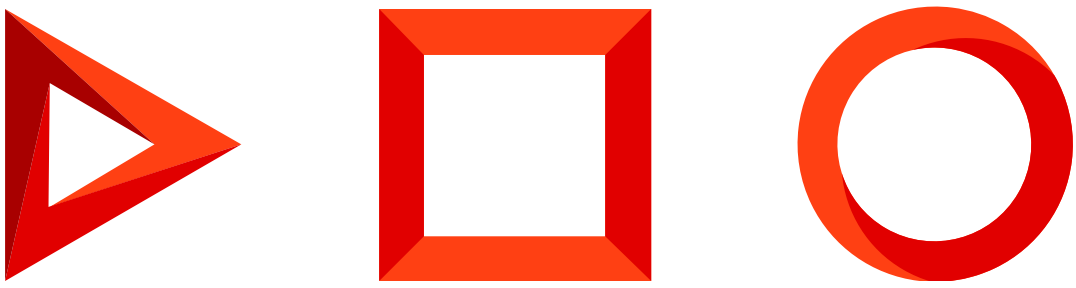
Creatio Academy

Формулы

Формулы в бизнес-процессах

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

Формулы в бизнес-процессах

ПРОДУКТЫ: ВСЕ ПРОДУКТЫ

С помощью окна формул в настройках элементов бизнес-процесса вы можете решить множество задач без привлечения разработчиков. Функциональность формул позволяет автоматически генерировать тексты email-сообщений, определять условия перехода по потокам. Для использования формул необходимо ознакомиться с основными правилами синтаксиса, которые описаны в этой статье.

Основные синтаксические правила

При работе с полем формул следует придерживаться определенного синтаксиса, который будет привычным для пользователей, знакомых с C#. При работе с полем [*Формула*] важно придерживаться типизации при заполнении формул. По возможности используйте значения одного типа, к примеру, текстовые с текстовыми, числовые с числовыми. Иначе необходимо преобразовать значения в нужный тип.

Кроме того, рекомендуем ознакомиться с основными операторами, которые помогут реализовать сложные условия в ваших формулах.

""	Текстовые строки необходимо заключить в кавычки.
+	Используется для соединения значений.
==	Определяет равенство двух значений.
!=	Определяет неравенство двух значений.
<, >	Сравнивает величину двух значений (меньше чем, больше чем).
>=, <=	Больше или равно, меньше или равно.
&&	Логическое “И”.
||	Логическое “Или”.
true, false	Булевы значения “Истина” и “Ложь”.
\n, 	Перенос текстовой строки.

Сформировать сложный текст

Один из самых распространенных сценариев использования поля [*Формула*] — формирование или “связывание” данных.

Пример. В ходе выполнения бизнес-процесса необходимо создать заголовок активности для сбора комплектации товара. Требуется указать продукт и клиента, для которого собирается комплектация.

Для решения задачи в формулу нужно добавить как параметры элемента [*Читать данные*], так и константные текстовые значения:

```
"Собрать комплектацию" + [#Читать продукты.Первый элемент результирующей коллекции.Название#] +
```

Между статичными данными и кавычками "" следует добавлять пробелы, чтобы разделить результирующий текст.

Пример. В поле [*Рекомендации по заполнению страницы*] элемента бизнес-процесса необходимо перенести длинный текст.

Вы можете объединить две строки и более с помощью диалогового окна формулы. Для конкатенации используйте символ +. Для добавления новой строки используйте управляющий символ \n, например:

```
"1. Назначить встречу с руководителем." + "\n" + "2. Обсудить с руководителем тактику ведения сд
```

Текст строки должен быть заключен в прямые двойные кавычки (" "). Для обеспечения корректной работы логических операторов установите признак [*Многострочное*]. Иначе все символы новой строки будут отфильтрованы.

На заметку. Для переноса строк также можно использовать заключенный в скобки HTML-тег “
”.

Сгруппировать различные типы данных

При работе с элементом [*Формула*] необходимо использовать данные одного типа. На странице заполнения формулы тип данных каждого параметра обозначен значком:

- Id — уникальный идентификатор;
- 123 — числовой;
- 0.5 — дробный;
- T — текстовый;
- — справочное значение;
- — значение даты и времени.

Пример. В ходе выполнения бизнес-процесса необходимо создать заголовок активности для сбора комплектации товара. Требуется указать название продукта, клиента и желаемую дату комплектации.

В данном случае нельзя обычным способом добавить параметр даты комплектации. Для решения задачи нужно преобразовать значение даты в текстовое значение:

```
"Собрать комплектацию" + [#Читать продукты.Первый элемент результирующей коллекции.Название#] +
```

Для преобразования параметра [*#Читать заказ.Первый элемент результирующей коллекции.Плановая дата выполнения#*] его необходимо заключить в круглые скобки и добавить метод .ToString(). В этом случае бизнес-процесс, в который включен элемент формирования заголовка активности, отработает корректно.

Настроить расчеты даты и времени

Чтобы выполнить бизнес-процессы с использованием операций с датой и временем, можно воспользоваться структурой DateTime языка программирования C#. Основные свойства и методы представлены в таблице:

.Date	Возвращает дату выбранного параметра.
.Hour	Возвращает значение часов выбранного параметра даты.
DateTime.MinValue	Минимальное значение даты и времени, 00:00, UTC 1 января 0001 года.
.TotalMinutes	Возвращает полное значение даты и времени в минутах.
.TotalHours	Возвращает полное значение даты и времени в часах.
.TotalDays	Возвращает полное значение даты и времени в количестве дней.
.AddMinutes(), .AddHours(), .AddDays()	Увеличивают выбранное значение даты и времени на определенное количество минут, часов или дней.

Пример. В процессе квалификации лида для перехода между условными потоками необходимо проверить, заполнено ли поле даты принятия решения.

Чтобы произвести проверку, воспользуйтесь оператором != и свойством DateTime.MinValue:

```
[#Читать Лид после Квалификации.Первый элемент результирующей коллекции.Дата принятия решения#]!
```

Пример. В рамках бизнес-процесса для перехода между условными потоками необходимо сравнить значения даты закрытия двух продаж.

Для сравнения двух значений воспользуйтесь оператором == и свойством .Date в виде:

```
[#Читать данные продажи 1.Первый элемент результирующей коллекции.Дата закрытия#].Date
== [#Читать данные продажи 2.Первый элемент результирующей коллекции.Дата закрытия#].Date
```

Пример. В ходе выполнения бизнес-процесса необходимо рассчитать время, которое потребовалось для закрытия продажи.

В случае, когда необходимо рассчитать разницу между двумя значениями даты, воспользуйтесь такой конструкцией:

```
(decimal)Округлить(([#Читать данные продажи.Первый элемент результирующей коллекции.Дата
 закрытия#]-[#Читать данные продажи.Первый элемент результирующей коллекции.Дата создания#]).Tot
```

То есть, необходимо в окне формулы выбрать функцию [*Округлить*] и заполнить ее нужными параметрами элемента процесса. В нашем случае — разницей значений, а затем добавить свойство .TotalMinutes. В итоге вы получите количество затраченного времени в минутах. Таким образом можно также использовать свойства .TotalHours и .TotalDays.

Пример. Для перехода по условным потокам бизнес-процесса необходимо определить, был ли лид создан более 12 часов назад.

Для выполнения задачи добавьте свойство .TotalHours к параметру элемента, значение параметра целиком заключите в круглые скобки, а затем произведите сравнение с числовым значением:

```
(decimal)Округлить(([#Системная переменная.Текущее значение даты и времени#]-[#Читать данные лид
```

При работе с параметрами даты и времени также можно использовать функции .AddMinutes(), .AddHours() и .AddDays() для увеличения значения времени и даты на определенную величину. Например, чтобы привести данные даты и времени к часовому поясу пользователя при использовании элемента [*Чтение данных*].

На заметку. В Creatio данные с типом “Дата/Время” хранятся в UTC. Элемент [*Чтение данных*] не приводит эти данные к часовому поясу пользователя.

Пример. При выполнении бизнес-процесса регистрации лида необходимо установить время перезвона через 3 часа после создания лида.

Для решения задачи укажите в значении .AddHours() добавляемое количество часов в виде:

```
([#Читать лид.Первый элемент результирующей коллекции.Дата создания#].AddHours(3)).Hour
```

Настроить формулу в условном потоке

Функциональность элемента [*Формула*] в условном потоке ничем не отличается от его функциональности в других элементах бизнес-процесса. То есть, актуальны как основные правила, так и операторы. Условные потоки применяются для перехода к следующему элементу процесса, поэтому проверяют условия перехода.

Часто в условных потоках справочные параметры сравнивают с константными значениями справочника.

Пример. В бизнес-процессе корпоративной продажи необходимо проверить, что продажа находится в стадии “Коммерческое предложение”.

Для решения задачи нужно в условном потоке сравнить текущую стадию продажи со справочным значением:

```
[#Читать данные продажи.Первый элемент результирующей коллекции.Стадия#]==[#Справочник.Opportuni
```

Пример. В условном потоке необходимо проверить, заполнено ли справочное поле контакта лида.

Чтобы проверить, заполнены ли справочные поля, используйте условие:

```
[#Читать лид.Первый элемент результирующей коллекции.Контакт#]!= Guid.Empty
```

Если же нужно проверить, что справочное поле контакта лида не заполнено, то необходимо воспользоваться следующей конструкцией:

```
[#Читать лид.Первый элемент результирующей коллекции.Контакт#]==Guid.Empty
```

Пример. В процессе визирования счета необходимо выполнить проверку, что виза получена.

Для решения задачи необходимо в условном потоке использовать условие:

```
[#Получение визы по счету.Результат#] == "Утверждена"
```

То есть, производится проверка результата визирования.

Обратите внимание, что вариантов решения задачи может быть несколько. Например, воспользуйтесь более сложной конструкцией:

```
[#Получение визы по счету.Результат#] == "Отклонена" || [#Получение визы по счету.Результат#] ==
```

В этом случае производится проверка, что статус визы не “Отклонена” или не “Ожидает визирования”.

Пример. В условном потоке необходимо проверить, что проведена презентация.

Чтобы проверить статус презентации, необходимо использовать такую конструкцию:

```
[#Провести презентацию.Результат#] == true
```

Если презентация не проведена, то процесс перейдет по другому условному потоку.