

7 Интерактивные отчеты

7	Интерактивные отчеты	1
7.1	Сворачивание строк в отчете	2
7.2	Детализация данных на отдельной странице	7
7.3	Создание оглавления и гиперссылок	16
7.4	Заключение	25

В большинстве случаев сформированные отчеты занимают небольшой размер – всего несколько страниц. Однако бывают случаи, когда необходимо формировать отчеты с данными на несколько десятков и сотен страниц. Для таких отчетов становится актуальной задача интерактивной навигации. Для этих целей **MasterReport** предлагает несколько средств – сворачивание и разворачивание строк, детализация отчетов и создание оглавлений.

7.1 Сворачивание строк в отчете

Самым простым способом реализовать детализацию данных в отчете – сделать сворачивание строк. Сформированный таким образом отчет, будет содержать небольшое количество строк, содержащих общую информацию о событии. Если же потребуются детальное изучение данного события, то оператор нажимает на иконку разворачивания, что приводит к отображению в отчете дополнительных строк с нужной информацией (Рисунок 7-1).

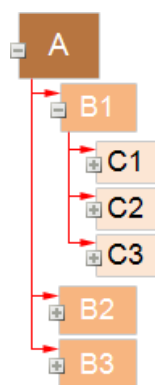


Рисунок 7-1

В качестве примера, сделаем сворачивание в отчете рассмотренном в главе 3 - «**Отчет с группировкой**» (Рисунок 7-2). Напомним, данный отчет представляет информацию о количестве изготовленного сока каждого сорта - в литрах и упаковках.

Сорт	Количество литров	Количество упаковок
Апельсиновый	470	41
Виноградный	230	25
Вишневый	250	25
Грушевый	800	75
Яблочный	170	17
Итого:	1920	183

Рисунок 7-2

В настоящий момент отчет предоставляет только суммарное количество сока каждого сорта, без подробной информации о каждом из цикле разлива. Сделаем, чтобы информация о каждом цикле разлива была свернута, и оператор мог бы ее посмотреть

при необходимости. Мы будем отображать время завершения каждого цикла разлива, количество литров и количество упаковок в этом цикле.

В исходном виде на странице отчета находится 4 бэнда – **Заголовок группы**, **Заголовок данных**, **Данные** и **Итог данных**. При этом размер бэнда Данные установлен у нас равным нулю – то есть данный бэнд не отображается на странице, и необходим лишь для корректного вычисления полей в бэнде Заголовок группы (Рисунок 7-3).

GroupHeaderBand1; Условие: {Данные.Сорт}		
{Данные.Сорт}	(GroupHeaderBand1,Данные.К	(GroupHeaderBand1,Данные.Ко
HeaderBand1		
Сорт	Количество литров	Количество упаковок
DataДанные; Источник данных: Данные		
FooterДанные		
Итого:	(DataДанные,Данные.КолвоЛи	(DataДанные,Данные.КолвоУпа

Рисунок 7-3

Начнем с того, что увеличим размер бэнда Данные и расположим на нем данные – добавим колонки **«Время»**, **«КолвоЛитров»**, **«КолвоУпаковок»**. Для лучшего визуального отделения сделаем размер шрифта текстовых полей этого бэнда чуть меньше, а также установим им серый фон (Рисунок 7-4).

GroupHeaderBand1; Условие: {Данные.Сорт}		
{Данные.Сорт}	(GroupHeaderBand1,Данные.К	(GroupHeaderBand1,Данные.Ко
HeaderBand1		
Сорт	Количество литров	Количество упаковок
DataДанные; Источник данных: Данные		
{Данные.Время}	{Данные.КолвоЛитров}	{Данные.КолвоУпаковок}
FooterДанные		
Итого:	(DataДанные,Данные.КолвоЛи	(DataДанные,Данные.КолвоУпа

Рисунок 7-4

Запустим предварительный просмотр (Рисунок 7-5).

Сорт	Количество литров	Количество упаковок
Апельсиновый	940	82
29.06.2014 16:11:40	320	31
29.06.2014 16:10:36	150	10
29.06.2014 16:11:44	320	31
29.06.2014 16:10:33	150	10
Виноградный	230	25
29.06.2014 16:12:40	230	25
Вишневый	500	50
29.06.2014 16:11:59	100	10

Рисунок 7-5

Теперь сделаем чтобы серые строки, с детальной информацией о каждом цикле были скрыты, и разворачивались по команде оператора.

Мы будем производить сворачивание групп. Для корректной работы сворачивания/разворачивания группы, необходимо добавить бэнд Итог группы. Добавим данный бэнд, и сделаем его размер нулевым, поскольку никакой информации выводить в него не нужно (Рисунок 7-6).

GroupHeaderBand1; Условие: {Данные.Сорт}		
{Данные.Сорт}	(GroupHeaderBand1.Данные.К	(GroupHeaderBand1.Данные.Ко
HeaderBand1		
Сорт	Количество литров	Количество упаковок
DataДанные; Источник данных: Данные		
{Данные.Время}	{Данные.КолвоЛитров}	{Данные.КолвоУпаковок}
FooterДанные		
Итого:	(DataДанные.Данные.КолвоЛи	(DataДанные.Данные.КолвоУпа
GroupFooterBand1		

Рисунок 7-6

Теперь приступим к настройке сворачивания. Выделим бэнд Заголовок группы, и развернем группу настроек **Взаимодействие** (Рисунок 7-7).

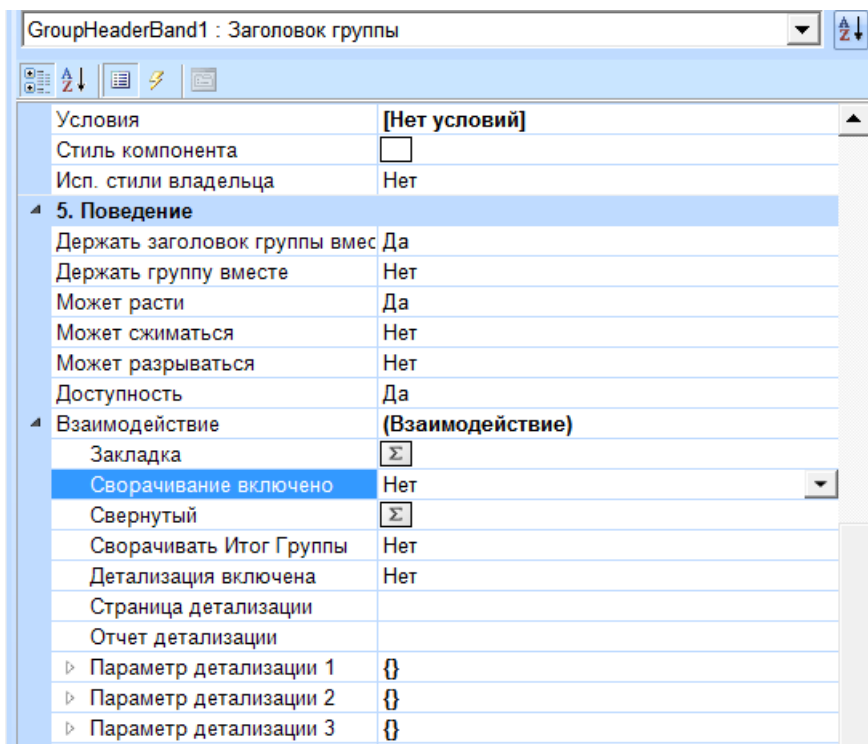


Рисунок 7-7

Включим настройку **Сворачивание включено** - это активизирует сворачивание содержимого по нажатию на компонент.

Запустим предварительный просмотр. Теперь в строку с сортом сока добавился управляющий элемент – кнопка для сворачивания/разворачивания (Рисунок 7-8).

Сорт	Количествов литров	Количество упаковок
☒ Апельсиновый	940	82
☒ Виноградный	230	25
29.06.2014 16:12:40	230	25
☒ Вишневый	500	50
29.06.2014 16:11:59	100	10
29.06.2014 16:12:15	150	15

Рисунок 7-8

По умолчанию все элементы развернуты и детальные данные видны.

Редактор отчетов позволяет задать условие, при котором элементы будут свернуты – для этого предназначена настройка **Свернутый** (Рисунок 7-9).

держат группу вместе	Нет
Может расти	Да
Может сжиматься	Нет
Может разрываться	Нет
Доступность	Да
Взаимодействие	(Взаимодействие)
Закладка	Σ
Сворачивание включено	Да
Свернутый	Σ
Сворачивать Итог Группы	Нет
Детализация включена	Нет
Страница детализации	
Отчет детализации	

Рисунок 7-9

Например, можно указать чтобы развернутой была только первая группа, прописав условие:

{GroupLine!=1}

Можно сформировать условие, связанное с выводимыми данными самого отчета – например, сделать развернутой ту группу, в которой было превышение максимально допустимого значения, поступление некорректных данных и т.п.

Если же необходимо чтобы вначале были свернуты все элементы, то нужно просто прописать:

{true}

Пропишем данное условие (Рисунок 7-10).

Может разрываться	Нет
Доступность	Да
Взаимодействие	(Взаимодействие)
Закладка	Σ
Сворачивание включено	Да
Свернутый	Σ {true}
Сворачивать Итог Группы	Нет
Детализация включена	Нет
Страница детализации	
Отчет детализации	

Рисунок 7-10

Запустим предварительный просмотр (Рисунок 7-11).

Сорт	Количество литров	Количество упаковок
Апельсиновый	940	82
Виноградный	230	25
Вишневый	500	50
Грушевый	1600	150
Яблочный	340	34
Итого:	3610	341

Рисунок 7-11

Теперь все группы свернуты, но оператор может развернуть любую из групп и увидеть детальные данные (Рисунок 7-12).

Сорт	Количество литров	Количество упаковок
Апельсиновый	940	82
Виноградный	230	25
Вишневый	500	50
29.06.2014 16:11:59	100	10
29.06.2014 16:12:15	150	15
29.06.2014 16:12:19	150	15
29.06.2014 16:12:03	100	10
Грушевый	1600	150
Яблочный	340	34
Итого:	3610	341

Рисунок 7-12

По аналогичном принципу можно сворачивать не только отчеты с группами, но и Master-Detail отчеты и отчеты других типов.

Примечание. Возможность сворачивания/разворачивания доступно только в окне просмотра отчетов. В сохраненном отчете (независимо от формата сохранения), данные кнопки отсутствуют.



Примечание. Готовый отчет можно посмотреть в проекте «Интерактивные отчеты» в объекте «Сворачивание строк».

7.2 Детализация данных на отдельной странице

Более продвинутым способом детализации данных является детализация на отдельной странице отчета. Суть данного метода, состоит в том, что в отчете создается еще одна

дополнительная страница, в которой и размещаются детализированные данные. Если, после формирования отчета, оператору необходимо увидеть дополнительную информацию он щелкает по определенному компоненту (строка данных, ячейка, график), и происходит открытие дополнительной страницы.

В качестве примера сделаем детализацию в **Master-Detail отчете**, рассмотренном в главе 5 «Непрерывно-периодический отчет». Напомним, данный отчет отображает информацию о проведенном циклическом процессе (отжиг, закалка). Отчет содержит в себе общую информацию о процессе – время начала и конца цикла, уставка, максимальное и минимальное значение, а также график процесса (Рисунок 7-13).

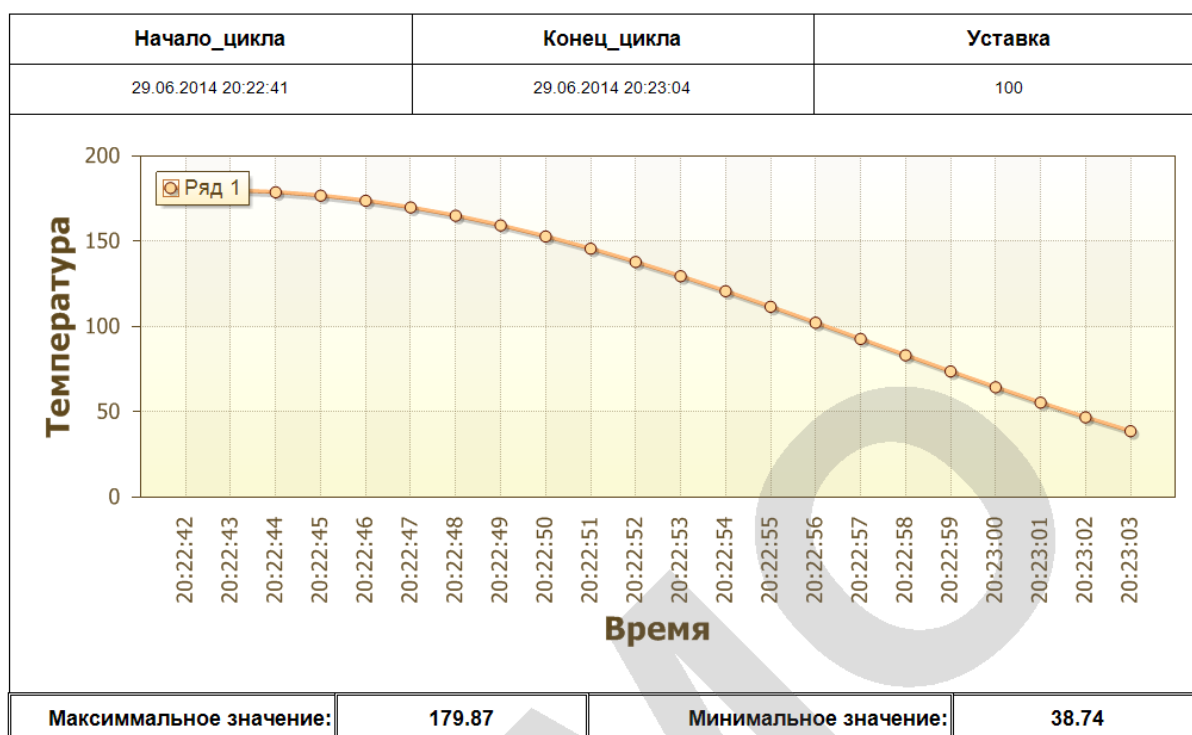


Рисунок 7-13

Сделаем, чтобы при щелчке по графику процесса открывалась дополнительная страница отчета, содержащую процесс изменения температуры в табличном виде.

Сначала создадим новую страницу (Рисунок 7-14).

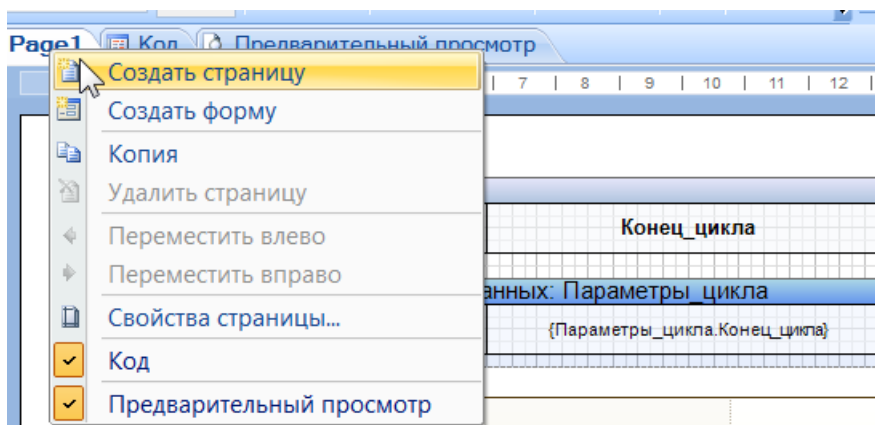


Рисунок 7-14

Вытащим на созданную страницу источник данных «Значения_температуры», разместим на странице три бэнда (Заголовок данных, Данные и Итог данных) и настроим внешний вид ячеек - сделаем границу и разместим текст по центру (Рисунок 7-15).

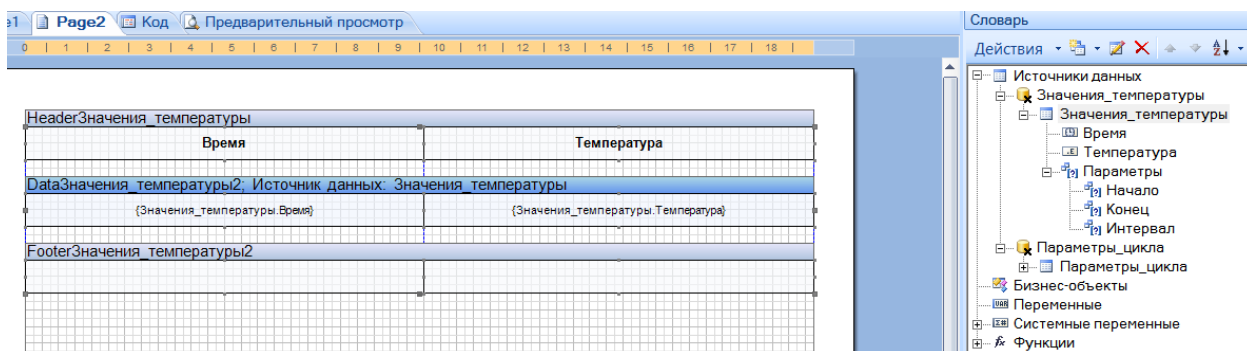


Рисунок 7-15

Теперь сделаем, чтобы щелчок по диаграмме на первой странице, приводил к открытию второй страницы. Для этого выделим диаграмму, и перейдем на панель свойств. В панели свойств, включим настройку Детализация включена в группе настроек Взаимодействие (Рисунок 7-16).

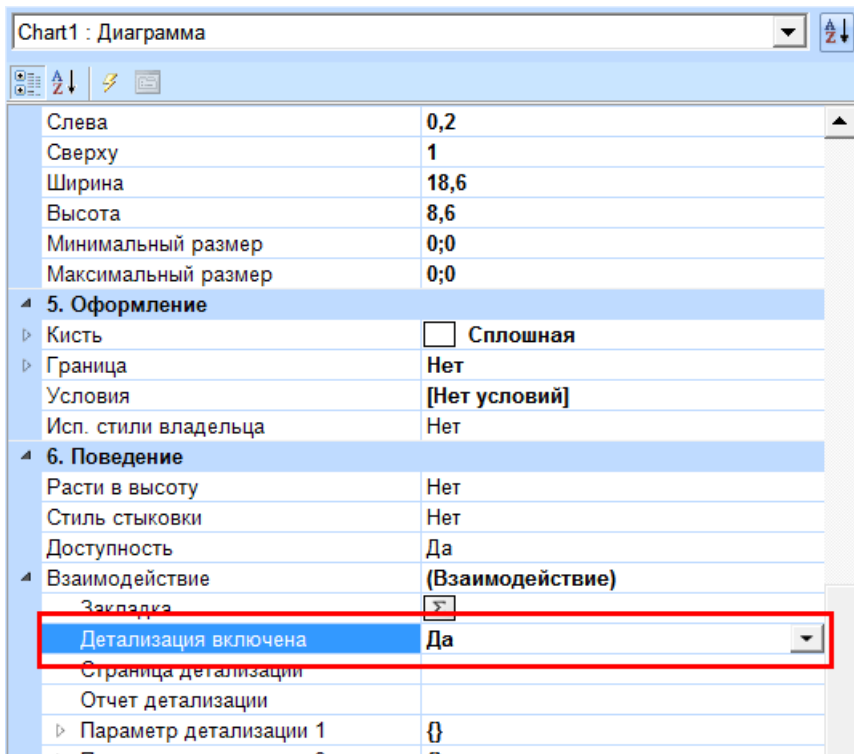


Рисунок 7-16

Теперь необходимо указать страницу детализации – настройкой **Страница детализации** (Рисунок 7-17).

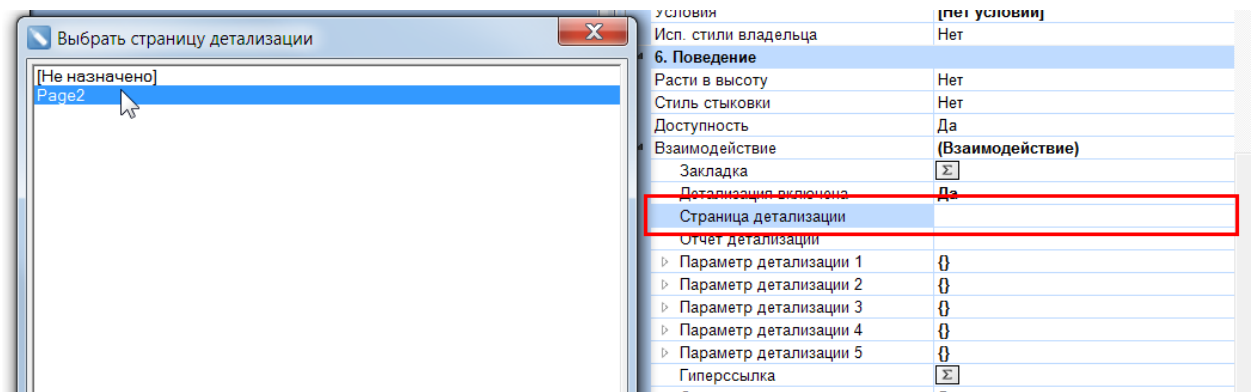


Рисунок 7-17

Выберем страницей детализации нашу вторую страницу – **Page2**.

Запустим предварительный просмотр (Рисунок 7-18). Наведем мышью на сформированный график (курсор мыши изменит свою форму), и нажмем левую кнопку мыши.

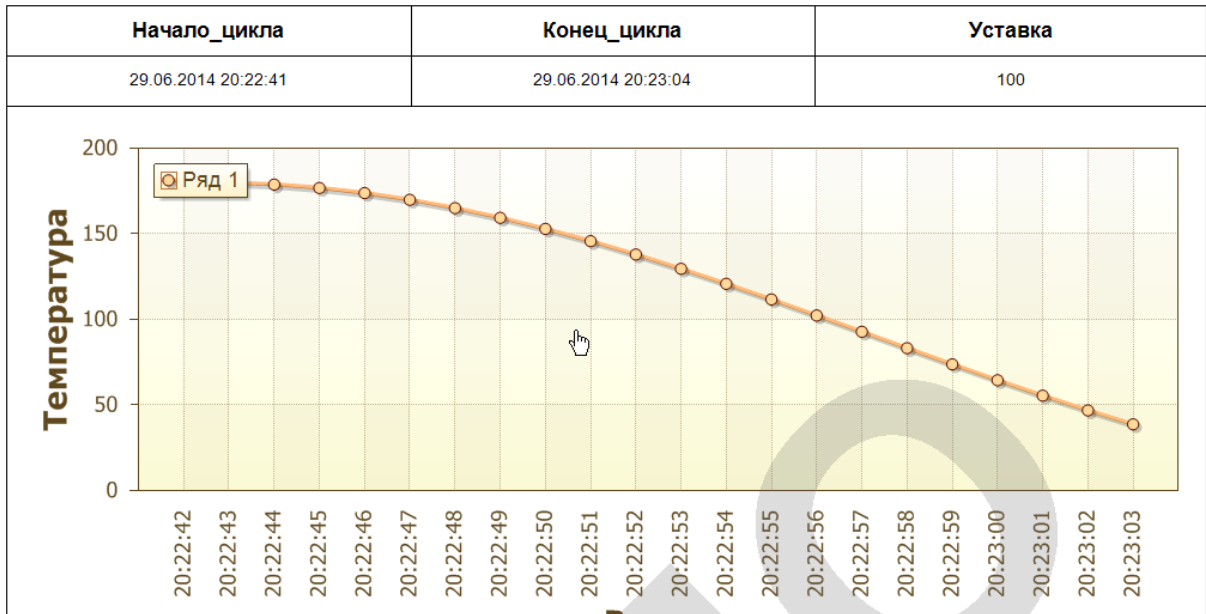


Рисунок 7-18

Произошло открытие нашей второй страницы. При этом в верхней части окна просмотра появились вкладки (Рисунок 7-19) – с их помощью можно осуществлять навигацию по открытым окнам (вернутся на исходную, или другую сформированную страницу).

Время	Температура
29.06.2014 20:22:32	129,611451211255
29.06.2014 20:22:33	137,953547210996
29.06.2014 20:22:34	145,755454181767
29.06.2014 20:22:35	152,929284891117
29.06.2014 20:22:36	159,394227285083
29.06.2014 20:22:37	165,07745482304

Рисунок 7-19

Текущий отчет содержит информацию о трех циклах, поэтому и на странице детализации отображены данные по всем этим циклам. Нам же необходимо, чтобы страница содержала данные только по выбранному нами циклу.

Чтобы убрать лишние данные, их необходимо отфильтровать по времени начала и времени конца цикла. В настройках бэнда Данные есть вкладка Фильтры, где можно произвести данную операцию. Однако для того чтобы выполнить фильтрацию, необходимо передать на страницу детализации время начала и конец цикла. Для решения данной задачи, у компонента существуют параметры детализации.

Вернемся в режим разработки, и снова выберем диаграмму. В группе **Взаимодействие** есть 5 настроек **Параметр детализации** (Рисунок 7-20).

6. Поведение	
Растить в высоту	Нет
Стиль стыковки	Нет
Доступность	Да
Взаимодействие	(Взаимодействие)
Закладка	Σ
Детализация включена	Да
Страница детализации	Page2
Отчет детализации	
Параметр детализации 1	{}
Наименование	
Выражение	Σ
Параметр детализации 2	{}
Параметр детализации 3	{}
Параметр детализации 4	{}
Параметр детализации 5	{}
Гиперссылка	Σ
Сортировка включена	Да

Рисунок 7-20

Наименование параметра – это имя, по которому можно будет обратиться на странице детализации для получения переданного значения. В поле **Выражение** задается функция которая присвоит параметру детализации нужное значение.

Нам необходимо передать на страницу детализации время начала (колонка источника данных «**Параметры_цикла.Начало_цикла**») и конца цикла (колонка «**Параметры_цикла.Конец_цикла**»).

Зададим **параметру детализации 1**, наименование – **Start**, а выражение –

Параметры_цикла.Начало_цикла

Взаимодействие	
(Взаимодействие)	
Закладка	Σ
Детализация включена	Да
Страница детализации	Page2
Отчет детализации	
Параметр детализации 1	Start,Параметры_цикла.Начало_цикла
Наименование	Start
Выражение	Σ Параметры_цикла.Начало_цикла ...
Параметр детализации 2	{}
Параметр детализации 3	{}
Параметр детализации 4	{}
Параметр детализации 5	{}

Рисунок 7-21

Второму параметру детализации зададим наименование – **End**, а выражение – **Параметры_цикла.Конец_цикла**.

Взаимодействие	(Взаимодействие)
Закладка	Σ
Детализация включена	Да
Страница детализации	Page2
Отчет детализации	
Параметр детализации 1	Start,Параметры_цикла.Начало_цикла
Наименование	Start
Выражение	Σ Параметры_цикла.Начало_цикла
Параметр детализации 2	End,Параметры_цикла.Конец_цикла
Наименование	End
Выражение	Σ Параметры_цикла.Конец_цикла
Параметр детализации 3	∅

Рисунок 7-22

Перейдем на **страницу 2**. Сначала проверим передачу параметров на страницу детализации – сделаем вывод значений параметров в заголовке отчета. Для этого добавим на страницу **бэнд Заголовок отчета**. Добавим на **бэнд** два **текстовых поля** с текстом «**Время начала цикла**» и «**Время конца цикла**».

ReportTitleBand1	
Время начала цикла -	
Время начала цикла -	
HeaderЗначения_температуры	
Время	Температура
DataЗначения_температуры2; Источник данных: Значения_температуры	

Рисунок 7-23

Добавим напротив верхнего текстового поля еще одно поле. Пропишем в поле код:

```
{{(DateTime) this["Start"]}}
```

С помощью **this["Start"]** происходит обращение к переданному параметру детализации **"Start"**. Функция **(DateTime)** выполняет преобразование полученного значения, к необходимому нам типу «**Дата-Время**».

Аналогично сделаем со вторым параметром детализации. Теперь код будет:

```
{{(DateTime)this["End"]}}
```

В итоге **бэнд** будет выглядеть следующим образом:

ReportTitleBand1	
Время начала цикла -	{{(DateTime)this["Start"]}}
Время начала цикла -	{{(DateTime)this["End"]}}

Рисунок 7-24

Запустим предварительный просмотр, и выберем одну из диаграмм. В ячейки будут прописаны время начала и конца выбранного цикла:

Время начала цикла -	29.06.2014 20:22:41
Время начала цикла -	29.06.2014 20:23:04
Время	Температура
29.06.2014 20:22:32	129,611451211255

Рисунок 7-25

Теперь настроим фильтрацию.

Выделим бэнд Данные, откроем окно настроек данных и перейдем на вкладку Фильтры.

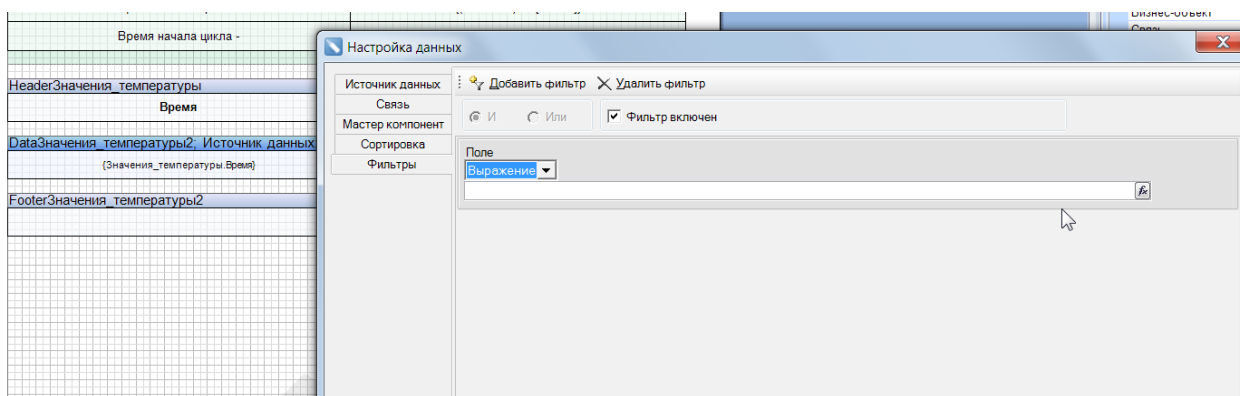


Рисунок 7-26

Введем выражение нашего фильтра (Рисунок 7-27):

**Значения_температуры.Время>(DateTime)this["Start"] &&
Значения_температуры.Время<(DateTime)this["End"]**

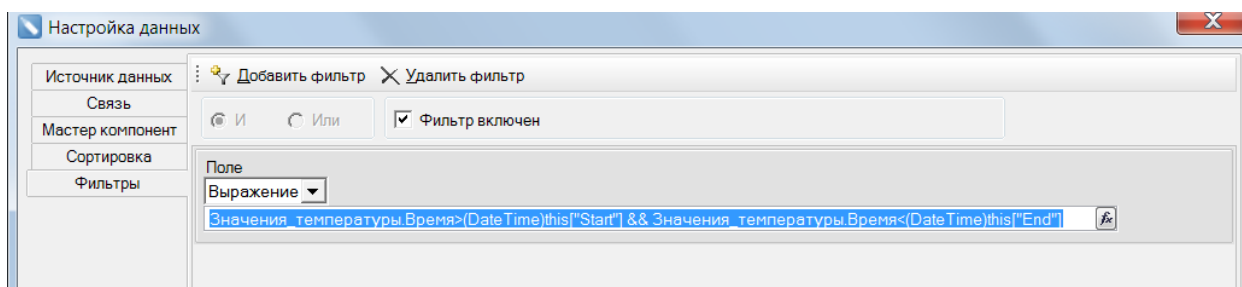


Рисунок 7-27

С помощью такой фильтрации мы оставим на странице данные только за нужный нам цикл.

Запустим предварительный просмотр, и выберем один из циклов (Рисунок 7-28).

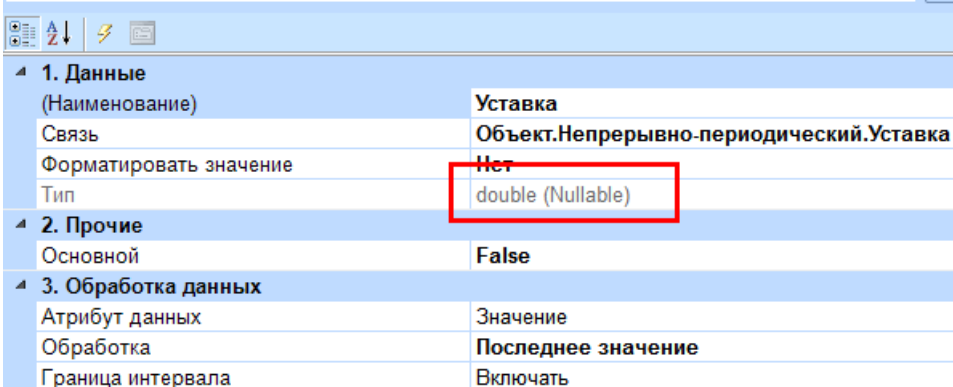
Время начала цикла -	29.06.2014 20:22:41
Время начала цикла -	29.06.2014 20:23:04

Время	Температура
29.06.2014 20:22:42	179,872680426903
29.06.2014 20:22:43	179,873680990596
29.06.2014 20:22:44	178,862268932328
29.06.2014 20:22:45	176,849837646946
29.06.2014 20:22:46	173,859056850562

Рисунок 7-28

Теперь на странице только нужные нам данные.

Аналогичным образом можно осуществлять фильтрацию по любому другому передаваемому параметру (например - по имени установки, по уставке значения, номеру счетчика и т.д.). Во всех случаях обращение к параметру на странице детализации будет осуществляться через ***this["Имя_Параметра"]***. В зависимости от типа данных параметра, функции преобразования будет отличаться: для строк – (***string***), для целых чисел – (***int***), для вещественных - (***double***) и т.д. Узнать тип данных колонки, можно на закладке Свойства (Рисунок 7-29).



1. Данные	
(Наименование)	Уставка
Связь	Объект.Непрерывно-периодический.Уставка
Форматировать значение	Нет
Тип	double (Nullable)
2. Прочие	
Основной	False
3. Обработка данных	
Атрибут данных	Значение
Обработка	Последнее значение
Граница интервала	Включать

Рисунок 7-29

Примечание. Возможность детализации доступно только в окне просмотра отчетов.



Примечание. Готовый отчет можно посмотреть в проекте «Интерактивные отчеты» в объекте «Детализация данных».

7.3 Создание оглавления и гиперссылок

Предыдущие способы детализации отчетов работают только в окне просмотра отчетов. Если же отчет необходимо распечатать или сохранить в сторонний формат, то можно создать отчет с оглавлением и гиперссылками.

В качестве примера рассмотрим создание месячного отчета расхода газа с почасовыми записями (*Рисунок 7-30*). В данном отчете (*Рисунок 7-31*) содержатся строки с данными о расходе газа в течении всего месяца, на каждую страницу выводится **заголовок данных**, а в конце отчета – **общий итог данных** (суммарный расход за месяц).

PageHeaderBand1	
Страница {PageNum}	
HeaderПоказанияСчетчика	
Время	Расход
DataПоказанияСчетчика; Источник данных: ПоказанияСчетчика	
{ПоказанияСчетчика.Время}	{ПоказанияСчетчика.Расход}
FooterПоказанияСчетчика	
Расход за сутки	(DataПоказанияСчетчика.ПоказанияСчетчика.Расход

Рисунок 7-30

Страница 3

Время	Расход
02.11.2013 9:00:00	145
02.11.2013 10:00:00	164
02.11.2013 11:00:00	179
02.11.2013 12:00:00	153
02.11.2013 13:00:00	152
02.11.2013 14:00:00	193
02.11.2013 15:00:00	143
02.11.2013 16:00:00	122
02.11.2013 17:00:00	211
02.11.2013 18:00:00	146
02.11.2013 19:00:00	180
02.11.2013 20:00:00	145
02.11.2013 21:00:00	134
02.11.2013 22:00:00	164
02.11.2013 23:00:00	157
03.11.2013 0:00:00	166
03.11.2013 1:00:00	195
03.11.2013 2:00:00	158

Рисунок 7-31

Сначала сделаем для данного отчета **оглавление** – сделаем чтобы оглавление указывало на начало каждого дня месяца в отчете.

Как правило оглавление должно располагаться в начале отчета. Однако в начале построения номера нужных страниц нам еще неизвестны. Неизвестно нам и в каком месте страницы начнутся показания новых суток. Для решения данных проблем в редакторе отчетов предусмотрены специальные средства.

Для создания оглавления, необходимо в нужном месте отчета создать специальный элемент - «якорь» (Anchor). Якорь – это специальная метка, на которую затем можно получить ссылку. Создание якоря происходит программно в коде события – при помощи метода:

AddAnchor("Имя_Якоря").

После того, как якорь создан к нему можно обратиться. Например, при помощи метода

GetAnchorPageNumber("Имя_Якоря")

можно получить номер страницы, на которой расположен якорь с данным именем. Если якорь с указанным именем не будет найден, то метод вернет **0**. С помощью этих двух методов и организуется построение оглавление. Поскольку оглавление должно выводиться в начале отчета, но только после того как сформирован весь отчет (и соответственно добавлены все якоря), то у текстового поля в которое выводится номер страницы якоря необходимо включить свойство **Обработать в - В конце отчета**.

Рассмотрим пошагово создание отчета с оглавлением.

Наш отчет содержит один источник данных – **ПоказанияСчетчика** (Рисунок 7-32).

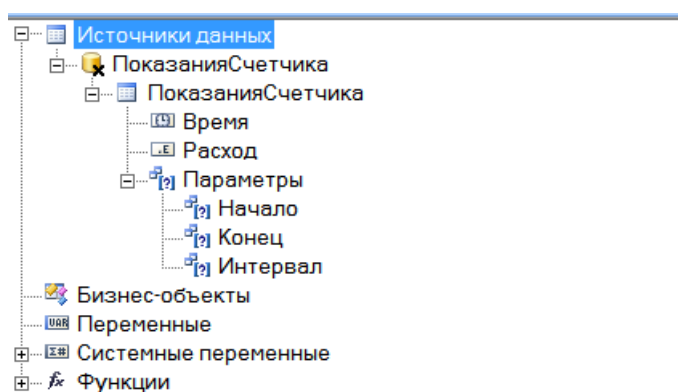


Рисунок 7-32

Этот источник данных хранит действующие значения расхода газа, в колонке **«Расход»**. Способ формирования отчета – **периодический**, с интервалом в 1 час. У колонки включен способ обработки **Интеграл** - таким образом производится вычисление

расхода газа за час на основе действующих значений.

1. Данные		1. Данные	
(Наименование)	ПоказанияСчетчика	(Наименование)	Расход
(Псевдоним)	ПоказанияСчетчика	Связь	Объект.Объект 1.Генерация архива.Выходы.
Колонки	(Коллекция)	Форматировать значение	Нет
2. Прочие		Тип	double (Nullable)
Способ формирования зна	Периодический	2. Обработка данных	
Единица измерения интерв:	Час	Атрибут данных	Значение
Предпочитаемый слой	Основной	Обработка	Интеграл
Использовать другие сло	Да	Параметр обработки	3600
Преобразовывать Nulls	Наследовать из отчета	Метод интегрирования	Метод левых прямоугольников
Конвертировать в Null нед	Нет	3. Обработка плохих признаков	
		Обработка признака Stop	Исключать недостоверные области; Uncertai

Рисунок 7-33

Через контекстное меню создадим новую страницу в отчете, и поместим ее слева.

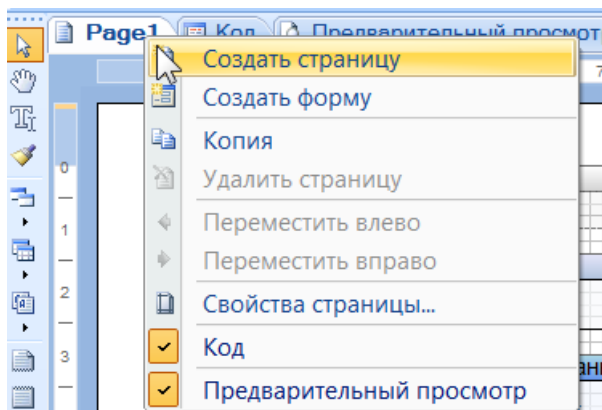


Рисунок 7-34

Дадим страницам более информативные имена, страницу с оглавлением назовем «**Contents**», а страницу с отчетом – «**ReportData**» (Рисунок 7-35). Имена страниц задаются через панель свойств.

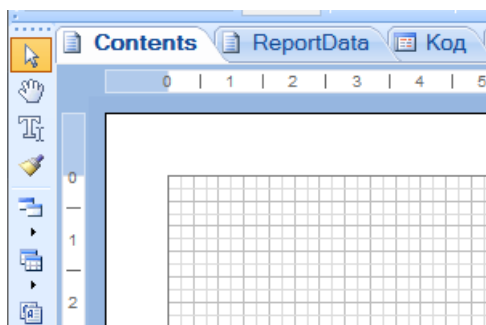


Рисунок 7-35

Наш отчет выводит строки на страницу с шагом в 1 час. При построении основного отчета, мы будем создавать якоря с именами по числу месяца («1», «2» и т.д). Таким образом на странице оглавления нам нужно создать количество строк по количеству дней в месяце. Кроме того, в каждую строку, кроме номера страницы, нужно выводить и информацию о заголовке. В качестве такой информации мы будем выводить полную дату (например – «5 ноября 2013»).

Самый простой способ получить источник данных с количеством строк соответствующих количеству дней в месяце – это создать новый источник данных, включить у него периодический способ формирования и задать интервал – 1 сутки.

Создадим новый источник данных и назовем его **«Вспомогательный»**. Свяжем параметры «Начало» и «Конец» с переменными из дерева объекта (аналогичными как для источника **«ПоказанияСчетчика»**).

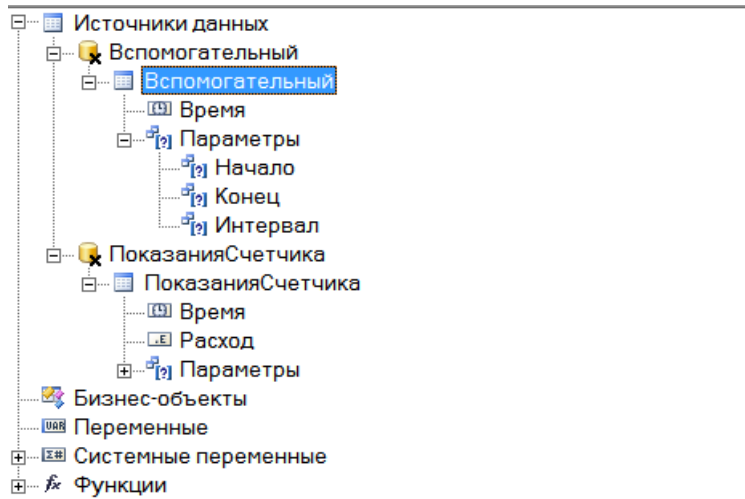


Рисунок 7-36

Зададим периодический способ формирования, с интервалом в сутки.

1. Данные	
(Наименование)	Вспомогательный
(Псевдоним)	Вспомогательный
Колонки	(Коллекция)
2. Прочие	
Способ формирования знач	Периодический
Единица измерения интерва	Сутки
Предпочитаемый слой	Основной
Использовать другие слои	Да
Преобразовывать Nulls	Наследовать из отчета
Конвертировать в Null недо	Нет

Рисунок 7-37

Добавим в источник данных рассчитываемую колонку, которую назовем **«ЗаголовокДата»** (Рисунок 7-38).

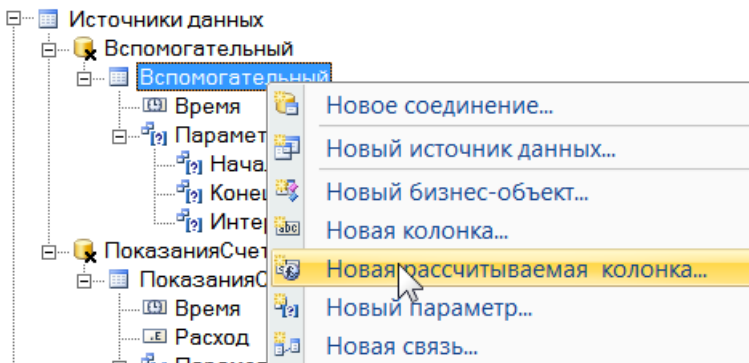


Рисунок 7-38

Тип установим **DateTime**, в поле Значение введем формулу (Рисунок 7-39). Данная колонка будет возвращать прошедшую дату – то есть из колонки «**Время**» мы будем вычитать сутки. Для этого нужно использовать метод **AddDays(-1)**. Текст выражения в итоге будет иметь вид:

Вспомогательный.Время.AddDays(-1)

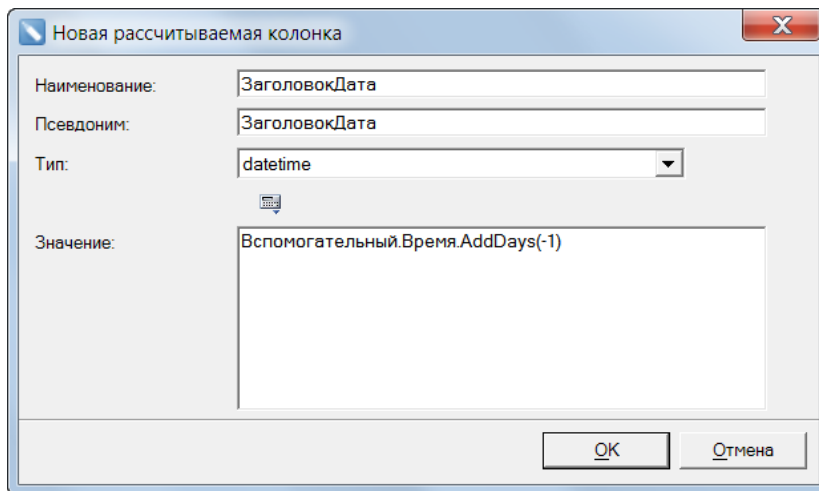


Рисунок 7-39

Вытащим источник данных на страницу. Кроме бэнда данных, добавим на страницу и заголовок данных.

HeaderВспомогательный	
Заголовок	Страница
DataВспомогательный; Источник данных: Вспомогательный	
{Вспомогательный.ЗаголовокДата}	

Рисунок 7-40

Создадим якоря. Перейдем на страницу отчета «**ReportData**», выделим **бэнд данных** и перейдем на вкладку событий (Рисунок 7-41).

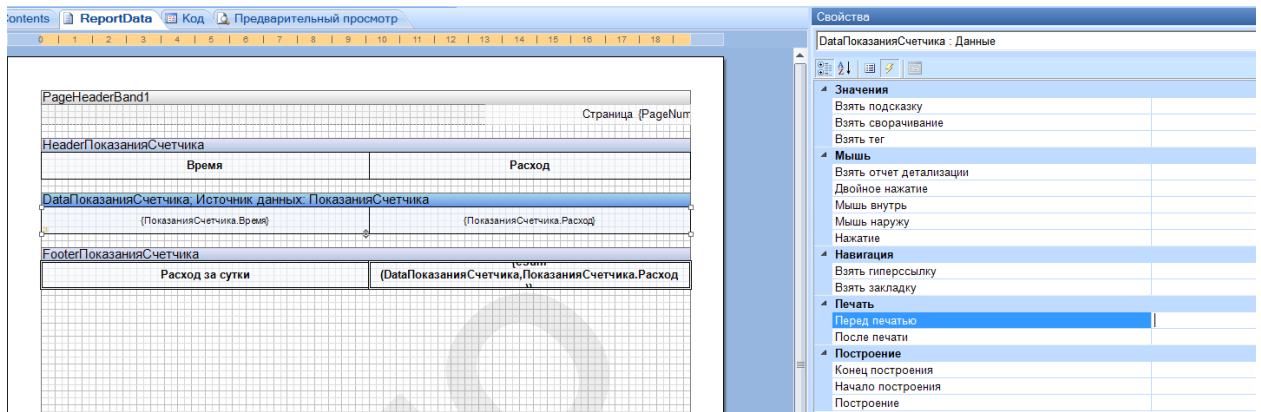


Рисунок 7-41

Мы будем создавать якорь в событии **Перед печатью**. Условием создания якоря у нас будет создание строки данных за новое число. Можно определить создание строки с новым числом отслеживая значение часа (если 0 часов – наступил новый день). Но в нашем случае можно создавать якорь через каждые 24 строки данных. Таким образом код будет выглядеть следующим образом:

```
if ((Line-1)%24 == 0)
```

```
AddAnchor((Day(ПоказанияСчетчика.Время)).ToString());
```

Разберем строку где происходит добавление якоря.

В помощью функции **Day()**, мы получаем день месяца из колонки времени. После этого при помощи метода **ToString()**, мы производим преобразование числа в строку (имя якоря обязательно должно иметь строковый тип данных). Именно эта строка и передается в метод **AddAnchor**, и якорь с именем соответствующий дню месяца создается.

Теперь разместим на странице оглавления номер страницы на котором находится каждый якорь. Вызовем редактор текста ячейки для номера страницы и пропишем код (Рисунок 7-42):

```
{GetAnchorPageNumber((Day(Вспомогательный.ЗаголовокДата)).ToString())}
```

Часть кода

```
((Day(Вспомогательный.ЗаголовокДата)).ToString())
```

действует аналогично, как и в случае с добавлением якоря – из даты извлекается день, а затем преобразуется в строку. Полученная строка передается в метод **GetAnchorPageNumber**, который и возвращает номер страницы, на которой находится указанный якорь.

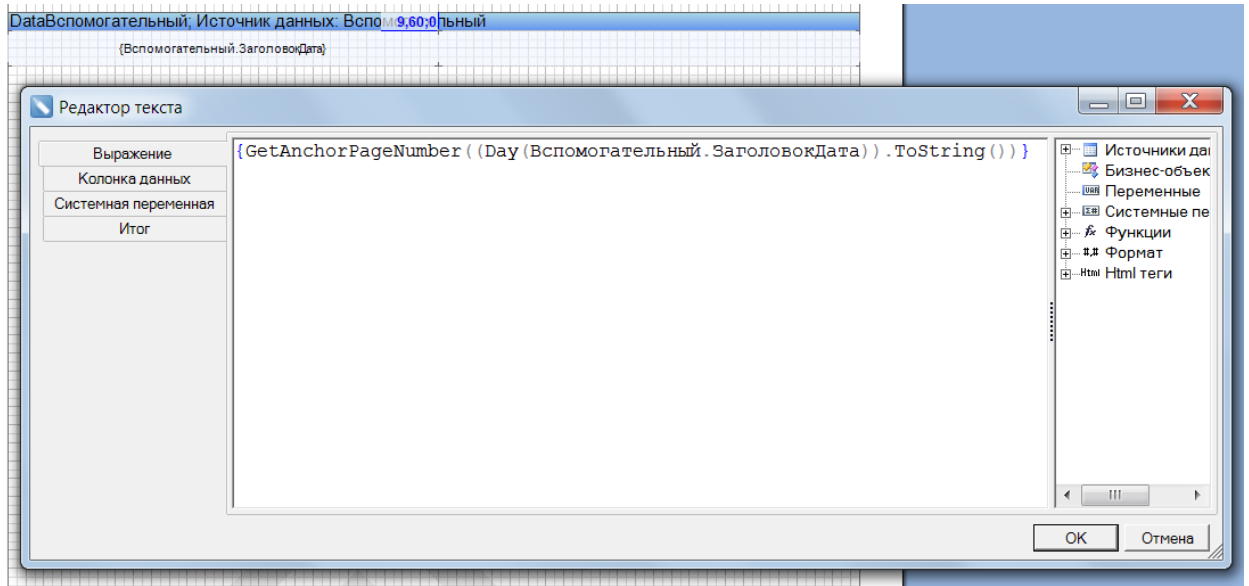


Рисунок 7-42

В свойствах ячейки, установим настройку **Обработать в - В конце отчета**. В этом случае ячейка будет заполнена в конце отчета – когда все якоря будут добавлены.

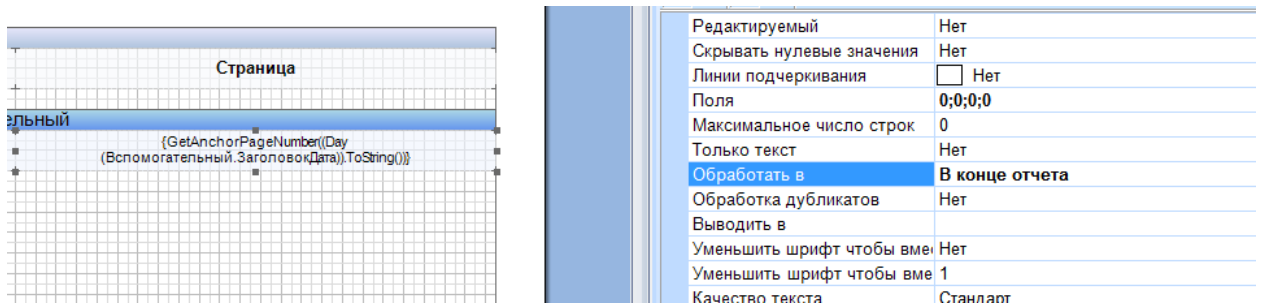


Рисунок 7-43

Запустим предварительный просмотр отчета. На первую страницу добавились имена заголовков (соответствующих дате), и соответствующие им номера страниц (Рисунок 7-44).

Заголовок	Страница
01 ноября 2013	2
02 ноября 2013	2
03 ноября 2013	3
04 ноября 2013	4
05 ноября 2013	4
06 ноября 2013	5
07 ноября 2013	6
08 ноября 2013	7
09 ноября 2013	7
10 ноября 2013	8
11 ноября 2013	9
12 ноября 2013	10
13 ноября 2013	10
14 ноября 2013	11



Рисунок 7-44

Теперь сделаем возможность перехода по гиперссылкам – сделаем чтобы при щелчке по полям в оглавлении, мы переходили на соответствующую страницу с данными.

Для создания гиперссылок необходимо у компонента, на который будет осуществляться переход создать закладку (например, «**Закладка1**»), а у компонента с которого будет осуществляться переход создать гиперссылку на эту закладку. Гиперссылка должна состоять из имени закладки, на которую ссылается с предварительным символом # (например, «**#Закладка1**»).

Сначала создадим закладку. Для этого на странице «ReportData», выделим **бэнд данных**, а затем в группе настроек **Взаимодействие**, настройке **Закладка**, пропишем код:

```
{Day(ПоказанияСчетчика.Время).ToString()}
```

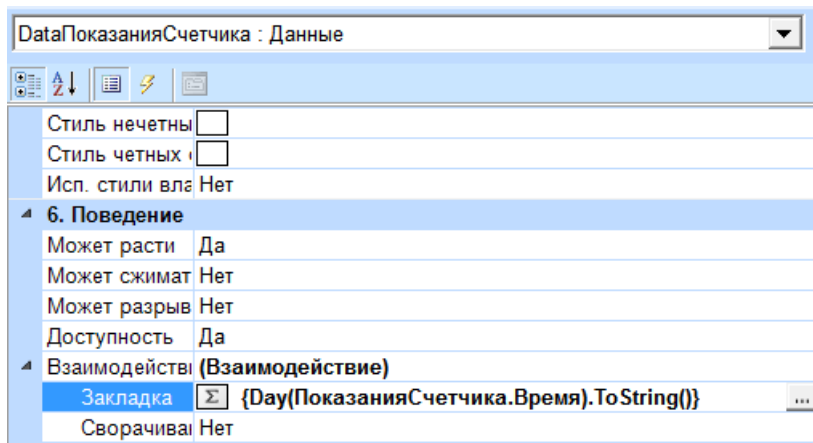


Рисунок 7-45

Теперь настроим гиперссылку. На странице «Contents» пропишем, у ячейки с номером страницы, в настройке Взаимодействие – Гиперссылка пропишем код:

`#{(Day(Вспомогательный.ЗаголовокДата)).ToString()}`

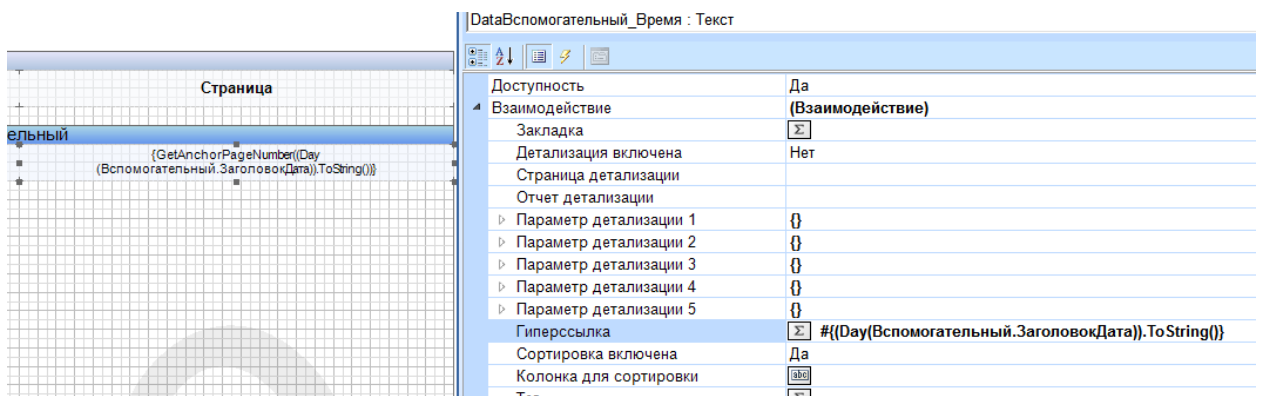


Рисунок 7-46

Аналогичный код в той же настройке пропишем и у поля с заголовком.

Также настроим для данных ячеек стиль текста, соответствующий гиперссылке – синий цвет, жирный шрифт с подчеркиванием.

HeaderВспомогательный	
Заголовок	Страница
DataВспомогательный; Источник данных: Вспомогательный	
<u>{Вспомогательный.ЗаголовокДата}</u>	<u>{GetAnchorPageNumber((Day(Вспомогательный.ЗаголовокДата)).ToString())}</u>

Рисунок 7-47

Запустим предварительный просмотр (Рисунок 7-48).

Заголовок	Страница
01 ноября 2013	2
02 ноября 2013	2
03 ноября 2013	3
04 ноября 2013	4
05 ноября 2013	4

Рисунок 7-48

В левой части окна появилась панель с закладками, с помощью которой можно осуществлять навигацию. При наведении мыши на текст, курсор изменяет свой вид, во всплывающей подсказке появляется имя гиперссылки. При нажатии мыши происходит переход на нужную страницу.

03.11.2013 22:00:00	234
03.11.2013 23:00:00	194
04.11.2013 0:00:00	183
04.11.2013 1:00:00	214
04.11.2013 2:00:00	158

Рисунок 7-49

Примечание. Гиперссылки со страницы оглавления, работают и при экспорте в другие форматы (pdf, html, doc).



Примечание. Готовый отчет можно посмотреть в проекте «Интерактивные отчеты» в объекте «Отчет с оглавлением».

7.4 Заключение.

Данная статья познакомила Вас с несколькими способами создания интерактивных отчетов. Используя полученные навыки, теперь вы можете создать собственные отчеты.

Со своими вопросами вы также можете обращаться в техническую поддержку компании ИнСАТ – support@insat.ru