

Редактор выражений

Лоция Софтвэа

*Опубликовано
Лоция Софтвэа
127422, г. Москва, Тимирязевская ул., д.1, стр. 2.
Телефон: (495) 748-04-74
Факс: (495) 748-03-74
E-mail: sales@lotsia.com
Интернет: <http://www.lotsia.com>*

*Авторские права
Авторскими правами обладает ООО «Лоция Софтвэа».*

Никакая часть данного документа не может быть воспроизведена или передана в любой форме и любыми способами в каких-либо целях без предварительного письменного разрешения ООО «Лоция Софтвэа».

Лицензионное соглашение, поставляемое с программным обеспечением, определяет процедуру пользования продуктом.

© 1997-2018 ООО «Лоция Софтвэа». С сохранением всех прав.

Лоция Софтвэа, Lotsia PDM, Lotsia PDM PLUS, LS Flow являются зарегистрированными торговыми марками ООО «Лоция Софтвэа».

Все остальные упомянутые в документе торговые марки являются собственностью их законных владельцев.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления.

*PN: 05-2500-06-RU
08.06.2018*

Содержание

1	<u>РЕДАКТОР ВЫРАЖЕНИЙ</u>	<u>4</u>
1.1	ОБЩИЕ ПРАВИЛА ИСПОЛЬЗОВАНИЯ РЕДАКТОРА ВЫРАЖЕНИЙ	9
1.2	ПРОВЕРКА ПРАВИЛЬНОСТИ ВЫРАЖЕНИЯ	10
1.3	МАКСИМАЛЬНАЯ ДЛИНА ЗНАЧЕНИЯ, ВОЗВРАЩАЕМАЯ ВЫРАЖЕНИЕМ	10
1.4	НЕКОТОРЫЕ ПРИМЕРЫ И РЕКОМЕНДАЦИИ ПО СОСТАВЛЕНИЮ ВЫРАЖЕНИЙ	10
1.5	ФИЛЬТРЫ	11
1.6	СОХРАНЕНИЕ ВЫРАЖЕНИЙ	12
1.7	ЗАГРУЗКА СОХРАНЕННЫХ ВЫРАЖЕНИЙ	13
1.8	РЕДАКТИРОВАНИЕ СОХРАНЕННЫХ ВЫРАЖЕНИЙ	13
1.9	УДАЛЕНИЕ СОХРАНЕННЫХ ВЫРАЖЕНИЙ	13
2	<u>ОПИСАНИЕ ФУНКЦИЙ</u>	<u>14</u>
	Группа функций «Даты/Времени»	14
	Группа функций «Карта работы»	18
	Группа функций «Массивы»	21
	Группа функций «Математические»	24
	Группа функций «Преобразования»	25
	Группа функций «Работа с БД»	25
	Группа функций «Разные»	30
	Группа функций «Статистические»	34
	Группа функций «Строковые»	37
	Группа функций «Формы»	39
	Группа функций «Формы задачи»	41

1 Редактор выражений

Пользователи, используя некоторые функции программы (настройку отчетов, импорт документов, задание условий переходов, настройку действий и т.д.), могут сталкиваться с необходимостью применения Редактора выражений.

Редактор выражений – это встроенный инструмент формирования выражений с возможностью использования специальных математических, строковых, временных, статистических и прочих функций. С его помощью можно составлять выражения различной сложности – от простейших вычислений до ввода условия, включающего проверку значений различных параметров.

Внешний вид Окна, наименование и количество вкладок Редактора выражений может быть различными, в зависимости от конкретного случая его применения и приобретенной Вами версии программы, но приемы составления выражений остаются неизменными. Примеры Окон Редактора выражений см. Рисунок 1-Рисунок 5.

В верхней части Окна Редактора выражений располагается поле для ввода выражения. Нижняя часть Окна содержит вкладки, в которых сгруппированы функции, операторы и другие элементы, из которых может быть построено выражение для использования в текущей функции программы.

Составление выражения может производиться:

- прямым вводом требуемого текста выражения (с клавиатуры или копированием), с соблюдением соответствующих синтаксических правил;
- последовательным выбором переменных, операторов, функций, атрибутов на различных вкладках редактора выражений;
- одновременно с помощью обоих вышеуказанных способов.

При выделении выражения в верхней части Окна или его фрагмента, ввод осуществляется поверх выделенного фрагмента. Для вставки фрагмента в требуемое место выражения необходимо установить курсор в верхней части Окна в соответствующую позицию выражения, и произвести вставку любым из перечисленных выше способов.

При наведении указателя мышки на фрагмент выражения, являющийся переменной или функцией, программа отобразит всплывающую подсказку с информацией о переменной/функции.

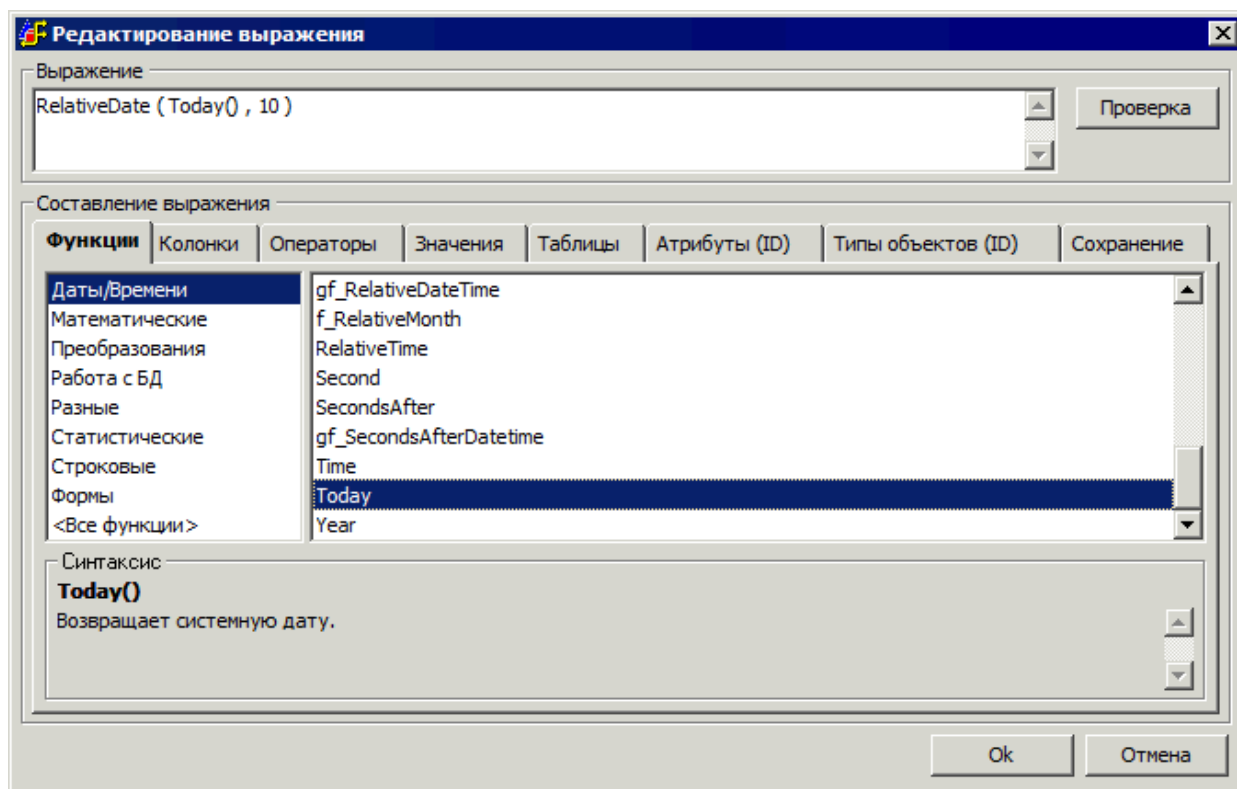


Рисунок 1 Окно Редактора выражений для вычисляемого поля. Пример

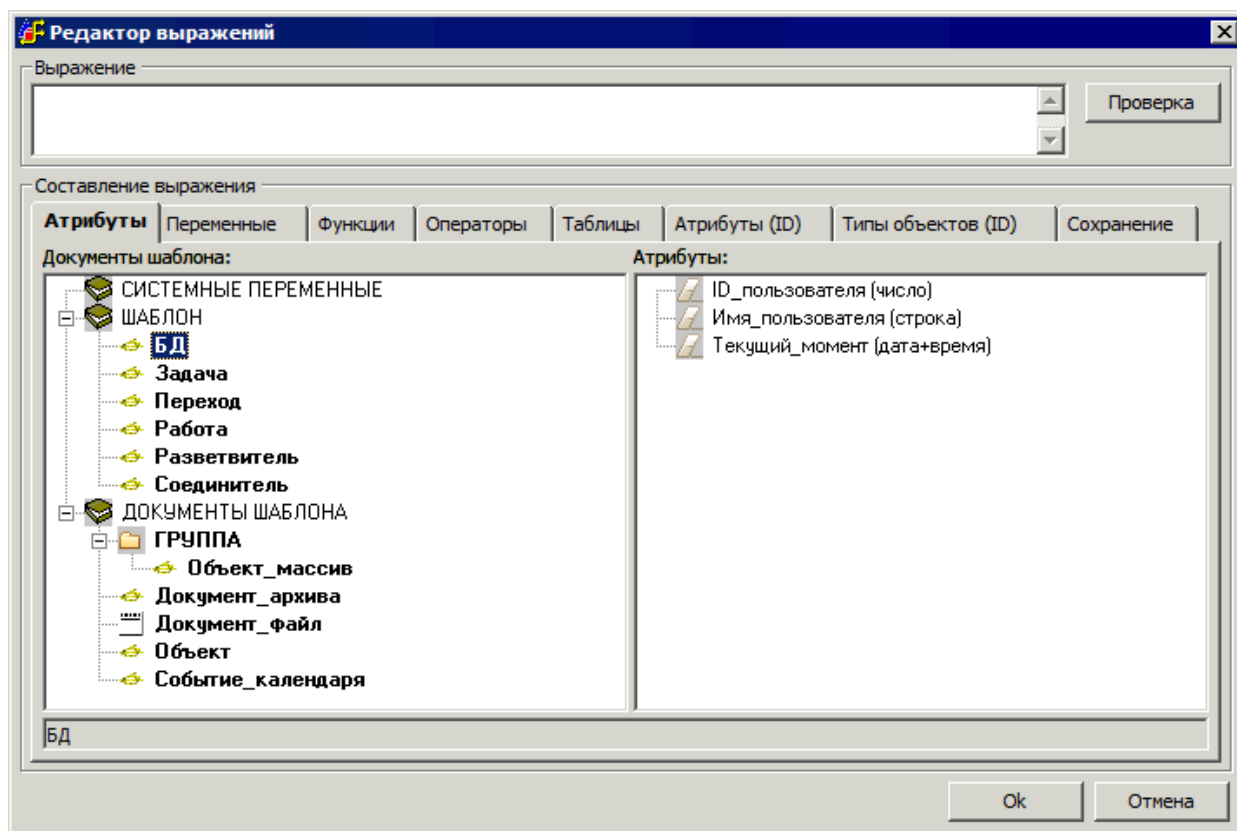


Рисунок 2 Окно Редактора выражений для документооборота. Вкладка «Атрибуты». Пример

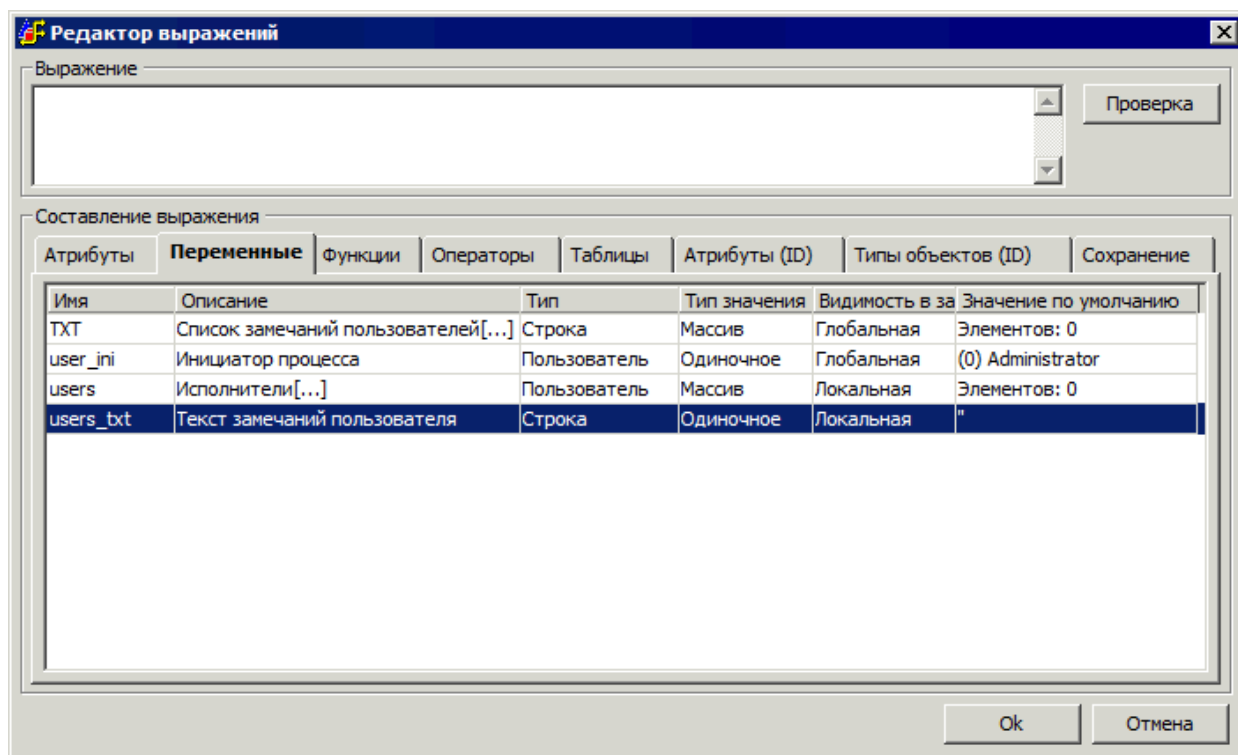


Рисунок 3 Окно Редактора выражений для документооборота. Вкладка «Переменные». Пример

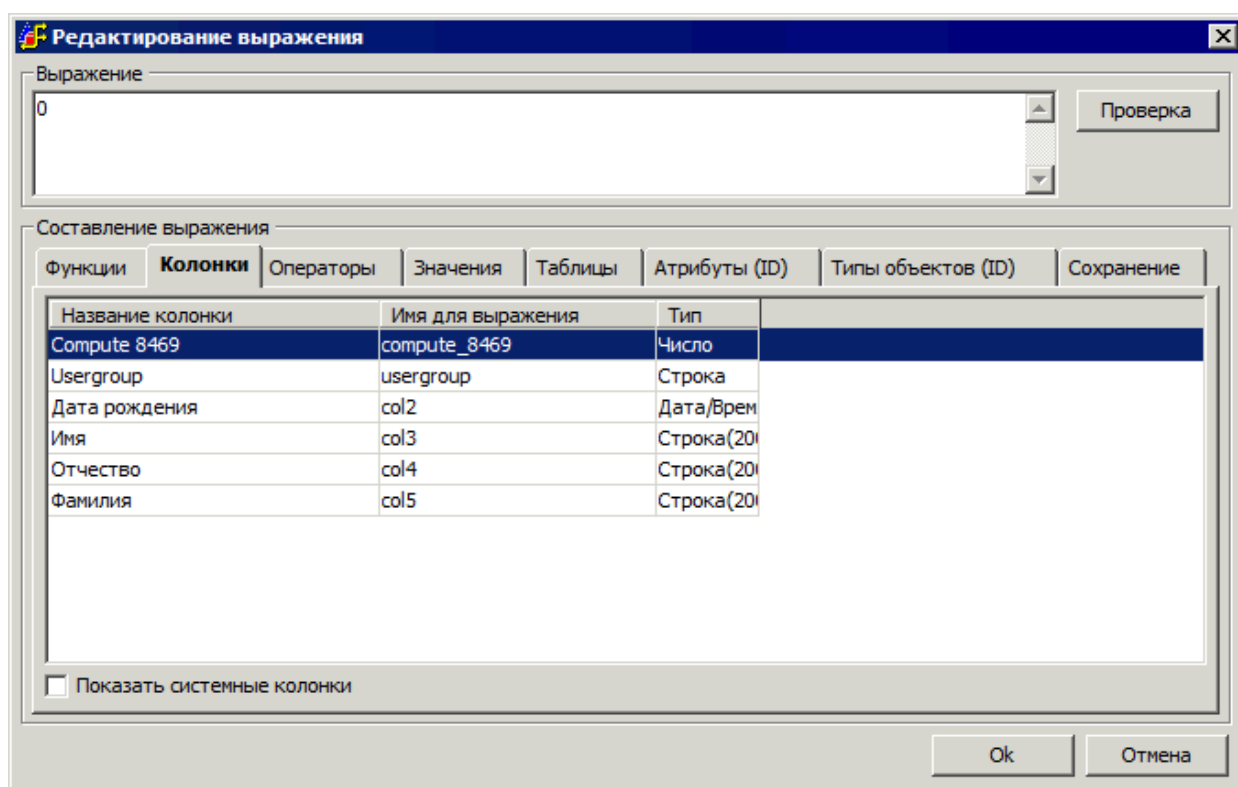


Рисунок 4 Окно Редактора выражений. Вкладка «Колонки». Пример

Вкладка «Функции» (Рисунок 1) – доступна всегда. В левой части располагаются группы функций, а в правой функции, относящиеся к этой группе. Описание синтаксиса и краткое описание текущей функции располагается в нижней части. Например, для того чтобы в выражение добавить текущую дату, необходимо на вкладке «Функции» выбрать

группу «Даты/Времени», выделить функцию Today() и дважды щелкнуть по ней левой кнопкой мыши. Выбранная функция будет добавлена в выражение в то место, где последний раз был курсор. Если в выражении был выделен текст, то он будет заменен выбранной функцией. Аналогично может выполняться выбор других элементов выражения. Аргументы функции и другие элементы выражения можно ввести любым удобным способом.

Вкладка «Операторы» – доступна всегда. Здесь представлены операторы, используя которые можно строить выражения. Ниже приведен пример выражения, результатом вычисления которого в первый день месяца будет строка, содержащая текст «Первый день месяца», а в остальные дни месяца текст «Не первый день месяца»:

If(Day (Today()) = 1, "Первый день месяца", "Не первый день месяца")

Здесь, функция Today() возвращает текущую дату, функция Day() из этой даты определяет день, а в функции «IF» с помощью оператора «=» выполняется сравнение даты с первым числом.

Вкладка «Колонки» или «Переменные» – доступна не всегда. Здесь представлен список используемых колонок (Рисунок 4). Для каждой колонки помимо ее названия указывается имя для использования в выражениях и тип данных. Чтобы отобразить в списке предопределенные колонки (переменные), включите флажок **«Показывать системные колонки»**, если он отображается. Щелчком левой кнопки мыши на заголовке поля списка колонок (переменных) можно отсортировать список по этому полю.

Вкладка «Атрибуты» – доступна не всегда. Здесь представлен список атрибутов программы. При использовании, в левой нижней части Окна выделите требуемый элемент (Рисунок 2). В правой части Окна отобразится список атрибутов, соответствующих выделенному элементу. Дважды щелкните по нужному атрибуту в правой части Окна – выбранный атрибут появится в верхней части Окна, в текущей позиции курсора или поверх выделенного фрагмента выражения. Подобное выражение будет возвращать значение атрибута.

Вкладка «Значения» – доступна не всегда. Выбор значений может осуществляться при наличии списка значений для выбранной заранее колонки. Если список значений пуст – ввод значений в выражение может быть осуществлен только ручным вводом или вставкой из буфера обмена.

Вкладка «Таблицы» – доступна всегда. Здесь есть возможность выбрать таблицы, представления таблиц и колонки. Имена таблиц и колонок могут использоваться в функциях, выполняющих SQL-запросы.

Вкладка «Атрибуты (ID)» – доступна всегда. Здесь выбираются атрибуты. В выражение возвращается идентификатор атрибута и комментарий, содержащий название атрибута. Идентификаторы атрибутов могут быть использованы, например, в функциях, выполняющих SQL-запросы, в выражениях передачи значений из шаблонов работ в действия над объектами и т.д.

Вкладка «Типы объектов (ID)» – доступна всегда. Здесь выбираются типы объектов. В выражение возвращается идентификатор типа объекта и комментарий, содержащий название типа объекта. Идентификаторы типы объектов могут быть использованы, например, в функциях, выполняющих SQL-запросы, в логических функциях, проверяющих тип объекта в шаблонах работ, в вычисляемых полях форм и т.д.

Вкладка «Отчеты (ID)» – доступна всегда. Здесь выбираются отчеты. В выражение возвращается идентификатор отчета и комментарий, содержащий название отчета. Идентификаторы отчетов могут быть использованы, например, при настройке вызовов действий над объектами, запускающих отчеты из кнопок форм типа «Атрибуты», пунктов типа «Действие» Главных меню.

Вкладка «Шаблоны (ID)» – доступна всегда. Здесь выбираются шаблоны работ. В выражение возвращается идентификатор шаблоны работы и комментарий, содержащий название шаблона работы. Идентификаторы шаблонов работ могут быть использованы, например, при настройке вызовов действий над объектами, запускающих работы из кнопок форм типа «Атрибуты», пунктов типа «Действие» Главных меню.

Вкладка «Сохранение» – доступна всегда. Здесь пользователь может [сохранить выражение](#), присвоив ему имя.

Вкладка «Примечания» – доступна не всегда. Здесь можно ввести примечание, комментирующее выражение.

В приведенном ниже примере, с помощью редактора выражений устанавливается фильтрация почты по одному из имеющихся значений колонки «Тема». После выбора данной колонки (ее имя для выражения – «wf_mail_and_map_v_subject») и оператора «=», переходим на вкладку «Значения» (Рисунок 5) (в различных случаях внешний вид Окна может несколько отличаться).

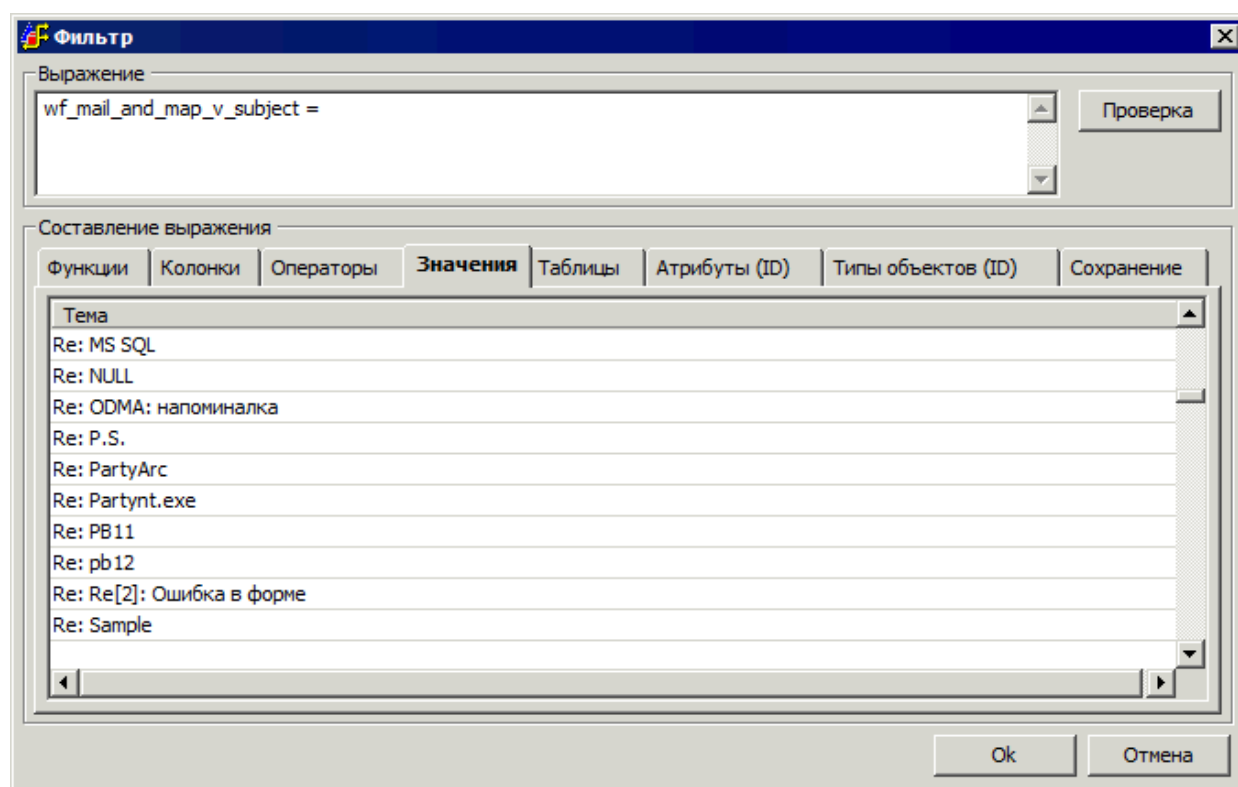


Рисунок 5 Пример выбора значения из списка при редактировании выражения для фильтра

После двойного щелчка по выбранному значению, оно отобразится в верхней части Окна, в текущей позиции курсора или поверх выделенного фрагмента выражения.

Вкладка «Переменные» (доступна не всегда). Здесь представлен список переменных, их описание и тип.

ВНИМАНИЕ! Переменные, начинающиеся с "%OLE_" вычисляются с использованием технологии OLE, которую поддерживают не все приложения! В частности, не поддерживаются OLE-переменные при импорте документов с использованием ODMA.

1.1 Общие правила использования Редактора выражений

Для выбора атрибута, переменной, функции, оператора или иного элемента выражения, зачастую достаточно перейти на соответствующую вкладку и дважды щелкнуть по требуемому элементу, после чего он появится в верхней части Окна, в текущей позиции курсора или поверх выделенного фрагмента выражения. Результат ручного редактирования и ввода значений не отличается от редактирования и ввода текста в обычном текстовом редакторе.

Для ввода числового значения переменной, имеющей соответствующий тип данных, достаточно ввести с клавиатуры соответствующие цифры.

Для ввода строкового значения переменной, имеющей тип данных «Строка», достаточно ввести с клавиатуры соответствующие символы, помещенные в двойные или одинарные кавычки.

Последовательность символов $\sim r \sim n$ воспринимается программой как символ переноса строки.

Выражение *char(9)* используется для вставки табулятора при формировании текстовой строки.

Если аргументом какой-либо функции является строка в кавычках, в свою очередь содержащая кавычки, то «внутренние» кавычки должны отличаться от «наружных» кавычек. Например, если наружные кавычки одинарные, то внутренние должны быть двойными или одинарными со знаком \sim (тильда) перед каждой из них.

При составлении сложных выражений рекомендуется использовать комментарии.

Для вставки комментариев (не выполняемых фрагментов) в текст выражения введите «/*» в начале комментария, далее текст комментария и «*/» после окончания текста комментария (кавычки вставлять не нужно). При вставке двух символов «/» (без кавычек), вест текст, расположенный правее указанных символов, рассматривается программой как комментарий.

Вставка символов комментария – удобный способ временно «отключить» выражение или его часть. Если вы комментируете все выражение, то помните, что выражение не может быть пустым. Другим словами, введите хотя бы цифру или простейшее выполняющееся условие (например, $1=1$) за пределами комментируемого выражения. Тогда Окно редактора выражений можно будет закрыть без потери содержимого.

При использовании Редактора выражений в генераторе отчетов следует помнить, что если в вычисляемом поле имеется ссылка, например, на числовой атрибут из колонки col5 в таком виде:

'Значение атрибута ' + col5 ,

то при отсутствии атрибута, поле будет пустым, не отобразится даже фраза *'Значение атрибута '.*

Поэтому рекомендуется использовать в выражениях функцию IsNull. Выражение может принять следующее содержание:

'Значение атрибута ' + (If (IsNull (col5), '0', col5))

То же самое относится к отображению в вычисляемых полях колонок с атрибутами связанных объектов.

Использование операторов Редактора выражений в Редакторе форм имеет особенность, которую необходимо учитывать: выполнение операторов при вычислении выражения производится в следующей очередности:

Оператор	Назначение
()	Группировка
^	Возведение в степень
*, /	Умножение, деление

+, -	Сложение, вычитание, соединение строк
IN, LIKE, BETWEEN	Условия SQL SELECT
=, >, <, <=, >=, <>	Сравнение
AND, OR	Логическое «И», логическое «ИЛИ»
NOT	Логическое отрицание

1.2 Проверка правильности выражения

Для проверки правильности выражения щелкните на кнопке «Проверка». Если выражение содержит ошибку, то будет выдано соответствующее предупреждение. Сохранить неверное выражение можно, только с помощью комментариев (см. раздел «Общие правила использования Редактора выражений»). По окончании ввода выражения нажмите «Ok».

1.3 Максимальная длина значения, возвращаемая выражением

При составлении выражений следует учитывать, что переменные действий могут хранить в памяти максимум 64000 символа. Но при записи в базу данных значение будет обрезано до 1998 символов или меньше, в зависимости от содержимого самого значения (некоторые символы, например, кавычки, при записи в базу данных сопровождаются служебным символом).

Максимальная длина результата вычисляемого поля также составляет 64000 символов.

1.4 Некоторые примеры и рекомендации по составлению выражений

При составлении выражений, особенно с использованием логических операторов, необходимо учитывать, что результат выражения может быть только одного типа. При несоответствии типов элементов, входящих в выражение, необходимо использовать функции преобразования. Ниже приведен пример выражения, в котором используется функция преобразования даты в строку:

«Текущая дата: » + string (Date (Today()))

При импорте документов можно использовать переменную «%OLE_PAGE_COUNT%». Переменная – строкового типа, возвращает число страниц. Реализована для документов MS Word, для других приложений возвращается пустая строка. Для документов MS Word в выражениях можно также использовать значения пользовательских свойств документа, например: «Номер документа», «Расположение» и т.д. Для получения значения в строковом виде в выражении нужно ввести переменную %OLE_имя%, где *имя* – имя пользовательского свойства документа. Например: %OLE_Номер документа% или %OLE_Расположение% и т.д.

Пример выражения:

%OLE_TITLE_DOC% + ' Создан: ' + string (%DOS_FILE_CREATION_DATE%)

В результате вычисления получим строку вида

«<Название документа> Создан: <Дата создания документа>».

При использовании в импорте документов переменной «%OBJECT_DESC%», возвращающей название объекта, следует учитывать, что для существующего объекта


возвращается полное название объекта, согласно настройкам пользователя, а для нового (создающегося) объекта – только его описание.

Ниже приведен пример выражения, результатом вычисления которого в первый день месяца будет строка, содержащая текст «Первый день месяца», а в остальные дни месяца текст «Не первый день месяца»:

If (Day (Today()) = 1, "Первый день месяца", "Не первый день месяца")

Здесь, функция Today() возвращает текущую дату, функция Day() из этой даты определяет день, а в функции «IF» с помощью оператора «=» выполняется сравнение даты с первым числом.

1.5 Фильтры

Частным случаем использования Редактора выражений является составление выражений для фильтрации строк. Выражения фильтров используются при настройке и получении отчетов и в других случаях. Для вызова Окна фильтра выберите в строке меню пункт «Вид» > «Фильтр» или щелкните на кнопке  Панели инструментов.

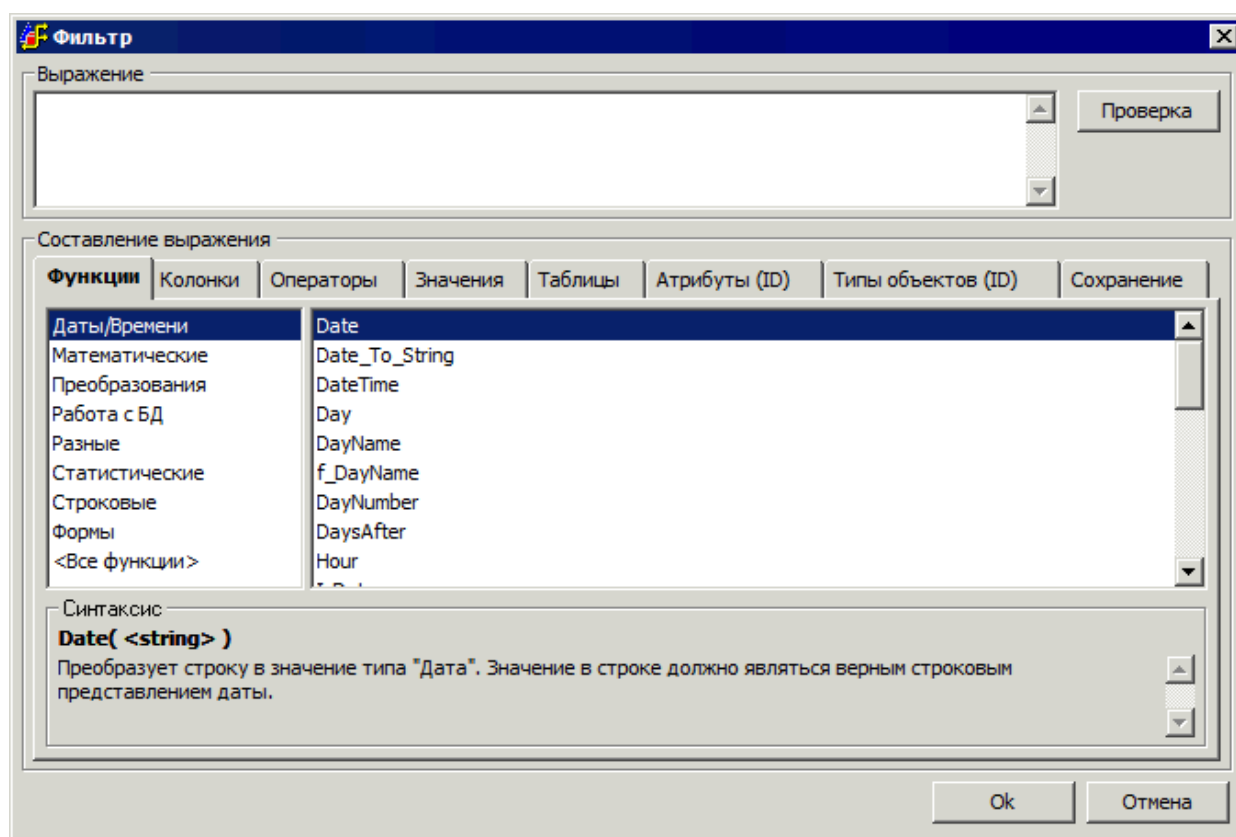


Рисунок 6 Окно задания выражения фильтра

Для составления выражения фильтра используется вышеописанный инструментарий Редактора выражений. Отличия Редактора выражений и фильтра в том, что в Окне фильтра отсутствует кнопка «Подробно», так как Окно фильтра сразу находится в детализированном режиме. Само выражение фильтра должно выглядеть в виде условия (т.е. иметь логический тип данных), например:

col3 = 'Конус'

или

wf_mail_and_map_v_user_from_id = 25

В качестве оператора можно использовать «LIKE». В этом случае в выражении можно использовать специальные знаки: «%» (любое количество любых знаков) и «_» (один любой знак). Например:

col3 like 'Цил%'

Пример выражения фильтра для столбцов с типом данных «Дата»:

col11 <= date ('1999-04-01')

1.6 Сохранение выражений

Для сохранения заданного выражения перейдите на вкладку «Сохранение» (Рисунок 7). Введите описание выражения в поле «Наименование выражения».

Рисунок 7 Сохранение выражения

Для некоторых окон система позволяет установить область применения выражения с помощью переключателей «Для текущего Окна» и «Для любого Окна».

Щелкните на кнопке «Сохранить» для сохранения выражения. Активизируются кнопки «Права» и «Удалить» (Рисунок 8).

Рисунок 8 Сохраненное выражение

Щелкнув на кнопке «Права» вы сможете определить права на использование данного выражения другими пользователями.

1.7 Загрузка сохраненных выражений

Загрузка ранее сохраненных выражений происходит в том же Окне, что и их сохранение. Щелкните на кнопке **«Открыть»** и, в появившемся Окне выбора выражения, выберите выражение по его наименованию и области применения и щелкните на кнопке **«Ок»**.

1.8 Редактирование сохраненных выражений

Редактирование сохраненного выражения возможно при наличии соответствующих прав.

Загрузите выражение способом, описанным в разделе «Загрузка сохраненных выражений».

В загруженном выражении произведите изменения и сохраните его в одном из следующих режимов (выбор переключателем):

- «Сохранить выражение как новое»** – в этом случае введите новое имя выражения. Возможно, в этом режиме активизируются переключатели области применения выражения;
- «Обновить исходное выражение»** – сохранение произойдет под именем загруженного выражения.

1.9 Удаление сохраненных выражений

Удаление сохраненного выражения возможно при наличии соответствующих прав.

Загрузите выражение способом, описанным в разделе «Загрузка сохраненных выражений». Щелкните на кнопке **«Удалить»**. Подтвердите свое намерение в открывшемся Окне.

2 Описание функций

На вкладке «Функции» Окна Редактора выражений используются следующие функции (см. Таблица 1).

Таблица 1 Функции Редактора выражений

Имя функции	Синтаксис	Описание
Группа функций «Даты/Времени»		
Date	Date(<string>)	Преобразует строку в значение типа "Дата". Значение в строке должно являться верным строковым представлением даты.
Date_To_String	Date_To_String (<datetime>)	Преобразовать дату по формату DD MMMMMMMM YYYY
DateTime	DateTime (<date {, time}>)	Объединяет значения даты и времени и возвращает значение типа "Дата/Время"
f_DatetAdd	f_DateAdd (<datetime>, <number> , <string>)	Возвращает дату, отстоящую от даты <datetime> на <number> интервалов (положительное или отрицательное целое число). Параметр <string> содержит параметры, разделенные через «,». Возможные параметры: "Y", «Year» – <number> содержит количество лет; "Q", «Quarter» – <number> содержит количество кварталов; "M", «Month» – <number> содержит количество месяцев; "W", «Week» – <number> содержит количество недель; "D", «Day» – <number> содержит количество календарных дней; "DW", «DayWork» – <number> содержит количество рабочих дней; "DO", «DayOff» – <number> содержит количество нерабочих дней; "DH", «DayHoliday» – <number> содержит количество праздничных дней; "P", «Plan» – использовать только производственный календарь (режим по умолчанию), только для интервалов DW, DO, DH; «M», «Mix» – использовать производственный календарь, если существует, только для интервалов DW, DO, DH; "B", «Base» – использовать только базовый

		<p>календарь (сб, вс – выходные), только для интервалов DW и DO; "I", «Inclusive» – включать текущую дату в интервал отсчета (режим по умолчанию), только для интервалов DW, DO и DH; "E", «Exclusive» – исключить текущую дату из интервала отсчета, только для интервалов DW, DO и DH. Если в параметре <string> указано «I», а в параметре <number> указано 1 или -1, то функция вернет ближайший день требуемого типа. Например, если к выходному прибавить 1 день – получится понедельник, а если к пятнице – получится тот же день, пятница.</p>
f_DateDiff	f_DateDiff (<datetime1>, <datetime2>, <string>)	<p>Возвращает число, количество полных интервалов между датами <datetime1> и <datetime2> (время игнорируется). Параметр <string> содержит параметры, разделенные через ",". Возможные параметры: "Y", «Year» – возвращается количество лет; "Q", «Quarter» – возвращается количество кварталов; "M", «Month» – возвращается количество месяцев; "W", «Week» – возвращается количество недель; "D", «Day» – возвращается количество календарных дней; "DW", «DayWork» – возвращается количество рабочих дней; "DO", «DayOff» – возвращается количество нерабочих дней; "DH", «DayHoliday» – возвращается количество праздничных дней; "P", «Plan» – использовать только производственный календарь (режим по умолчанию); "M", «Mix» – использовать производственный календарь, если существует; "B", «Base» – использовать только базовый календарь (сб, вс – выходные).</p>
f_DateType	f_DateType (<datetime>)	<p>Возвращает строку, тип дня: "W" – рабочий день; "O" – выходной; "H" – праздничный. Используется только производственный календарь.</p>
f_DateType2	f_DateType2(<datetime> , <string>)	<p>Возвращает строку, тип дня: "W" –</p>

		рабочий день; "О" – выходной; "Н" – праздничный. Параметр <string> содержит параметр: "P", «Plan» – использовать только производственный календарь; "М", «Mix» – использовать производственный календарь, если существует; "В", «Base» – использовать только базовый календарь (сб, вс – выходные).
Day	Day(<date>)	Получает день месяца даты, заданной аргументом
DayName	DayName(<date>)	Определяет день недели даты, заданной аргументом, и возвращает день недели в символьном представлении на английском языке (например, «Tuesday»)
DayNumber	DayNumber(<date>)	Возвращает номер дня недели даты, заданной аргументом. 1-воскресенье, 2-понедельник и т.д.
DaysAfter	DaysAfter(<date1> , <date2>)	Вычисляет количество дней, прошедших с даты <date1> до наступления даты <date2>.
f_DayName	f_DayName (datetime (<Date>))	Определяет день недели даты, заданной аргументом, и возвращает день недели в символьном представлении на русском языке
Hour	Hour(<time>)	Получает час из значения типа "Время". Используется 24 часовое представление.
IsDate	IsDate(<string>)	Проверяет, содержит ли строка корректное символьное представление даты. Например, IsDate('2002-08-09') = true
IsTime	IsTime(<timevalue>)	Проверяет, содержит ли строка корректное символьное представление времени. Например, IsTime('10:19:56') = true
Minute	Minute(<time>)	Получает количество минут из переменной типа "Время", передаваемой в качестве аргумента.
Month	Month(<date>)	Получает месяц (1-12) из даты.
f_MonthDayCount	f_MonthDayCount(<Month>, <Year>)	Функция возвращает значение типа Число. Данное значение вычисляется, как количество дней в месяце, указанном в параметрах <Month> и <Year>.
f_MonthLastDay	f_MonthLastDay(<datetime>)	Функция возвращает дату, в которой день вычисляется как последний день месяца, указанного в параметре. Так как на вход функция получает значение типа

		дата/время, то лучше использовать: f_MonthLastDay (datetime (<Date>, Now())).
f_MonthAfter	f_monthsafter (datetime (<Date1>), datetime (<Date2>))	Вычисляет количество полных месяцев, прошедших с даты Date1 до наступления даты Date2.
f_MonthWorkHours	f_MonthWorkHours(<datetime> , <string>)	Возвращает норму рабочих часов по производственному календарю на месяц и год из параметра <datetime>. Параметр <string> содержит параметр: "P", "Plan" -- использовать только производственный календарь; "M", "Mix" -- использовать производственный календарь, если существует; "B", "Base" -- использовать только базовый календарь (сб, вс -- выходные).
MonthName	Monthname (<месяц (число)>)	Возвращает имя месяца на русском языке
MonthName_skl	Monthname_skl (<месяц (число)>)	Возвращает имя месяца на русском языке для использования в строковом написании имени месяца в дате, например, января, марта...
Now	Now()	Возвращает текущее время.
RelativeDate	RelativeDate(<date> , <number>)	Возвращает дату, отстоящую от указанной в качестве параметра на заданное количество дней
gf_RelativeDate Time	gf_RelativeDateTime (<datetime>, <no_of_days>, <time>)	Функция возвращает значение типа дата/время. Данное значение вычисляется, как момент времени спустя <no_of_days> и <time>.
f_RelativeMonth	f_relativemonth (datetime (<Date>), <int_month>)	Возвращает дату, отстоящую от указанной в качестве параметра на заданное количество месяцев.
RelativeTime	RelativeTime(<time> , <number>)	Получает время, большее указанного на заданное количество секунд в пределах 24-часового периода.
Second	Second(<time>)	Возвращает количество секунд из переменной типа "Время"
SecondsAfter	SecondsAfter(<time1> , <time2>)	Вычисляет разницу в секундах между первым и вторым аргументами типа "Время".
gf_SecondsAfter Datetime	(<start>, <end>)	Функция возвращает количество секунд между двумя значениями типа «Дата/Время».
Time	Time(<string>)	Преобразует строку к типу "Время".
Today	Today()	Возвращает системную дату.
gf_WeekNumber	gf_WeekNumber (<datetime>)	Возвращает номер недели в году для аргумента типа "дата/время"

f_WorkCalExist	f_WorkCalExist(<Year>)	Возвращает Истину, если существует производственный календарь на указанный год.
Year	Year(<date>)	Возвращает год, заданный аргументом типа "Дата"
Группа функций «Карта работы»		
f_ButtonPost	f_ButtonPost (this , ' <BtnName> ')	Возвращает пустую строку. Имитирует нажатие, если допустимо, указанного управляющего объекта текущего окна после выполнения текущих действий. <BtnName> – строка, содержащая имя управляющего объекта. Может принимать значение: 'cb_Ok' – кнопка 'Выполнить'; 'cb_Cancel' – кнопка 'Отмена'; 'cb_PrintPreview' – 'Предварительный просмотр'; 'cb_Print' – 'Печать'; 'cb_PrintImmediate' – 'Печать без диалогов'. Только для окна задачи: 'cb_Apply' – кнопка 'Сохранить'; 'cb_InWork' – кнопка 'Принять'; 'cb_Refuse' – кнопка 'Отказ'.
f_CalEvt_GetID	f_CalEvt_GetID (<root_id>, <user_id>)	Возвращает строковый ID события для пользователя. Параметры: root_id (строка) – ID корневого события, user_id (number или пользователь) – ID пользователя, для которого производится поиск.
f_CalEvt_GetItem	f_CalEvt_GetItem (<user_event_id>, <col_name>)	Возвращает свойство события в строковом виде. Параметры: user_event_id (строка) – ID пользовательского события, col_name (строка) – имя поля события, содержащего искомое свойство.
f_CalEvt_Set	f_CalEvt_Set (<root_event_id>, <desc>, <note>, <when>, <begin>, <duration>, <users_arr>, <ini_user>)	Создает или изменяет событие календаря. Если передан пустой root_event_id (строка), создается новое событие и возвращается строка – ID созданного корневого события (root_event_id). В случае ошибки функция возвращает пустую строку. Если передан не пустой root_event_id, то происходит обновление событий. Параметры: desc (строка) – тема события, note (строка) – примечание к событию, when (дата/время)- дата начала (время в расчет не берется)

		<p>события, begin (время) – время начала события, duration (число) – продолжительность события в минутах, users_arg (строка) – список ID пользователей – участников событий, разделенных знаком «;», если параметр принимает значение «-», то событие удаляется из пользовательских календарей, ini_user (число или пользователь) – ID инициатора события.</p>
f_CalEvt_SetItem	f_CalEvt_SetItem (<user_or_root_event_id>, <col_name>, <value>, <process_children>)	<p>Изменяет значение свойства события. Возвращает ЧИСЛО: -1, если произошла ошибка, 1 – при успешном выполнении. Параметры: user_or_root_event_id (строка) – ID изменяемого корневого или пользовательского события, col_name (строка) – имя поля события, содержащего изменяемое свойство, value (строка) – новое значение свойства, process_children (число) – может принимать значение 0 или 1; если первый параметр функции (user_or_root_event_id) – является ID корневого события, то значение 0 означает, что будет изменяться только корневое событие, а значение – 1 означает, что будут меняться все связанные пользовательские события. Если первый параметр функции – ID пользовательского события, то параметр process_children игнорируется.</p>
f_ExpandGroups	f_ExpandGroups(<UserArrVariable> , <Exclude_Double_Users>)	<p>Возвращает МАССИВ ПОЛЬЗОВАТЕЛЕЙ. Разворачивает группы в переданном массиве пользователей и возвращает получившийся в результате данной операции МАССИВ. Второй аргумент-флаг, если > 0 – исключать повторяющихся пользователей из итогового массива.</p>
f_GetCycleCounter	f_GetCycleCounter(this)	<p>Возвращает счетчик (номер) текущего цикла циклического действия. Если действие не циклическое, то всегда равен 1. Переменная this – предопределенная системная</p>

		переменная типа строка. Содержит некоторые системные параметры текущего действия (упакованные специальным образом).
f_GetRootMapIDs	f_GetRootMapIDs (<IDsArray>)	Возвращает МАССИВ идентификаторов карт, являющихся корневыми по отношению к картам, переданным в качестве аргумента-массива (текущая карта может быть картой глубокой вложенности и данная функция единственный способ получить код ее самой верхней – корневой родительской карты).
f_ModifyWin	f_ModifyWin (this , <CtrlName> , <ModifyExp>)	Возвращает пустую СТРОКУ если ошибок нет и текст ошибки в противном случае. Применяет <ModifyExp> для изменения свойств объекта <CtrlName> текущего окна. <CtrlName> – строка, содержащая имя управляющего объекта. Может принимать значение: 'cb_All' – все кнопки, кроме 'Отмена'; 'cb_Ok' – кнопка 'Выполнить'; 'cb_Cancel' – кнопка 'Отмена'; 'cb_PrintPreview' – 'Предварительный просмотр'; 'cb_Print' – 'Печать'; 'cb_PrintImmediate' – 'Печать без диалогов'; 'Window' – окно. Только для окна задачи: 'cb_Apply' – кнопка 'Сохранить'; 'cb_InWork' – кнопка 'Принять'; 'cb_Refuse' – кнопка 'Отказ'; 'tb_Documents' – вкладка 'Документы'; 'tb_Message' – вкладка 'Сообщение'; 'tb_Audit' – вкладка 'Аудит'. <ModifyExp> – строка, содержащая выражение для изменения свойств. Для кнопок и вкладок может содержать значения: 'Enabled=1' – разрешить использование объекта; 'Enabled=0' – запретить использование объекта; 'Visible=0' – скрыть объект; 'Visible=1' – отобразить объект. Для окна может содержать значение: 'Title="Заголовок"' – сменить заголовок окна.
f_SetMailTaskStatus	f_SetMailTaskStatus (<Status>)	Возвращает пустую СТРОКУ. Изменяет параметры сообщения текущей задачи. Параметр <Status> – строка, содержащая описание

		<p>изменений. Может содержать: 'MarkRead' – для всех получателей сообщения устанавливается статус "Прочитано".</p> <p>Не влияет на статус сообщения у пользователя, вызывающего функцию.</p>
Группа функций «Массивы»		
f_ArrayToString	f_ArrayToString (<ArrVariable> , <Format>, <delimiter>, <SkipEmpty>)	Возвращает строку, сформированную из элементов массива <ArrVariable>, разделенных строкой <delimiter>. Строка <Format> может содержать формат, используемый для преобразования каждого элемента массива в строку. Если число <SkipEmpty> содержит нулевое значение, то пустые элементы не будут игнорироваться.
f_DifferenceOfArrs	f_DifferenceOfArrs (<ArrVariable_1> , <ArrVariable_2>)	Возвращает МАССИВ того же типа, что и первый МАССИВ-аргумент. Возвращает массив состоящий из элементов первого массива, которые отсутствуют среди элементов второго массива.
f_DistinctArr	f_DistinctArr(<ArrVariable>)	Возвращает МАССИВ того же типа, что и МАССИВ-аргумент. Исключает из МАССИВа-аргумента элементы с повторяющимися значениями и возвращает МАССИВ, получившийся в результате данной операции.
f_ExcludeArrEl	f_ExcludeArrEl(<ArrVariable> , <NUMArrVariable>)	Возвращает МАССИВ того же типа, что и первый МАССИВ-аргумент. Исключает из первого МАССИВа-аргумента элементы с индексами, находящимися в NUM-массиве индексов (2-ой аргумент) и возвращает получившийся в результате данной операции МАССИВ. Второй аргумент данной функции может быть: 1. Числовым МАССИВОМ. 2. Строкой вида '1,3,7,99' – список индексов исключаемых элементов. 3. Строкой 'ALL' – исключить ВСЕ индексы (то есть, попросту, очистить МАССИВ).
f_ExecAgregateEXP	f_ExecAgregateEXP (<ArrVariable>, 'Max (@ArrEL for All)')	Выполняет агрегатную обработку массива. Возвращает МАССИВ

		<p>одинаковых значений, тип данных которого зависит от возвращаемого вторым аргументом результата, а число элементов соответствует числу элементов массива первого аргумента функции. Первый аргумент – обрабатываемый массив. Второй аргумент – выражение в строковом виде, содержащее агрегатную функцию, применяемую к первому аргументу или возвращающее константу. Во втором аргументе можно использовать predetermined аргумент @ArrEL, означающий элементы массива. Если во втором аргументе используется имя переменной – оно должно быть указано вне кавычек. Например, если второй аргумент имеет вид: 'Max (@ArrEL for All)', то функция вернет массив максимальных значений массива-первого аргумента. Если второй аргумент имеет вид: 'Привет' или '~" + variable + "~", то функция вернет массив элементов со значением «Привет» или со значениями переменной variable соответственно.</p>
f_GetArrElInd	f_GetArrElInd(<ArrVariable> , '@ArrEL > 0')	<p>Возвращает числовой МАССИВ. Ищет в МАССИВе-аргументе элементы по строке-шаблону, переданной в качестве 2-го аргумента, и возвращает упорядоченный МАССИВ индексов найденных элементов. Примеры строки-шаблона: '@ArrEL > 0' – вернет массив индексов тех элементов числового массива, которые больше 0; '@ArrEL like ~"%xxx%"~' – вернет массив индексов тех элементов строкового массива, которые в своем тексте содержат подстроку 'xxx'. Если во втором аргументе используется имя переменной – оно должно быть указано вне кавычек. Примеры поиска по не строковым данным: f_GetArrElInd (dt, '@ArrEL = DateTime (~" + string(<дата/время>) + '~')')</p>

		f_GetArrElInd (dt, '@ArrEL = Number (~" + string(<число>) + '~')')
f_GetArrUpperBound	f_GetArrUpperBound(<ArrVariable>)	Возвращает верхнюю границу (размер) переданного в качестве аргумента массива.
f_GetDArrEl	f_GetDArrEl(<ArrVariable>, <Ind>)	Возвращает дату, соответствующую элементу под номером <Ind> массива дат <ArrVariable>.
f_GetDArrElInd	f_GetDArrElInd(<ArrVariable>, <FindedDate>, <Start>)	Возвращает число-индекс того элемента массива дат <ArrVariable>, значение которого совпадает со значением даты в искомой переменной или константе типа «дата/время». Если критерий поиска – переменная или константа типа «дата», следует преобразовать его в тип «дата/время» с помощью функции «DateTime». <Start> – индекс, с которого начинается поиск.
f_GetDTArrEl	f_GetDTArrEl(<ArrVariable>, <Ind>)	Возвращает дату и время соответствующие элементу под номером <Ind> массива дат и времен <ArrVariable>.
f_GetDTArrElInd	f_GetDTArrElInd(<ArrVariable>, <FindedDateTime>, <Start>)	Возвращает число-индекс того элемента массива дат и времен <ArrVariable>, значение которого в точности совпадает с искомой переменной или константой типа дата и время. <Start> – индекс, с которого начинается поиск.
f_GetNArrEl	f_GetNArrEl(<ArrVariable>, <Ind>)	Возвращает число, соответствующее элементу под номером <Ind> числового массива или массива пользователей <ArrVariable>.
f_GetNArrElInd	f_GetNArrElInd(<ArrVariable>, <FindedNumber>, <Start>)	Возвращает число-индекс того элемента числового массива или массива пользователей <ArrVariable>, значение которого в точности совпадает с искомой числовой переменной или константой. <Start> – индекс, с которого начинается поиск.
f_GetSArrEl	f_GetSArrEl(<ArrVariable>, <Ind>)	Возвращает строку, соответствующую элементу под номером <Ind> строкового массива <ArrVariable>.
f_GetSArrElInd	f_GetSArrElInd(<ArrVariable>, <FindedString>, <Start>)	Возвращает число-индекс того элемента строкового массива

		<ArrVariable>, значение которого в точности совпадает с искомой строковой переменной или константой. <Start> – индекс, с которого начинается поиск.
f_GetTArrEl	f_GetTArrEl(<ArrVariable>, <Ind>)	Возвращает время, соответствующее элементу под номером <Ind> массива времен <ArrVariable>.
f_GetTArrElInd	f_GetTArrElInd(<ArrVariable>, <FindedTime>, <Start>)	Возвращает число-индекс того элемента массива времен <ArrVariable>, значение которого в точности совпадает с искомой переменной или константой типа время. <Start> – индекс, с которого начинается поиск.
f_GetTemplDocDescription	f_GetTemplDocDescription (this , '<TemplDocName>')	Возвращает массив описаний для документов с указанным идентификатором.
f_GetTemplDocFiles	f_GetTemplDocFiles (this, <TemplDocName>)	Возвращает массив полных имен файлов, привязанных к документу, идентификатор которого передан в качестве аргумента функции.
f_GetTemplDocID	f_GetTemplDocID (this, '<TemplDocName>')	Возвращает строковый массив значений ID для документов с указанным идентификатором.
f_StringToArray	f_StringToArray (<string>, <delimiter>)	Возвращает строковый МАССИВ, элементы которого содержат части строки <string>, ограниченные разделителем <delimiter>.

Группа функций «Математические»

Abs	Abs(<number>)	Вычисляет модуль числа.
Ceiling	Ceiling(<number>)	Определяет наименьшее целое число, которое больше или равно числу, указанному в качестве аргумента
Cos	Cos(<angle>)	Вычисляет косинус угла.
Exp	Exp(x)	Возводит число e в указанную степень.
Fact	Fact(<number>)	Вычисляет факториал числа.
Int	Int(<number>)	Возвращает максимальное целое число, которое меньше или равно указанному в качестве аргумента числу.
Log	Log(<number>)	Вычисляет натуральный логарифм числа.
LogTen	LogTen(<number>)	Вычисляет десятичный логарифм числа.
Pi	Pi(<number>)	Умножает число Pi на указанное число.

Round	Round(<number> , <decimal places>)	Округляет число до указанного количества десятичных знаков.
Sign	Sign(<number>)	Определяет знак числа. Функция определяет, является ли число отрицательным, положительным или нулём.
Sin	Sin(<angle>)	Вычисляет синус угла.
Sqrt	Sqrt(<number>)	Вычисляет квадратный корень числа.
Tan	Tan(<angle>)	Вычисляет тангенс угла.
Truncate	Truncate(<number> , <decimal places>)	Обрезает число до указанного количества десятичных разрядов.
Группа функций «Преобразования»		
Asc	Asc(<string>)	Возвращает число, соответствующее ASCII коду первого символа строки.
Char	Char(<integer>)	Преобразует целое число в символ.
Integer	Integer(<string>)	Преобразует строку к целому числу в диапазоне от -32 768 до 32 767. Если преобразование невозможно, то возвращается 0. Например, Integer('123') = 123.
Long	Long(<string>)	Преобразует строку к целому числу в диапазоне от -2 147 483 648 до 2 147 483 647. Если преобразование невозможно, то возвращается 0. Например, Long('123') = 123.
Number	Number(<string>)	Преобразует строку к числу с плавающей точкой.
Real	Real(<string>)	Преобразует строковое значение в тип данных с плавающей точкой "Real"
String	String(<string> { , <format> })	Преобразует значение в строку в соответствии с форматом. вы можете преобразовать в строку значения типов "Дата", "Время", "Дата/время", "Число", а также отформатировать значение типа «Строка».
Группа функций «Работа с БД»		
f_CheckUserRights	f_CheckUserRights (<RightID> , <ObjID> , <Grants> , <Live>)	Проверяет права текущего пользователя. Возвращает истину, если текущему пользователю даны запрошенные права. <RightID> – ID запрашиваемых прав. Значения RightID: <u>Объект:</u> 2004000 Доступ к объекту 2004001 Доступ к атрибутам

		<p>объекта 2004003 Доступ к проекту 2004004 Создание исполнения 2004005 Правка ЭЦП <u>Документ:</u> 2000000 Правка версий документа <u>Версия документа:</u> 2001000 Правка версии документа <u>Атрибут:</u> 2002000 Правка атрибута у экземпляра объекта 2002001 Правка списка значений атрибута <u>Тип объекта:</u> 2005000 Правка экземпляра объекта данного типа; <ObjID> – ID объекта защиты, если необходимо; <Grants> – строка, содержащая четыре символа, описывающие требуемые разрешения в формате «ДСИУ»: Д – доступ. С – создание. И – изменение. У – удаление. На месте символов «Д», «С», «И» и «У» должны быть служебные латинские символы в верхнем регистре, соответствующие варианту доступа к данному праву: Е – разрешено. «пробел» – без изменений или отсутствие права. G – право передачи. <Live> – указывает сколько секунд хранить в кэше результаты функции. Функция используется для форм.</p>
f_DbProfileString	f_DbProfileString (<section>, <key>, <default>)	<p>Возвращает строку со значением настройки, хранящейся в БД, для текущего пользователя. Аргументы <section> и <key> определяют настройку. Значение <default> возвращается в том случае, если ни в настройке профиля, ни в настройке пользователя нет секции <section> (настройки ни разу не сохранялись). Результат кэшируется. При изменении настроек пользователем (выполненном в текущей копии</p>

		<p>приложения) кэш значений сбрасывается. Например, для получения пути к пользовательским пиктограммам используйте:</p> <p>f_DbProfileString ('PInterfaceAttribute', 'IconPath', ").</p> <p>Для получения других настроек обратитесь за значениями аргументов функции в службу технической поддержки.</p>
<p>f_ExecSQLSelect (см. Примечание)</p> <p>Не рекомендуется использовать. Оставлена для совместимости со старыми версиями Lotsia PDM PLUS. Рекомендуется использовать функцию f_ExecSQLSelect_3</p>	<p>f_ExecSQLSelect (<SQLSelect_Statement> , <FieldsDELIM> , <RowsDELIM>)</p>	<p>Возвращает СТРОКУ. Выполняет в текущей Базе Данных SQL select-инструкцию <SQLSelect_Statement>.</p> <p>Возвращает набор необработанных строк (в виде одной строки), составленный из приведенных к типу string значений колонок Select-a, соединенных посредством разделителя <FieldsDELIM>.</p> <p>Строки Select-a соединяются посредством разделителя <RowsDELIM>. Если результат используется в функции формирования выпадающего списка значений переменной формы, то она должна возвращать два поля – описание значения и само значение. Значения должны разделяться табулятором (char(9) или '~t'), а строки – символом '/'.</p>
<p>f_ExecSQLSelect_2 (см. Примечание)</p> <p>Не рекомендуется использовать. Оставлена для совместимости со старыми версиями Lotsia PDM PLUS. Рекомендуется использовать функцию f_ExecSQLSelect_3</p>	<p>f_ExecSQLSelect_2 (<SQLSelect_Statement> , <FieldsDELIM> , <RowsDELIM>)</p>	<p>Возвращает СТРОКУ. Выполняет в текущей Базе Данных SQL select-инструкцию <SQLSelect_Statement>.</p> <p>Возвращает набор строк, обработанных специальным образом, (в виде одной строки), составленный из приведенных к типу string значений колонок Select-a, соединенных посредством разделителя <FieldsDELIM>.</p> <p>Строки Select-a соединяются посредством разделителя <RowsDELIM>. Если результат используется в функции формирования выпадающего списка значений переменной формы, то она должна возвращать два поля – описание значения и само значение. Значения должны</p>

		разделяться табулятором (char(9) или '~t'), а строки – символом '/'.
f_ExecSQLSelect_3 (см. Примечание)	f_ExecSQLSelect_3 (<SQLSelect_Statement> , <FieldsDELIM> , <RowsDELIM>, <Flags>, <Live>)	<p>Возвращает СТРОКУ. Выполняет в текущей БД запрос в виде команды SELECT или вызова процедуры, возвращающей результат. Текст SQL-select'a может быть скомпонован из переменных и констант, но при использовании функции в формах, команда SELECT должна быть указана явно. Например, 'select ' + <переменная, содержащая остальную часть запроса>.</p> <p>В случае использования процедуры, ожидается, что она возвращает один результат (resultset). Возвращает набор строк (в виде одной строки), составленный из приведенных к строковому виду значений колонок результата, соединенных посредством разделителя <FieldsDELIM>. Строки результата соединяются посредством разделителя <RowsDELIM>. Если результат используется в функции формирования выпадающего списка значений переменной формы, то значения должны разделяться табулятором (char(9) или '~t'), а строки – символом '/'. Параметр <Flags> может содержать дополнительные параметры, перечисленные через символ ";".</p> <p>Возможные параметры: "ddlb" – возвращаемая строка будет подготовлена для использования ее в выпадающем списке; "cols=type1,type2,...,typen" – указывает типы колонок результата. Допустимые типы: char(n), date, datetime, decimal(n), long, number, time. Для преобразования значений типов datetime, date и time в строку используется формат «DateTimeFormat», «DateFormat» и «TimeFormat» соответственно. Параметр <Flags> необходимо использовать при вызове процедуры, а дополнительный</p>

		<p>параметр cols необходимо использовать всегда. Параметр <Live> – число, количество секунд хранения результата функции в буфере (кеше). Буфер необходим для уменьшения числа обращения к БД. Результаты хранятся в буфере отдельно для каждого сочетания аргументов функции. Если при очередном вызове функции в буфере найден соответствующий результат, то он будет использован вместо обращения к БД. Буфер очищается перед выполнением действия над переменными или объектами, обновлением формы или выполнением отчета. Если Live содержит 0, то работа с буфером не используется. Если Live меньше 0, то значение в буфере хранится до конца сессии.</p>
f_GetCurrentProfileId()	f_GetCurrentProfileId()	Используется только в вычисляемых полях. Возвращает код (число) профиля текущего пользователя.
f_GetCurrentUserID()	f_GetCurrentUserID()	Возвращает идентификатор текущего пользователя.
f_GetGroupID	f_GetGroupID(<group_name>)	Возвращает идентификатор группы по её имени.
f_GetNextSequence	f_GetNextSequence(<Name>, <Description>)	<p>Возвращает следующее значение пользовательского счетчика <Name>. Если пользовательского счетчика <Name> не существует, он будет создан с описанием <Description> и вернет 1. В случае ошибки возвращает NULL.</p> <p>Внимание: не используйте данную функцию в вычисляемых полях форм!</p>
f_GetObjectName	f_GetObjectName (<ObjectID>)	Возвращает название объекта по его идентификатору.
gf_GetUserGroups	gf_GetUserGroups(<user_id>)	Функция возвращает список идентификаторов групп, в которые входит пользователь.
f_GetUserID	f_GetUserID(<user_name>)	Функция возвращает идентификатор пользователя по его имени.
gf_IsUserInGroup	gf_IsUserInGroup(<user_id>, <group_id>)	Функция проверяет, входит ли указанный пользователь в указанную группу. Возвращает значение истина или ложь.
f_GroupName	f_GroupName(<group_id>)	Функция возвращает имя группы

		по её идентификатору.
f_MailPartGet	f_MailPartGet (<MsgID>, <Part>)	<p>Функция возвращает в строковом виде указанный параметр сообщения (Part) с указанным ID (MsgID). ID сообщения можно получить из свойства «ID исходного сообщения» объекта «Работа».</p> <p>Существенные параметры внутренних входящих сообщений:</p> <p>Subject – тема.</p> <p>Created – дата и время создания сообщения в БД.</p> <p>priority – приоритет.</p> <p>recipients – имена получателей.</p> <p>user_to_ids – коды пользователей-получателей через запятую.</p> <p>Files – список прикрепленных файлов, разделенных табулятором («~t»).</p> <p>msg_text – текст сообщения.</p> <p>user_from_id – код пользователя-отправителя сообщения. Если код отправителя равен 1, сообщение внешнее (email), в этом случае следует использовать параметры с именами, начинающимися с «@@».</p> <p>Имена эти параметров можно посмотреть, если указать пустую строку вместо параметра сообщения – будет возвращена строка со списком имен всех параметров сообщения и их значения. Полученный список имен параметров можно использовать как справочник параметров.</p>
f_UserName	f_UserName(<user_id>)	Функция возвращает имя пользователя по его идентификатору.
Группа функций «Разные»		
Case	Case (<column> When <condition a> Then <value b> ... Else <value c>)	Определяет значение из списка на основании условных выражений.
f_GetComputerName	f_GetComputerName ()	Возвращает полное имя компьютера.
f_GetComputerNameEx	f_GetComputerNameEx (<NameType>)	<p>Возвращает имя компьютера согласно числовому параметру NameType:</p> <p>0 – NetBIOS-имя локального компьютера. Если локальный</p>

		<p>компьютер является узлом в кластере, возвращается NetBIOS-имя сервера виртуального кластера.</p> <p>1 – Имя хоста DNS локального компьютера. Если локальный компьютер является узлом в кластере, возвращается имя хоста DNS сервера виртуального кластера.</p> <p>2 – Имя домена DNS, присвоенного локальному компьютеру. Если локальный компьютер является узлом в кластере, возвращается имя домена DNS сервера виртуального кластера.</p> <p>3 – Полное имя DNS, которое уникально идентифицирует данный компьютер. Это имя является комбинацией имени хоста DNS и имени домена DNS, использующей форму HostName.DomainName. Если локальный компьютер является узлом в кластере, возвращается Полное имя DNS сервера виртуального кластера.</p> <p>4 – NetBIOS-имя локального компьютера. Если локальный компьютер является узлом в кластере, возвращается NetBIOS-имя локального компьютера, а не сервера виртуального кластера.</p> <p>5 – Имя хоста DNS локального компьютера. Если локальный компьютер является узлом в кластере, возвращается имя хоста DNS локального компьютера, а не сервера виртуального кластера.</p> <p>6 – Имя домена DNS, присвоенного локальному компьютеру. Если локальный компьютер является узлом в кластере, возвращается имя домена DNS локального компьютера, а не сервера виртуального кластера.</p> <p>7 – Полное имя DNS, которое уникально идентифицирует данный компьютер. Если локальный компьютер является узлом в кластере, возвращается полное имя DNS локального компьютера, а не сервера виртуального кластера.</p>
f_GetFileInfo	f_GetFileInfo (<srcfile >, <value>)	Возвращает информацию о файле

		<srcfile>. <value> может быть "SIZE" – размер в байтах, "CREATION" – дата создания, "WRITE" – дата изменения.
f_GetImageMetadata	f_GetImageMetadata (<srcfile> , <metaname>)	Возвращает строковое значение поля метаданных файла изображения <srcfile>. Список полей можно увидеть, передав в <metaname> пустую строку. При импорте документов и сканировании в выражениях можно использовать те же поля в виде макропеременных %EXIF_WIDTH%, %EXIF_MODEL%, и т.п.
f_GetImageThumbnail	f_GetImageThumbnail (<srcfile> , <width> , <height> , <backcolor> , <templ>)	Создает эскиз к <srcfile> с заданными параметрами и возвращает путь к нему. Параметры эскиза: <width> и <height> – соответственно ширина и высота эскиза в пикселях. <backcolor> – RGB цвета фона. В <templ> можно указать имя временного файла (создается в папке %TEMP%) или полный путь к файлу эскиза. Функция также кеширует метаданные, которые потом можно получить через функцию f_GetImageMetadata.
f_GetPathPart	f_GetPathPart (<path> , <part>)	Возвращает часть <part> из полного пути к файлу <path>. Параметр <part> может принимать значения: «F» – «file», имя файла с расширением; «P» – «path», путь без файла, включая «\» в конце; «N» – «name», имя файла без расширения; «E» – «ext», расширение файла, включая точку в начале.
f_GetVerPathName	f_GetVerPathName (<doc/file>)	Функция временно открывает на просмотр версию документа архива – изображение и возвращает путь к нему. В форме изображение можно отобразить функцией Bitmap (f_GetVerPathName (<doc/file>)). <doc/file> – число, код версии документа архива.
IsNull	IsNull(<expression> or <column>)	Проверяет, является ли значение NULL-значением.
IsNumber	IsNumber(<string>)	Проверяет, содержит ли строка символьное представление числа.

		Например, IsNumber('2192') = true
f_IsWeb	f_IsWeb ()	Возвращает True, если приложение работает на WEB-сервере
f_IsXpStyle	f_IsXpStyle ()	Возвращает True, если приложение работает в режиме оформления XP Style.
f_MediaPlay	f_MediaPlay («sound_file» , «flags»)	<p>Начинает воспроизведение мультимедиа «sound_file». «flags» – параметр воспроизведения, может быть пустым или иметь значение «wait». Значение «wait» означает, что управление в программу не вернется до окончания воспроизведения. Если параметр не указан, управление возвращается сразу, то есть файл будет проигрываться, а Lotsia PDM PLUS будет доступна для дальнейшей работы.</p> <p>Возвращает число: «1» – ошибки отсутствуют, «-1» – операция не удалась.</p> <p>Пример: f_MediaPlay («NewMail.wav» , «wait»)</p> <p>В действиях над объектами функция используется через функцию Set, например: <числовая_переменная> = Set (f_MediaPlay («NewMail.wav» , «»))</p>
f_MediaStop	f_MediaStop («sound_file»)	<p>Останавливает воспроизведение мультимедиа «sound_file», если при его воспроизведении не был указан параметр «wait».</p> <p>Возвращает число: «1» – ошибки отсутствуют, «-1» – операция не удалась.</p> <p>Пример: f_MediaStop («NewMail.wav»)</p> <p>В действиях над объектами функция используется через функцию Set, например: <числовая_переменная> = Set (f_MediaStop («NewMail.wav»))</p>
f_OpenFileOrUrl	f_OpenFileOrUrl (<file_or_url>)	<p>Функция позволяет открыть документ, файл или URL во встроенном окне просмотра.</p> <p>Возвращает число, 1 в случае успеха, 0 в случае ошибки.</p>

		Функцию имеет смысл использовать в действиях.
f_OpenFileOrUrl2	f_OpenFileOrUrl2 (<file_or_url>, <options>)	Функция позволяет открыть документ, файл или URL во встроенном окне просмотра. Если <options> = «D», то окно просмотра можно использовать во время работы действий над объектами. Возвращает число, 1 в случае успеха, 0 в случае ошибки. Функцию имеет смысл использовать в действиях.
ProfileInt	ProfileInt(<filename> , <section> , <key> , <default>)	Получает целое значение из указанной секции указанного INI файла.
ProfileString	ProfileString(<filename> , <section> , <key> , <default>)	Получает строковое значение из указанной секции указанного INI файла.
Rand	Rand(<number>)	Возвращает случайное целое число от 1 до указанного верхнего предела.
f_RegistryGetString	f_RegistryGetString (<key> , <value> , <default>)	Возвращает строковое значение с именем Value ключа Key из системного реестра. Key – строковое полное имя ключа реестра. Value – строковое имя значения ключа. Если указанное значение не найдено, то возвращается значение параметра Default.
RGB	RGB (<redvalue>, <greenvalue>, <bluevalue>)	Возвращает номер цвета по 3 составляющим (Красный/Зелёный/Синий (RGB)).
Группа функций «Статистические»		
Avg	Avg(<column> for All)	Вычисляет среднее значение ячеек указанной колонки для каждой строки заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Count	Count(<column> for All)	Определяет общее количество строк указанной колонки для каждой строки заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
CrossTabAvg	CrossTabAvg(1)	Вычисляет среднее для значений, возвращаемых выражением в списке значений отчёта стиля crosstab.
CrossTabCount	CrossTabCount(1)	Вычисляет количество значений,

		возвращаемых выражением в списке значений отчёта стиля crosstab.
CrossTabMax	CrossTabMax(1)	Вычисляет максимальное значение, возвращаемое выражением в списке значений отчёта стиля crosstab.
CrossTabMin	CrossTabMin(1)	Вычисляет минимальное значение, возвращаемое выражением в списке значений отчёта стиля crosstab.
CrossTabSum	CrossTabSum(1)	Вычисляет сумму значений, возвращаемых выражением в списке значений отчёта стиля crosstab.
CumulativePercent	CumulativePercent(<column> for All)	Вычисляет процентное отношение количества строк колонки до текущей строки включительно к общему количеству строк колонки в заданной области (бегущий процент): All – для всех строк, Group N – для строк области группы N, где N – номер группы.
CumulativeSum	CumulativeSum(<column> for All)	Вычисляет сумму значений ячеек колонки до текущей строки включительно в заданной области (бегущая сумма): All – для всех строк, Group N – для строк области группы N, где N – номер группы.
First	First(<column> for All)	Возвращает значение первой строки указанной колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Large	Large(<column> , <column> , <rank> for All)	Находит наибольшее значение колонки на указанном уровне (например, второе наибольшее по значению или третье наибольшее по значению) и возвращает значение другой колонки или выражения, основанного на результате.
Last	Last(<column> for All)	Определяет значение последней строки указанной колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Max	Max(<column> for All)	Определяет максимальное значение

		ячеек указанной колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Median	Median(<column> for All)	Вычисляет медиану значений ячеек колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы. Медиана – среднее значение в наборе значений, такое, что в наборе есть одинаковое количество больших и меньших величин, чем данная величина.
Min	Min(<column> for All)	Определяет минимальное значение ячеек указанной колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Mod	Mod(<number> , <divisor>)	Вычисляет остаток от деления первого аргумента на второй.
Mode	Mode(<column> for All)	Вычисляет моду значений колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы. Мода – наиболее часто встречающаяся величина.
Percent	Percent(<column> for All)	Определяет процентное отношение текущего значения колонки к сумме всех значений для колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Small	Small(<column> , <column> , <rank> for All)	Находит наименьшее значение колонки указанного уровня (например, второе наименьшее по значению или третье наименьшее по значению) и возвращает значение другой колонки или выражения, основанного на результате.
StDev	StDev(<column> for All)	Вычисляет оценку стандартной девиации для значений указанной колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы. Стандартная девиация – это мера

		разброса значений величин из набора относительно среднего значения.
StDevP	StDevP(<column> for All)	Вычисляет стандартную девиацию для значений указанной колонки в заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы. Стандартная девиация – это мера разброса значений величин из набора относительно среднего значения.
f_StringListEx (см. Примечание)	f_StringListEx (Describe ('DataWindow.Syntax'), Describe('DataWindow.Syntax.Data') , <ColName>, <Separator>)	Функция возвращает строку, сцепленную из данных колонки <ColName> в текстовом виде по всем строкам отчета. Данные сцепляются через <Separator>, указанный в кавычках. <ColName> – имя соответствующей колонки формы, указанное в кавычках. Остальные параметры не должны изменяться. Использование функции имеет смысл только в формах стиля "Сетка" и "Таблица". Функция может обрабатывать вычисляемые поля и учитывает текущий фильтр формы.
Sum	Sum(<column> for All)	Вычисляет сумму значений ячеек указанной колонки для каждой строки заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Var	Var(<column> for All)	Вычисляет оценку дисперсии для значений ячеек указанной колонки для каждой строки заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
VarP	VarP(<column> for All)	Вычисляет дисперсию для значений ячеек указанной колонки для каждой строки заданной области: All – для всех строк, Group N – для строк области группы N, где N – номер группы.
Группа функций «Строковые»		
Fill	Fill(<string> , <number>)	Создаёт строку, указанного размера, в которой повторяются

		переданная в качестве параметра строка. Например, fill('abc',7) = 'abcabca'.
f_GlobalReplace	f_GlobalReplace (source, old, new, ignorecase)	Производит замену в строке «source» всех вхождений «old» на «new». Параметр «ignorecase» может принимать значения 0 или 1 (число). Значение 0 включает регистрозависимость поиска строки «old»
LastPos	LastPos (<string1> , <string2> { , <searchlength> })	Ищет с учетом регистра последнее вхождение строки string2 в строке string1 в первых searchlength символах и возвращает номер последней позиции строки string2 в указанном поисковом фрагменте строки string1 или ноль, если string2 не найдена.
Left	Left(<string> , <number>)	Выделяет указанное количество начальных символов строки.
LeftTrim	LeftTrim(<string>)	Удаляет начальные пробелы строки.
Len	Len(<string>)	Возвращает длину строки.
Lower	Lower(<string>)	Преобразует все символы строки в нижний регистр.
Match	Match(<string> , <textpattern>)	Проверяет, соответствует ли строка определённой маске.
Mid	Mid(<string> , <start>{ , <length> })	Получает из строки подстроку заданной длины, начиная с указанной позиции.
Pos	Pos(<string1> , <string2> { , <start> })	Ищет строку string2 в строке string1 с учетом регистра, начиная с позиции start и возвращает номер первой позиции строки string2 в строке string1 или ноль, если string2 не найдена. Например, для того, чтобы найти номер позиции второго, встречающегося в string1 знака косой черты ('/') можно использовать выражение: Pos (string1, '/', pos (string1, '/')+1).
f_Propis_Ex	f_Propis_Ex (<Sum> , <int_name> , <fract_name>)	Возвращает сумму прописью. <Sum> – описываемое число. Допускается не более двух разрядов дробной части. <int_name> – имя целой части, может принимать следующие специальные значения: 'NONE' – не отображать дробную часть и имя целой части; " (пустые кавычки) – вернуть сумму прописью в рублях и копейках (рубли склоняются).

		<fract_name> – наименование дробной части. Используется только если задано имя целой части.
f_RemoveEscape	f_RemoveEscape (<string>)	Возвращает строку string с транслированными escape-последовательностями.
Replace	Replace(<string1> , <start> , <length> , <string2>)	Заменяет подстроку строки другой подстрокой.
Reverse	Reverse (<string>)	Реверсирует порядок следования символов в строке.
Right	Right(<string> , <number>)	Выделяет указанное количество концевых символов строки.
RightTrim	RightTrim(<string>)	Удаляет завершающие пробелы строки.
f_SetEscape	f_SetEscape(<StringArg>)	Возвращает строку-аргумент, преобразованную для корректного использования в функции f_ModifyForm (обработаны кавычки и т.д.). Пример: f_ModifyForm (this, 'ls_titles', 'ls_titles.values = ~" + f_SetEscape ('a~'1~'~t1/a~'2~'~t2/a~'3~'~t3/') + '~"). Здесь, для колонки ls_titles, содержащей выпадающий список, формы массива ls_titles изменяются значения выпадающего списка.
Space	Space(<number>)	Создаёт строку из указанного количества пробелов.
Trim	Trim(<string>)	Удаляет ведущие и завершающие пробелы в строке.
Upper	Upper(<string>)	Преобразует строку к верхнему регистру.
WordCap	WordCap(<string>)	Преобразует каждое слово строки так, чтобы первая его буква была заглавной, а остальные строчными.
Группа функций «Формы»		
Bitmap	Bitmap (<bitmap filename>)	Отображает указанный файл изображения одного из следующих форматов: BMP, GIF, JPEG, RLE или WMF.
f_BitmapPath	f_BitmapPath (<bitmap filename>)	Возвращает полное имя файла изображения, путем добавления к пути, указанному в настройке пользователя "Путь к пользовательским пиктограммам" содержимого <bitmap filename>. Если <bitmap filename> содержит полный путь или имя встроенного изображения, то оно возвращается без изменения. Используется

		совместно с функциями Bitmap, f_GetBitmapHeight и f_GetBitmapWidth. Например: Bitmap (f_BitmapPath(<bitmap filename>))
CurrentRow	CurrentRow ()	Определяет номер текущей строки.
Describe	Describe(<string>)	Возвращает значения атрибутов формы и объектов формы .
f_GetBitmapHeight	f_GetBitmapHeight (<bitmap filename> , <live>)	Возвращает высоту изображения из указанного файла. Значение хранится во внутреннем кеше на протяжении <live> секунд. Если live меньше 0, то значение в кеше хранится до конца сессии. Функция используется в формах в выражениях высоты для вычисляемых полей, содержащих изображение, формируемое функцией Bitmap.
f_GetBitmapWidth	f_GetBitmapWidth (<bitmap filename> , <live>)	Возвращает ширину изображения из указанного файла. Значение хранится во внутреннем кеше на протяжении <live> секунд. Если live меньше 0, то значение в кеше хранится до конца сессии. Функция используется в формах в выражениях ширины для вычисляемых полей, содержащих изображение, формируемое функцией Bitmap.
f_GetObjPreview	f_GetObjPreview (<id>)	Используется только в вычисляемых полях. Временно открывает на просмотр версию документа архива-изображение, и возвращает путь к нему. Изображение можно отобразить в форме функцией Bitmap (f_GetObjPreview (<id>)). <id> – число, код объекта. У объекта должен быть атрибут, со значением, хранящим код версии отображаемого документа-изображения. Атрибут задается в Общих параметрах БД на вкладке «Атрибуты» в поле «Код версии файла предпросмотра».
GetRow	GetRow()	Возвращает порядковый номер строки.
GetText	GetText()	Получает текст, введенный в колонке.
IsRowModified	IsRowModified()	Возвращает true, если строка была изменена.

IsRowNew	IsRowNew()	Проверяет, была ли строка добавлена в форму.
IsSelected	IsSelected()	Проверяет, выделена ли строка.
LookUpDisplay	LookupDisplay(<column>)	Возвращает значение для отображения, соответствующее текущему значению указанной колонки.
Page	Page()	Определяет номер текущей страницы.
PageAcross	PageAcross()	Определяет номер текущей страницы по горизонтали.
PageCount	PageCount()	Возвращает общее количество страниц.
PageCountAcross	PageCountAcross()	Возвращает общее количество страниц по горизонтали, когда форма шире, чем окно предварительного просмотра.
RowCount	RowCount()	Возвращает количество строк, доступных в данный момент в форме.
RowHeight	RowHeight()	Возвращает высоту строки.
Группа функций «Формы задачи»		
f_DescribeForm	f_DescribeForm (this , <FormVarName> , <DescribeExp>)	Возвращает СТРОКУ. Применяет <DescribeExp> к форме <FormVarName> . <FormVarName> – имя переменной (указанное в кавычках), которая содержит имя анализируемой формы. Если <FormVarName> принимает значение пустой строки ("), то <DescribeExp> будет применена к форме задачи, исключая формы внедренных массивов. Если <FormVarName> принимает значение имени внедренной переменной-массива, то <DescribeExp> будет применена к форме этого внедренного массива. <DescribeExp> указывается в кавычках. Функция возвращает строку – описание или указанное свойство указанного объекта формы. Переменная this – предопределенная системная переменная типа строка. Содержит некоторые системные параметры текущего действия (упакованные специальным образом).
f_EmbeddedSQLSelect (см. Примечание)	f_EmbeddedSQLSelect (this , <SQLSelect_Statement> , <ArgsList> , <FormVarName>)	Внедряет в форму SQL Select и возвращает пустую СТРОКУ, если ошибок нет и текст ошибки в

		<p>противном случае. Количество, типы и последовательность колонок в форме и запросе должны совпадать. Запрос может содержать аргументы. Аргументы запроса должны быть описаны в строке <ArgsList>, например: 'arg1,string;arg2,datalist'. Типы аргументов запроса: 'datetime','date','string','number','datetimestring','datalist','stringlist','numberlist'. Аргументы, заканчивающиеся на "list" являются массивами. Последний аргумент функции – строковое имя переменной, содержащей форму для внедрения запроса.</p>
f_FilterForm	f_FilterForm (this , <FormVarName> , <FilterStatement>)	<p>Возвращает пустую СТРОКУ, если ошибок нет и текст ошибки в противном случае. Фильтрует строки формы массива <FormVarName> в соответствии с инструкцией <FilterStatement>. Выполняется на уровне формы и не затрагивает значения переменных. При необходимости обновления значений переменных используйте функцию f_FormDataToArray. Пример инструкции <FilterStatement>: 'Переменная_1 > 0'. Переменная_1 – имя переменной формы.</p>
f_FormArrayToData	f_FormArrayToData (this , '<FormVarName>')	<p>Возвращает пустую СТРОКУ. Заполняет форму данными из массивов, то есть принудительно обновляет форму массива.</p>
f_FormDataToArray	f_FormDataToArray (this , '<FormVarName>')	<p>Возвращает пустую СТРОКУ, если ошибок нет и текст ошибки в противном случае. Заполняет массивы данными из формы.</p>
f_GetCurrentRow	f_GetCurrentRow (this)	<p>Возвращает номер текущей выделенной строки (или первой из выделенных, если допускается выделение нескольких строк) для текущего действия по событию формы. Например, если выражение по изменению значения переменной включено в действие по событию 'после вставки строки в форму', то данная функция вернет номер последней вставленной строки.</p>

f_GetDFormItem	f_GetDFormItem (this, <FormVarName>, <Row>, <Column>)	Возвращает значение даты, находящееся в строке с номером <Row> колонки-переменной <Column> формы <FormVarName>. Может возвращать значение не только из формы массива, но и из формы задачи. В этом случае, параметр FormVarName = ‘’, а параметр Row = 1.
f_GetDTFormItem	f_GetDTFormItem (this, <FormVarName>, <Row>, <Column>)	Возвращает значение даты и времени, находящееся в строке с номером <Row> колонки-переменной <Column> формы <FormVarName>. Может возвращать значение не только из формы массива, но и из формы задачи. В этом случае, параметр FormVarName = ‘’, а параметр Row = 1.
f_GetNFormItem	f_GetNFormItem (this, <FormVarName>, <Row>, <Column>)	Возвращает числовое значение, находящееся в строке с номером <Row> колонки-переменной <Column> формы <FormVarName>. Может возвращать значение не только из формы массива, но и из формы задачи. В этом случае, параметр FormVarName = ‘’, а параметр Row = 1.
f_GetSelRowsForm	f_GetSelRowsForm (this , <FormVarName>)	Возвращает массив номеров выделенных строк указанной формы.
f_GetSFormItem	f_GetSFormItem (this , <FormVarName> , <Row> , <Column>)	Возвращает строковое значение, находящееся в строке с номером <Row> колонки-переменной <Column> формы <FormVarName>. Может возвращать значение не только из формы массива, но и из формы задачи. В этом случае, параметр FormVarName = ‘’, а параметр Row = 1.
f_GetTFormItem	f_GetTFormItem (this, <FormVarName>, <Row>, <Column>)	Возвращает значение времени, находящееся в строке с номером <Row> колонки-переменной <Column> формы <FormVarName>. Может возвращать значение не только из формы массива, но и из формы задачи. В этом случае, параметр FormVarName = ‘’, а параметр Row = 1.
f_GroupCalcForm	f_GroupCalcForm (this , <FormVarName>)	Пересчитывает группы указанной формы. Если в форме не задано ни одной группы не делает ничего.

		Возвращает пустую строку если ошибок нет и текст ошибки в противном случае.
f_ModifyForm	f_ModifyForm (this , <FormVarName> , <ModifyString>)	Возвращает пустую СТРОКУ, если ошибок нет и текст ошибки в противном случае. Применяет <ModifyString> к форме <FormVarName>. <FormVarName> – имя переменной (указанное в кавычках), которая содержит изменяемую форму. Если <FormVarName> принимает значение пустой строки (""), то <ModifyString> будет применена к форме задачи, исключая формы внедренных массивов. Если <FormVarName> принимает значение имени внедренной переменной-массива, то <ModifyString> будет применена к форме этого внедренного массива. <ModifyString> – modify-инструкция в виде строки, то есть в кавычках. Если modify-инструкция должна применяться к нескольким свойствам и/или нескольким объектам формы, то изменяемые свойства разделяются пробелом. Переменная this – предопределенная системная переменная типа строка. Содержит некоторые системные параметры текущего действия (упакованные специальным образом).
f_Open_FreeForm	f_Open_FreeForm (this , <FreeFormName>)	Возвращает ЧИСЛО. Открывает в диалоговом окне форму массива с именем, переданным в качестве второго строкового аргумента. Если открыто диалоговое окно с формой массива, то следующая форма массива открывается в <u>этом же окне вместо текущей</u> . Возвращает: -1 – ошибка. 0 – пользователь нажал «Отмена» (закрыл окно), то есть отменил редактирование переменных формы. 1 – пользователь нажал «Ок» (закрыл окно), то есть подтвердил редактирование переменных формы. 2 – форма открылась на месте ранее существующей (в том же окне).

f_RetrieveForm	f_RetrieveForm (this , <VarArgsList> , <FormVarName>)	Извлекает данные из Базы Данных в соответствии с внедренным в форму запросом и возвращает число извлеченных строк или -1 в случае ошибки. Аргумент функции <VarArgsList> содержит строковый список переменных карты, содержащих значения аргументов запроса, разделенных запятой, например, 'arg1,arg2,arg3'. Порядок следования аргументов должен соответствовать порядку перечисления аргументов в запросе, внедренном в форму массива или в соответствующем параметре функции f_EmbeddedSQLSelect
f_SetSelRowsForm	f_SetSelRowsForm (this , <FormVarName> , <RowsNumArray>)	Возвращает пустую СТРОКУ, если ошибок нет и текст ошибки в противном случае. Выделяет в форме массива те строки, номера которых переданы в качестве аргумента-массива RowsNumArray. RowsNumArray может также принимать значение 'All' – выделить все строки. Для формы массива должна быть задана возможность выделения строк.
f_SortForm	f_SortForm (this , <FormVarName> , <SortStatement>)	Возвращает пустую СТРОКУ, если ошибок нет и текст ошибки в противном случае. Сортирует строки формы массива <FormVarName> в соответствии с инструкцией <SortStatement>. Выполняется на уровне формы и не затрагивает значения переменных. При необходимости обновления значений переменных используйте функцию f_FormDataToArray. Пример инструкции <SortStatement>: 'Переменная_1 A, Переменная_2 D'. Переменная_1 и Переменная_2 – имена переменных формы, A – сортировка по возрастанию, D – сортировка по убыванию.

Примечание:

1.

Синтаксис 1:

*f_StringListEx (Describe('DataWindow.Syntax'),
Describe('DataWindow.Syntax.Data'), <ColName> , <Separator>)*

Общий вид вызова функции. Рекомендуется использовать только в том случае, если отчет содержит только одно вычисляемое поле с этой функцией.

Синтаксис 2:

`f_StringListEx (Describe('DataWindow.Syntax'),
Describe('DataWindow.Syntax.Data'), '*', '')`

Если отчет содержит более одного вычисляемого поля с вызовом функции `f_StringListEx`, то необходимо создать в заголовке отчета вычисляемое поле с использованием «Синтаксиса 2». Функция, вызванная с такими параметрами всегда возвращает пустую строку (''), но наличие такого поля необходимо для возможности использования «Синтаксиса 3».

Синтаксис 3:

`f_StringListEx ('*', '', <ColName>, <Separator>)`

Если отчет содержит более одного вычисляемого поля с вызовом функции `f_StringListEx`, рекомендуется использовать «Синтаксис 3», что дает значительное ускорение обработки отчета. Для использования необходимо вычисляемое поле в заголовке (см. «Синтаксис 2»).

Синтаксис 4:

`f_StringListEx ('debug', '', '*', '')`

На этапе настройки отчета вместо «Синтаксиса 2» можно использовать «Синтаксис 4». В результате все функции `f_StringListEx` («Синтаксис 3») будут возвращать пустую строку, но настраивать отчет станет легче (замедления работы программы практически не будет заметно). Внимание: после окончания настройки отчета не забудьте заменить «Синтаксис 4» на «Синтаксис 2».

2. При использовании функций, обрабатывающих SQL-запросы, следует в списке колонок запроса использовать имена не таблиц, а view. В противном случае, SQL-сервер может вернуть сообщение об отсутствии у пользователя прав на доступ к таблице.