

Компоненты Sales

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Планирование	4
Изменить логику расчета в разделе [Планирование]	4
1. Скопировать исходный код модуля построения плана	4
2. Создать замещающий модуль построения плана	5
3. Изменить метод открытия страницы плана	6
4. Внести изменения в хранимую процедуру tsp_RecalculateForecastFact	7
Результат выполнения примера	9

Планирование

 Сложный

Функциональность Sales Creatio позволяет планировать объемы продаж компании и анализировать фактическое достижение целей. Используя раздел [*Планирование*], можно формировать различные планы по ключевым срезам, данные о которых внесены в систему, и рассчитывать фактически полученные значения. Это позволяет анализировать выполнение продаж по выбранному периоду и оценивать эффективность отдела продаж в целом на основе сводных таблиц раздела [*Планирование*]. Подробнее возможности раздела описаны в статье "[Планирование](#)" документации пользователя.

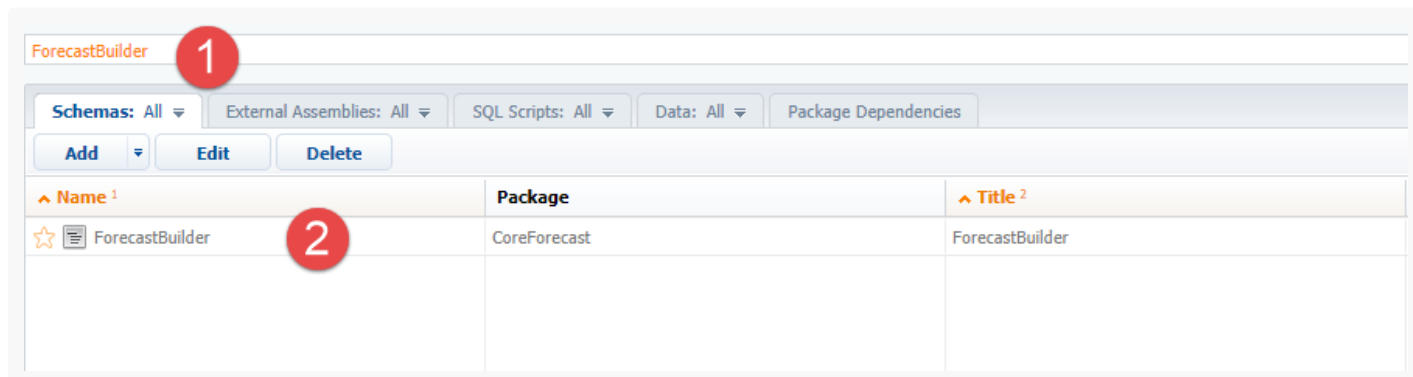
Изменить логику расчета в разделе [Планирование]

 Средний

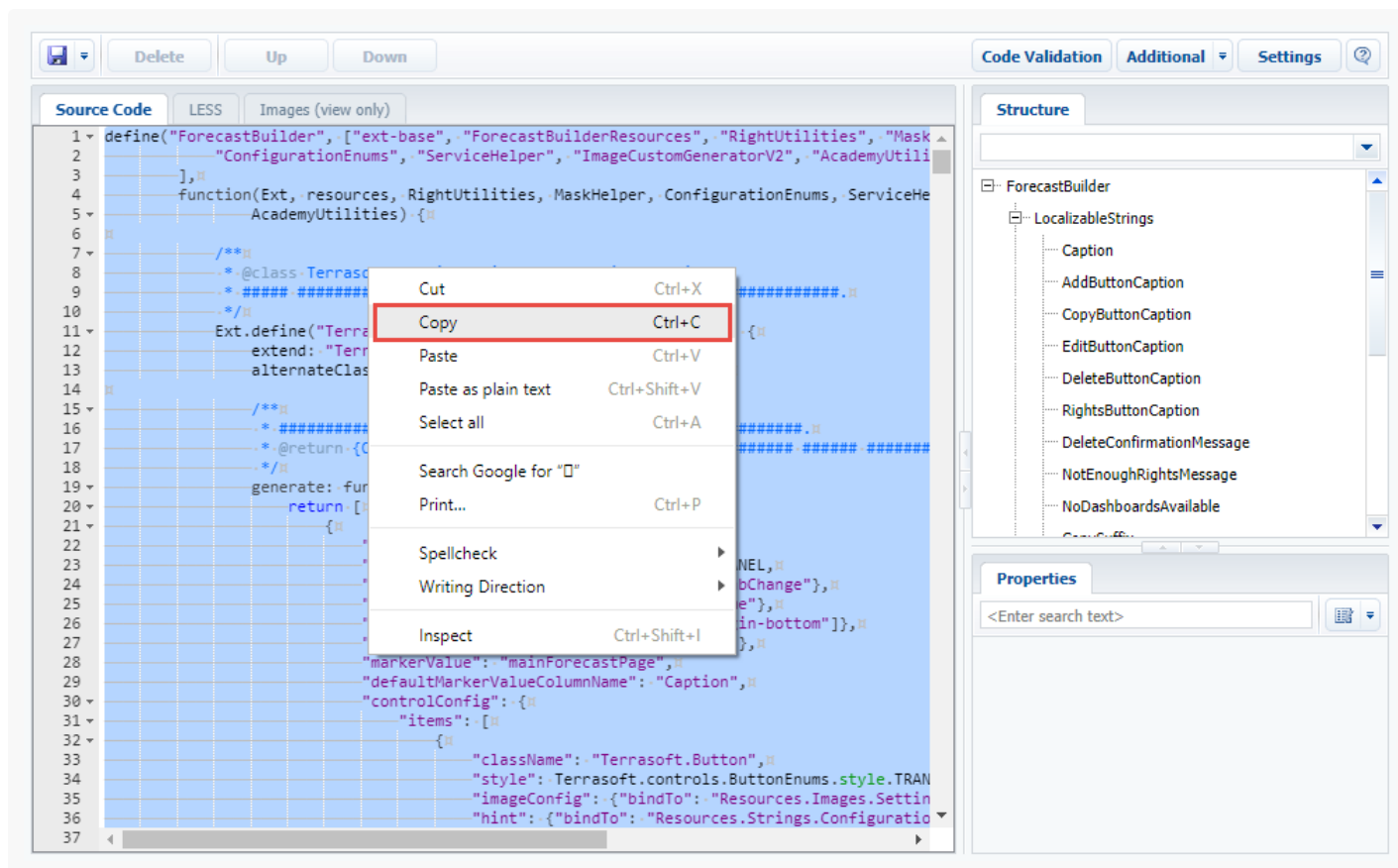
Пример. В разделе Планирование изменить логику расчета факта: производить расчет на основании счетов, а не продаж.

1. Скопировать исходный код модуля построения плана

Для этого необходимо в разделе [*Конфигурация*] ([*Configuration*]) ввести название схемы ForecastBuilder в строку поиска (1). После двойного клика по строке результата поиска (2) в дизайнера модуля откроется схема модуля.

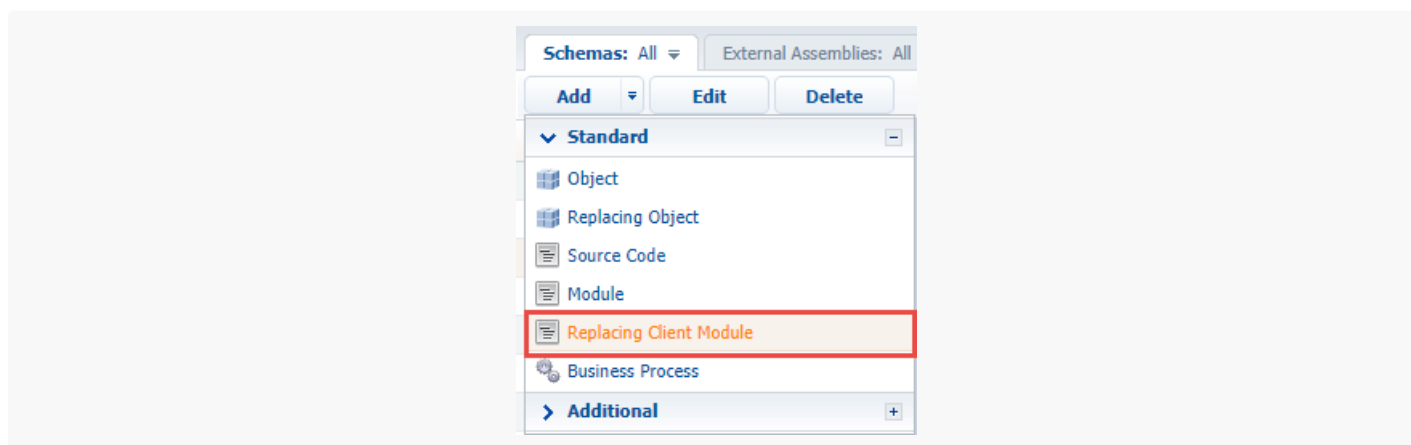


Затем нужно скопировать весь исходный код модуля из вкладки [*Исходный код*] ([*Source Code*]) и сохранить, например, в текстовом файле.

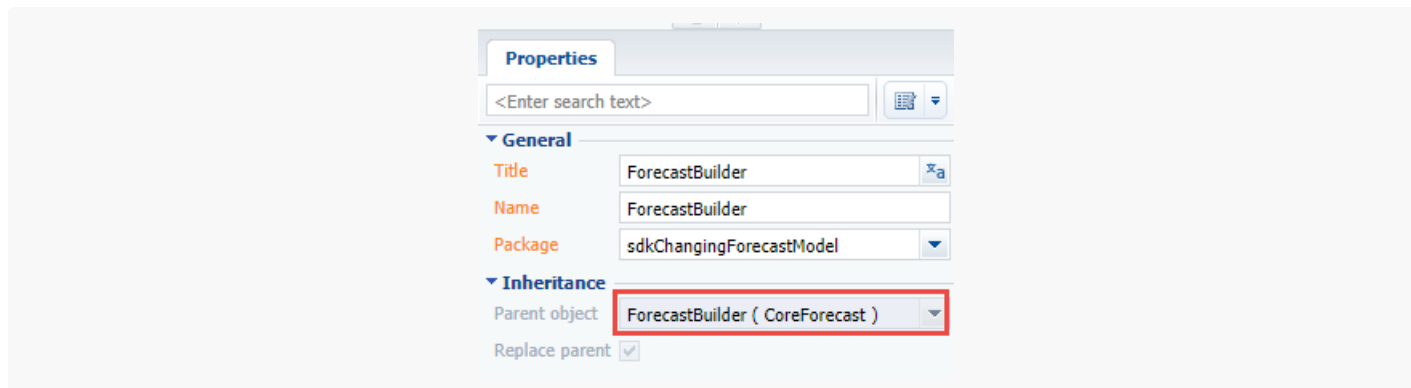


2. Создать замещающий модуль построения плана

Для этого необходимо в разделе [Конфигурация] ([*Configuration*]) выбрать пользовательский пакет и на вкладке [Схемы] ([*Schemas*]) выполнить команду [Добавить] ([*Add*]) — [Замещающий клиентский модуль] ([*Replacing Client Module*]) . Создание замещающего клиентского модуля подробно описано в статье "Создать клиентскую схему".



Для созданной схемы модуля в свойстве [Родительский объект] ([*Parent object*]) нужно установить значение "ForecastBuilder" . После этого автоматически будут применены значения всех свойств схемы, взятые из родительского модуля.



Во вкладке [*Исходный код*] ([*Source Code*]) схемы нужно добавить исходный код родительской схемы, скопированный на предыдущем шаге, и сохранить схему.

3. Изменить метод открытия страницы плана

В исходном коде созданной схемы модуля ForecastBuilder необходимо заменить значения массива `valuePairs` в методе `openForecastPage()` класса `Terrasoft.configuration.BaseForecastsViewModel` на значения, соответствующие схеме объекта [*Счет*] ([*Invoice*]). Исходный код изменений приведен ниже (предыдущие значения закомментированы).

```
...
openForecastPage: function(moduleId, operation) {
    ...
    var valuePairs = [
        {
            name: "EntitySchemaUid",
            value: "bfb313dd-bb55-4e1b-8e42-3d346e0da7c5" //value: "ae46fb87-c02c-4ae8-ad31-a923
        },
        {
            name: "EntitySchemaName",
            value: "Invoice" //value: "Opportunity"
        }
    ];
    ...
},
...
```

Значение идентификатора `EntitySchemaUid` можно получить, выполнив следующий SQL-запрос к базе данных Creatio :

```
select lower(Uid), Name from SysSchema
Where name = 'Invoice'
and ExtendParent = 0
```

После внесения изменений, схему необходимо сохранить.

4. Внести изменения в хранимую процедуру `tsp_RecalculateForecastFact`

Хранимая процедура `tsp_RecalculateForecastFact` выполняет пересчет значений фактов за выбранный временной период.

Важно. При составлении и выполнении SQL-запроса к базе данных следует быть предельно внимательным. Выполнение неправильно составленного SQL-запроса может привести к необратимому повреждению существующих данных и нарушению работы системы.

Поскольку для подсчета значений нужно использовать счета, а не продажи, то в процедуре необходимо внести следующие изменения (предыдущие значения закомментированы).

1. Изменить значение, хранящееся в переменной `@CompletedId`.

```
--SET @CompletedId = '{60D5310C-5BE6-DF11-971B-001D60E938C6}'
SET @CompletedId = '{698D39FD-52E6-DF11-971B-001D60E938C6}'
```

В этой переменной хранится идентификатор статуса полностью оплаченного счета. Значение переменной можно получить, выполнив следующий SQL-запрос к базе данных Creatio:

```
select Id, Name from InvoicePaymentStatus
where Name = 'Paid'
```

2. Изменить запрос, результат которого сохраняется в переменной `@MaxDueDate`.

```
--SET @MaxDueDate = (SELECT Convert(Date, MAX(DueDate), 104) FROM Opportunity o WHERE o.StageId
SET @MaxDueDate = (SELECT Convert(Date, MAX(StartDate), 104) FROM Invoice o WHERE o.PaymentStatu
```

В запросе выполняется поиск самого нового счета среди всех оплаченных счетов.

3. Изменить выражение подзапроса, хранящееся в переменной `@SQLText`. Именно в этом подзапросе реализована логика расчета факта и потенциала.

```
--Исходное значение
/*SET @SQLText = N'
SELECT
  (SELECT SUM(ISNULL(fiv.[Value], 0))
   FROM [ForecastItemValue] fiv
   WHERE fiv.[ForecastItemId] = @P5
```

```

AND fiv.[PeriodId] = @P6
AND fiv.[ForecastIndicatorId] = @P7
) PlanAmount,
(SELECT SUM(ISNULL(o.[Amount], 0))
FROM [Opportunity] o
WHERE o.[StageId] = @P1
AND o.[DueDate] >= @P2
AND o.[DueDate] < @P3
AND o.' + @ColumnName + N' = @P4
) FactAmount,
(SELECT SUM(ISNULL(o.[Amount], 0) * ISNULL(o.[Probability], 0) / 100)
FROM [Opportunity] o
INNER JOIN [OpportunityInStage] ois ON ois.[OpportunityId] = o.[Id]
INNER JOIN [OpportunityStage] os ON os.[Id] = ois.[StageId]
WHERE os.[End] = 1
AND ois.[DueDate] >= @P2
AND ois.[DueDate] < @P3
AND ois.[Historical] = 0
AND o.' + @ColumnName + N' = @P4
) PotentialAmount'*/
--Новое значение
SET @SQLText = N'
SELECT
(SELECT SUM(ISNULL(fiv.[Value], 0))
FROM [ForecastItemValue] fiv
WHERE fiv.[ForecastItemId] = @P5
AND fiv.[PeriodId] = @P6
AND fiv.[ForecastIndicatorId] = @P7
) PlanAmount,
(SELECT SUM(ISNULL(o.[Amount], 0))
FROM [Invoice] o
WHERE o.[PaymentStatusId] = @P1
AND o.[StartDate] >= @P2
AND o.[StartDate] < @P3
AND o.' + @ColumnName + N' = @P4
) FactAmount,
(SELECT SUM(ISNULL(o.[Amount], 0))
FROM [Invoice] o
INNER JOIN [InvoicePaymentStatus] os ON os.[Id] = o.[PaymentStatusId]
WHERE os.[FinalStatus] = 0
AND o.[StartDate] >= @P2
AND o.[StartDate] < @P3
AND o.' + @ColumnName + N' = @P4
) PotentialAmount'

```

Для применения изменений нужно запустить выполнить SQL-сценарий.

Результат выполнения примера

В результате расчет фактов и потенциалов будет выполняться не по продажам, а по счетам.

Результат расчета фактов по продажам

CUSTOM FORECAST FORECAST BY CUSTOMER FORECAST BY SALES DIVISION FORECAST BY SALES MANAGER				
ADD ACTIONS				
	<div>1</div> Expected	April 2017	<div>2</div> Closed, %	Pipeline
Alpha Business	10,000	12,520	125	0
Axiom	5,000	6,200	124	0
Fast Works	0	0	0	0

Результат расчета фактов по счетам

CUSTOM FORECAST FORECAST BY CUSTOMER FORECAST BY SALES DIVISION FORECAST BY SALES MANAGER				
ADD ACTIONS				
	<div>1</div> Expected	April 2017	<div>2</div> Closed, %	Pipeline
Alpha Business	10,000	11,300	113	19,502
Axiom	5,000	0	0	15,350
Fast Works	0	0	0	0