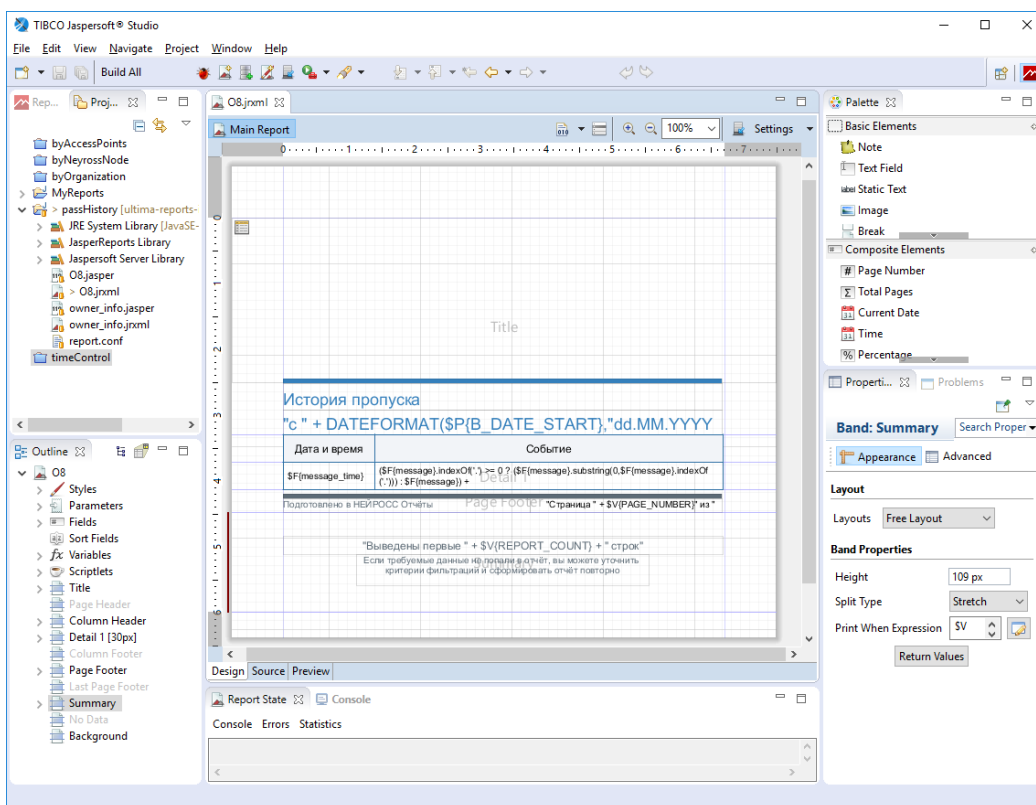


# Краткое руководство по JasperReports

**JasperReports** — это технология и соответствующая Java-библиотека построения отчётов, которые могут быть использованы для создания отчётов в **Платформе НЕЙРОСС**. Средства JasperReports позволяют получать данные из различных источников (в том числе реляционных СУБД), фильтровать их, преобразовывать и формировать на выходе документы в форматах PDF, HTML, Microsoft Excel и многих других. Генерация отчёта осуществляется на основе одного или нескольких XML-шаблонов, в которых можно определить все аспекты формирования документа — от запроса к источнику данных до структуры и внешнего вида конечного документа.

JasperReports относится к классу свободного программного обеспечения и распространяется под свободной лицензией GNU Lesser General Public License, что позволяет свободно использовать JasperReports в коммерческих продуктах.

Редактирование и отладку шаблонов JasperReports можно осуществлять в бесплатной программе **Jaspersoft Studio**. Программа позволяет редактировать, настраивать и тестировать шаблон в визуальном редакторе. На текущий момент для разработки шаблонов для Платформы НЕЙРОСС мы рекомендуем использовать Jaspersoft Studio версии 6.9.0. Все версии Jaspersoft Studio доступны для загрузки [на официальном сайте](#).



Внешний вид визуального редактора Jaspersoft Studio.

## Основные сведения

*Шаблон отчёта* в JasperReports — это файл .jrxml в XML-формате, в котором хранится структура отчёта, его дизайн и логика.

Входные данные отчёта — это параметры и источник данных.

**Параметры** — это входные аргументы. Их значения неизменны во всём отчёте. Например, при формировании отчёта по записям в базе данных за некоторое время, дата / время начала и конца данного временного интервала могут быть переданы в шаблон в форме параметров.

**Источник данных** — это подключение к реляционной базе, XML-файл или другой источник, из которого в соответствии с заданными критериями формируется набор данных. В случае реляционной базы набор данных выбирается из базы данных в соответствии с SQL-запросом, определённым в шаблоне отчёта.

**Набор данных** — это коллекция записей, каждая из которых представляет собой набор полей. Например, в случае реляционной базы записи и поля — это строки и значения колонок таблиц базы данных, выбираемые в соответствии с SQL-запросом.

**Поле** — это именованное значение из набора данных. В случае реляционной базы данных соответствует значению некоторой колонки. Список используемых из набора данных полей также определяется в шаблоне в разделе Fields.

Если нужно вывести в отчёте номер страницы, номер записи, количество записей и т.д., можно использовать переменные.

**Переменная** — это вычисляемое в процессе формирования отчёта значение. Доступно большое количество стандартных переменных, таких как номер страницы, но разработчик шаблона также может создать свои переменные.

**Dataset and Query Dialog**

Language: sql

Query: `1 WITH person AS (SELECT element_id AS id FROM access_get_owners_by_pass(SPI(AUTRUM_PASS)) LIF  
2 SELECT  
3 log.message_time AS message_time,  
4 messages.message_name AS message,  
5 operators.element_name AS operator,  
6 sources.element_name AS source,  
7 2 AS order_id  
8 FROM messages_log AS log  
9 JOIN messages ON log.message_code = messages.message_code  
10 JOIN elements AS sources ON log.audit_element_id = sources.element_id  
11 LEFT OUTER JOIN elements AS operators ON log.operator_id = operators.element_id  
12 WHERE (log.message_time BETWEEN SPI(B_DATE_START) AND SPI(C_DATE_END)) AND target_id = $f  
13 UNION  
14 SELECT  
15 log.time_changed AS message_time,  
16 'Состояние изменено с ' || (state_old.state_name) || ' на ' || (state_new.state_name) AS message,  
17 ' AS operator,  
18 ' AS source,`

**Parameters**

- REPORT\_CONTEXT
- REPORT\_PARAMETERS\_MAP
- JASPER\_REPORTS\_CONTEXT
- JASPER\_REPORT
- REPORT\_CONNECTION
- REPORT\_MAX\_COUNT
- REPORT\_DATA\_SOURCE
- REPORT\_SCRIPTLET
- REPORT\_LOCALE
- REPORT\_RESOURCE\_BUNDLE
- REPORT\_TIME\_ZONE
- REPORT\_FORMAT\_FACTORY
- REPORT\_CLASS\_LOADER
- REPORT\_URL\_HANDLER\_FACTORY
- REPORT\_FILE\_RESOLVER
- REPORT\_TEMPLATES
- SORT\_FIELDS
- FILTER
- REPORT\_VIRTUALIZER
- IS\_IGNORE\_PAGINATION
- DEFINITION\_DIR
- FROM
- TO
- ORGANIZATIONS
- POST
- PASS
- ENTRY
- EXIT

**Fields**

- organization\_id
- surname
- name
- patronymic
- post\_id
- post\_name
- message\_log

**Variables**

- PAGE\_NUMBER
- MASTER\_CURRENT\_PAGE
- MASTER\_TOTAL\_PAGES
- COLUMN\_NUMBER
- REPORT\_COUNT
- PAGE\_COUNT
- COLUMN\_COUNT
- OrganizationGroup\_COUNT
- PostGroup\_COUNT
- total\_count
- organization\_name

На изображениях выше показан редактор SQL-запроса для формирования набора данных и списки параметров, полей и переменных.

Отчёты состоят из *секций*, которые также называются bands (полосы). Секции следуют одна за другой строго по вертикали. Каждая секция принадлежит к определённому типу, который влияет на её отображение в отчёте:

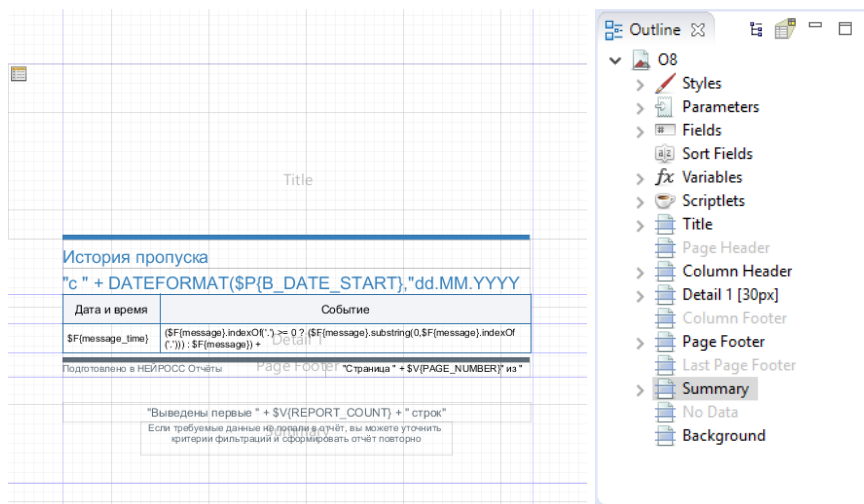
Секция	Описание и назначение
Title	Присутствует только в начале отчёта
Page Header	Выводится вверху каждой страницы
Column Header	Выводится в начале столбца отчёта (секции Detail могут выводиться на странице в несколько столбцов)
Detail	Выводится для каждой записи в источнике данных
Column Footer	Выводится в конце столбца отчёта
Page Footer	Выводится внизу каждой страницы
Last Page Footer	Присутствует только внизу последней страницы (вместо Page Footer)
Summary	Присутствует только в самом конце отчёта
No Data	Выводится вместо всех остальных секций, если источник данных не содержит записей
Background	Выводится на заднем плане каждой страницы

Кроме того, записи в наборе данных делить на группы в соответствии с заданными критериями. Тогда перед и после каждой группы можно выводить дополнительные секции:

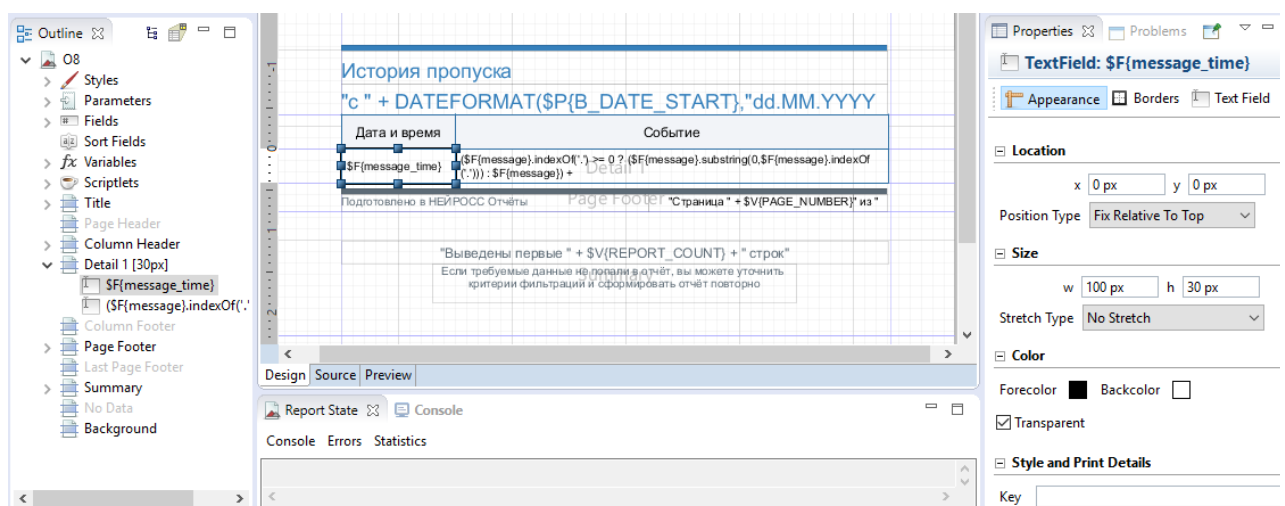
Секция	Описание и назначение
Group Header	Выводится перед каждой новой группой
Group Footer	Выводится после каждой группы

Не все секции необходимо использовать в шаблоне. Ненужные секции можно удалить (отключить) — тогда они не будут участвовать в формировании документа.

Секции подписаны и разделяются серыми горизонтальными линиями на листе отчёта. Список секций также выводится в панели Outline (серым цветом подписаны удалённые секции).



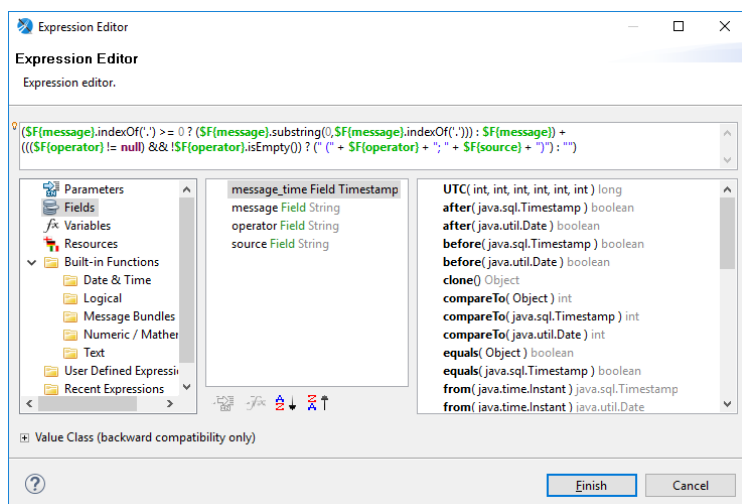
Для вывода текста в отчётах используются два основных компонента: Text Field (динамическое поле) и Static Text (статическое поле). Динамическое поле используется для вывода значений параметров или полей набора данных, а статическое — для вывода постоянного текста (например, заголовков колонок).



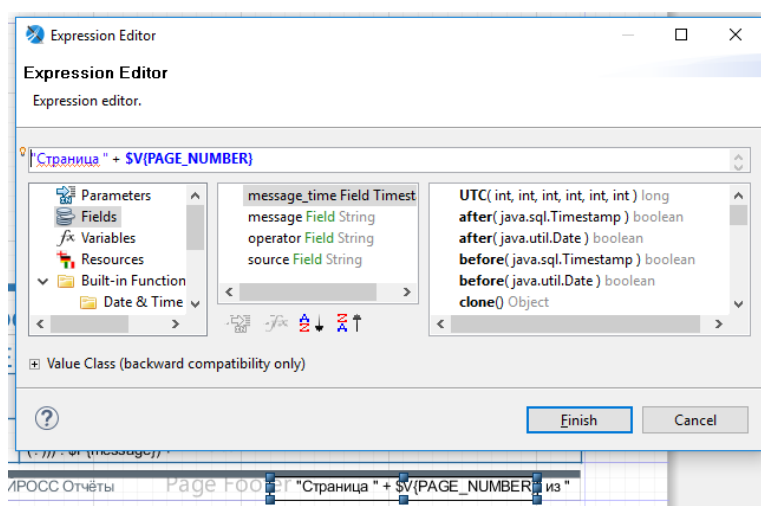
На изображении выше выбран элемент Text Field, в котором в секции Details выводится значение поля `message_time` (время события) для строк из базы данных.

В элементах типа Text Field можно выводить результаты сложных выражений, написанных на языке Java. Так, для вывода текста события в примере выше события используется следующее выражение:

```
($F{message}.indexOf('.') >= 0 ? ($F{message}.substring(0,$F{message}.indexOf('.'))) :
$F{message}) +
((( $F{operator} != null) && !$F{operator}.isEmpty()) ? (" (" + $F{operator} + "; " + $F
{source} + ")") : "")
```



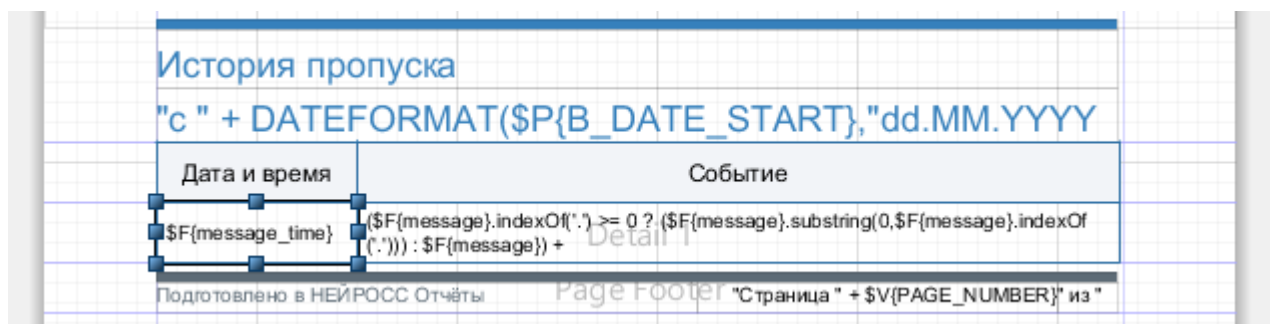
При формировании выражений можно использовать все возможности Java, значения параметров, полей и переменных.



На рисунке выше переменная PAGE\_NUMBER используется для вывода номера текущей страницы.

## Возможности оформления

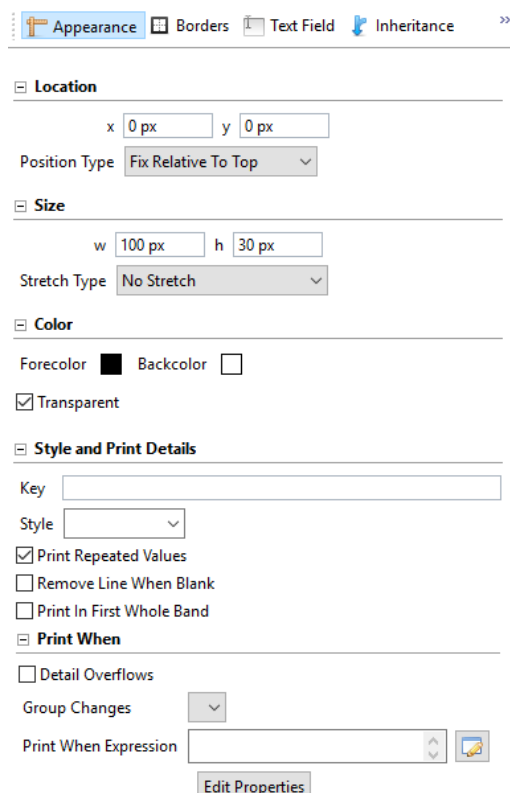
Итак, верстка шаблона отчёта заключается в расположении различных элементов — полей вывода текста, геометрических фигур, изображений и др. — в соответствующих секциях. Помимо непосредственного позиционирования таких элементов на листе и наполнения их данными разработчик шаблона может достаточно гибко настраивать внешний вид этих элементов. Рассмотрим возможности по изменению внешнего вида на примере текстового поля (Text Field).



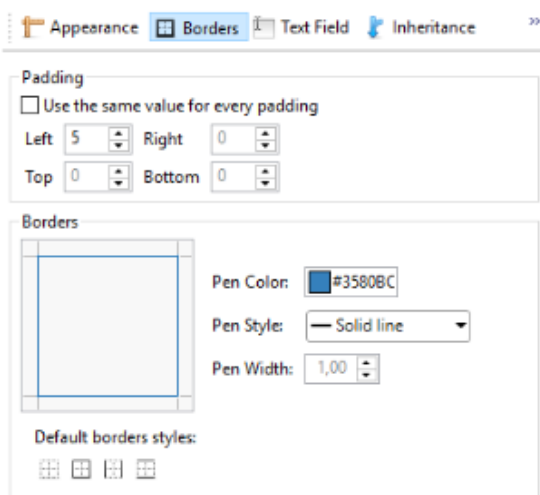
Настройка элементов осуществляется в панели Properties:



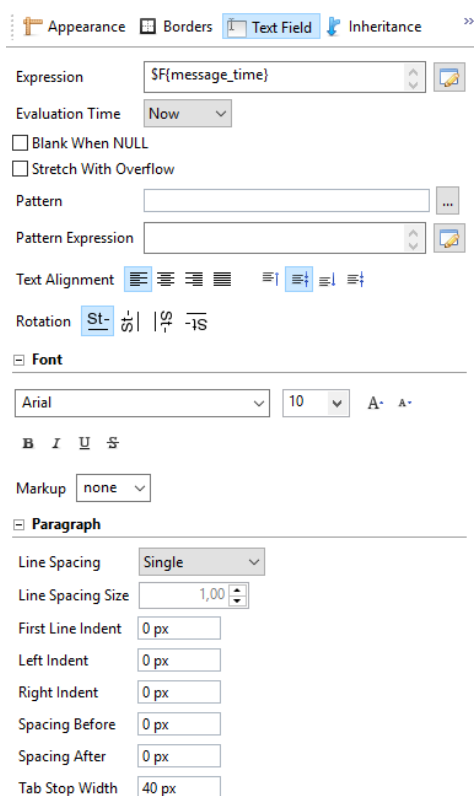
На вкладке Appearance можно настроить точное местоположение и размер элемента, задать основной цвет элемента и цвет фона, указать когда выводить / скрывать элемент и другие параметры.



Для большинства элементов также можно задать стиль оформления границ элемента (какие границы отображать, их цвет, толщина линии) и размер полей (отступов) вокруг элемента.



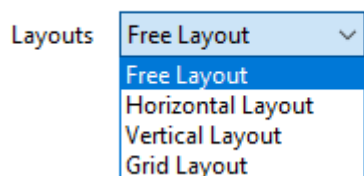
Для текста, выводимого в отчёте с помощью элементов Text Field и Static Text, можно задавать цвет, размер, начертание (жирное, курсив, с подчеркиванием), шрифт, выравнивание текста в блоке по горизонтали и вертикали, а также всевозможные отступы. Более того, текст можно выводить с поворотом на 90, 180 и 270 градусов.



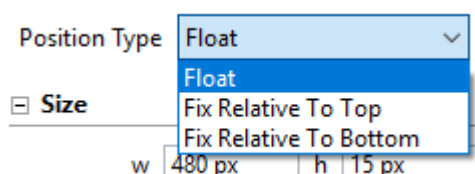
Если размер элемента заранее неизвестен (например, размер текстового поля должен зависеть от объёма текста, полученного из источника данных), то можно настроить относительное позиционирование элементов в секции:

а) либо с помощью выбора правила расположения (layout): элементы располагаются последовательно по вертикали / по горизонтали / в форме таблицы X на Y:

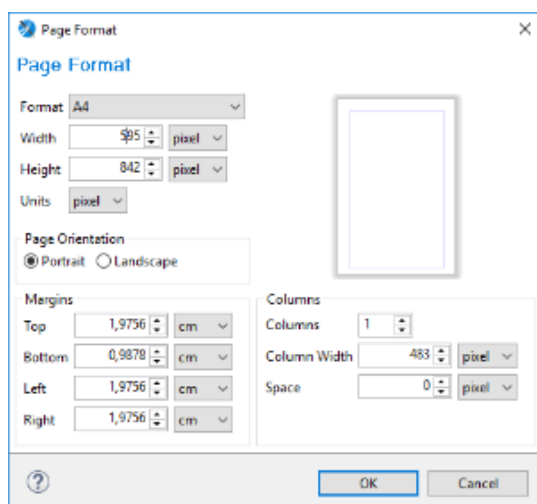
#### Layout



б) либо с помощью привязки элемента к границам секции / границам других элементов:



Наконец, для того, чтобы документ отчёта соответствовал требуемому формату бумаги, например, A4, можно настраивать размер, ориентацию и поля страницы отчёта:

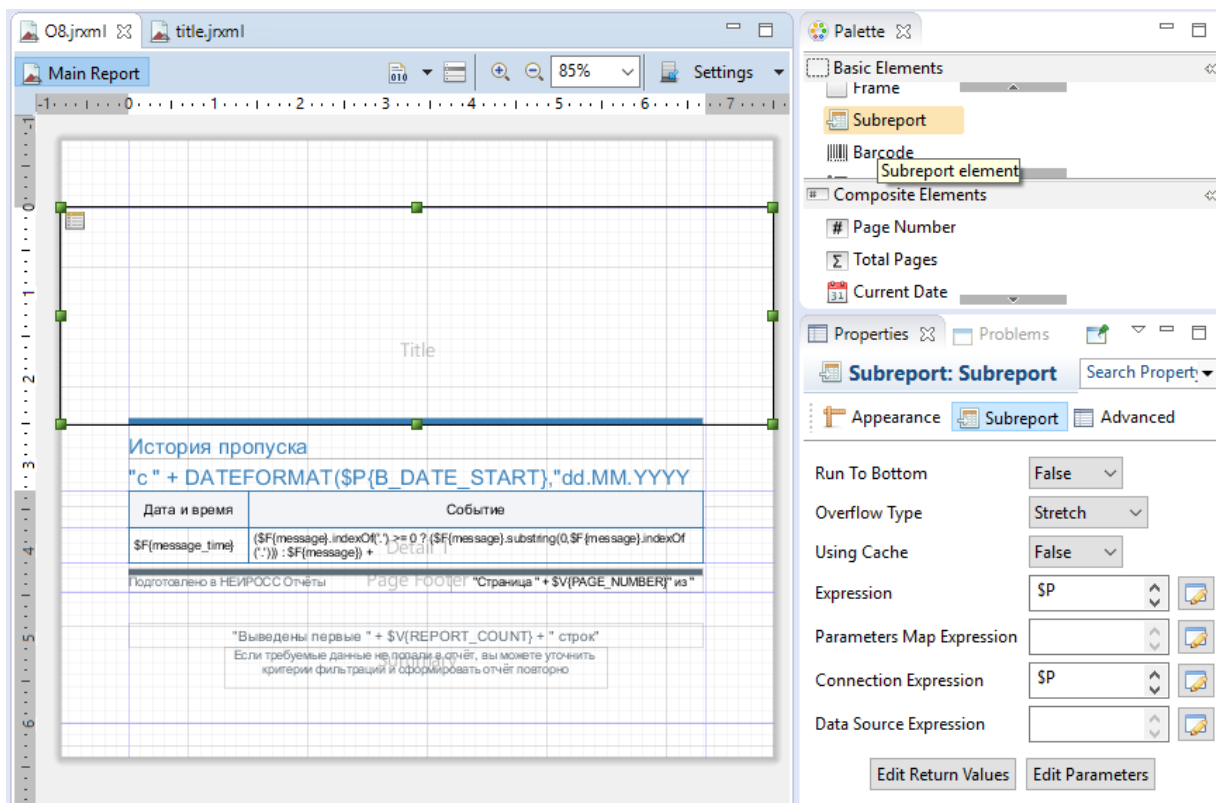


## Подотчёты

Достаточно часто нет возможности получить все необходимые данные из источника с помощью одного запроса. Или бывает так, что в отчёт необходимо включить данные из нескольких наборов. Например, в отчёт типа «Досье по пропуску» в самом начале отчёта требуется вывести подробную информацию о владельце пропуска, а далее в основной части отчёта отобразить таблицу событий доступа, связанных с данным пропуском.

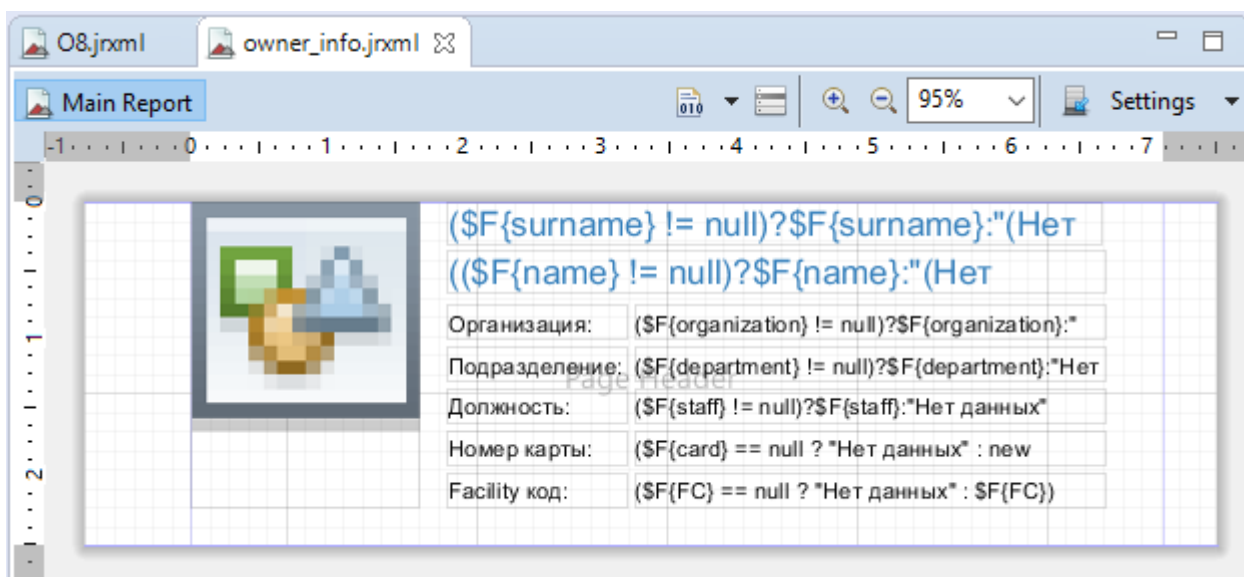
В таких ситуациях на помощь приходит элемент Subreport, который позволяет встраивать один отчёт (подотчёт) в другой.





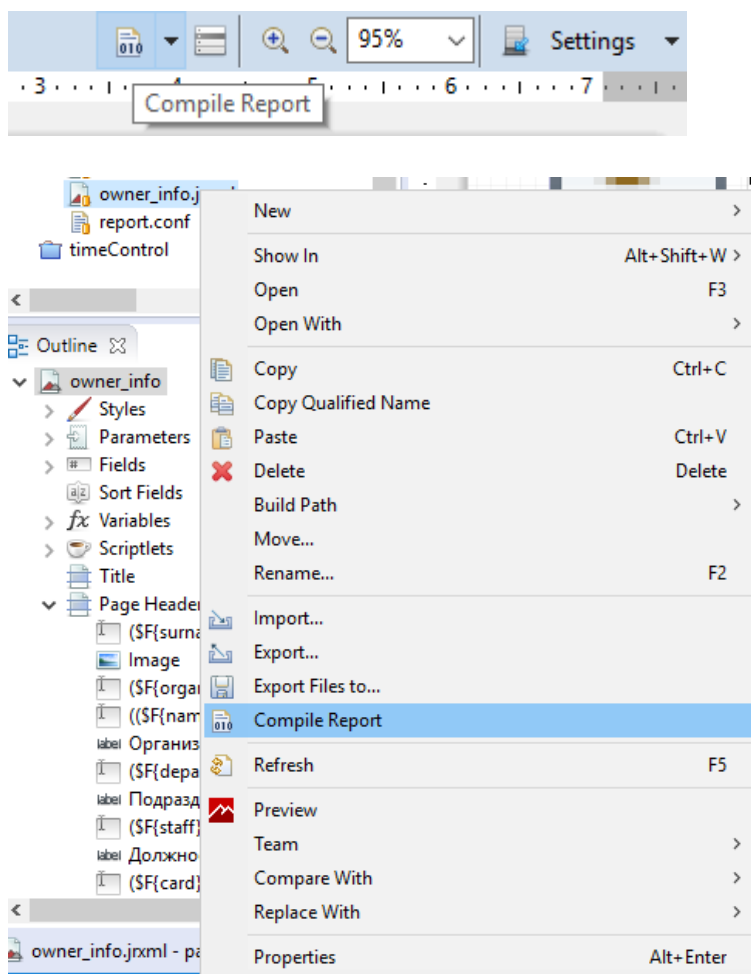
В общем случае подотчёт может использовать любой источник данных, но обычно он просто формирует другой запрос к источнику данных родительского отчёта.

Подотчёт — это ещё один отчёт, который точно также можно редактировать и настраивать:

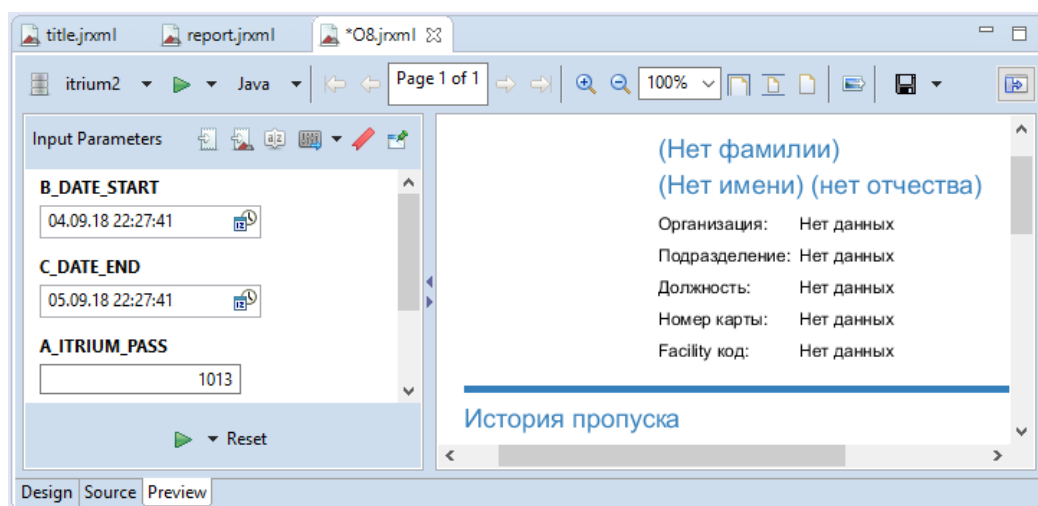


## Компиляция и предпросмотр

JasperReports для генерации отчёта использует не \*.jrxml XML-файлы с определением шаблонов, а скомпилированные из них \*.jasper-файлы. В Jaspersoft Studio для компиляции шаблона необходимо вызвать соответствующую команду по кнопке на панели или в контекстном меню на \*.jrxml-файле:

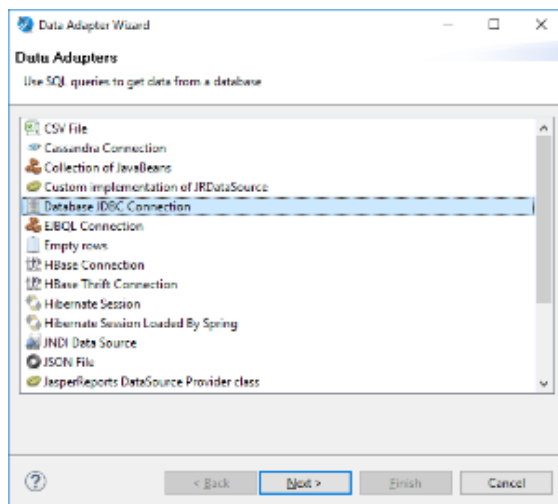


Компиляция шаблона также выполняется автоматически при попытке предпросмотра отчёта:



Механизм предпросмотра позволяет проверить корректность составления шаблона отчёта, запроса к источнику данных и оценить получившийся внешний вид отчёта.

В окне предпросмотра формирование отчёта осуществляется по выбранному источнику данных. Разработчик может добавить нужный ему источник данных (например, подключение к реляционной базе данных) в соответствующем разделе:



## Заключение

На этом краткое руководство кончается.

Дополнительную информацию по использованию JasperReports и Jaspersoft Studio можно почерпнуть в разделе [документации на официальном сайте сообщества](#).